Title
3D Scene and Event Understanding by Joint Spatio-temporal Inference and Reasoning

Author
Xu, Yuanlu

Publication Date
2019

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

3D Scene and Event Understanding by Joint Spatio-temporal Inference and Reasoning

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Yuanlu Xu

2019

ABSTRACT OF THE DISSERTATION

3D Scene and Event Understanding by
Joint Spatio-temporal Inference and Reasoning

by

Yuanlu Xu
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2019
Professor Song-Chun Zhu, Chair

It is a challenging yet crucial task to have a comprehensive understanding of human activities and events in the 3D scene. This task involves many many mid-level vision tasks (*e.g.*, detection, tracking, pose estimation, action/interaction recognition) and requires high-level understandings and reasoning about their relations. In this dissertation, we aim to propose a novel and general framework for both mid-level and high-level tasks under this track, towards a better solution for complex 3D scene and event understanding. Specifically, we aim to formulate problems with interpretable representations, enforce high-level constraints with domain knowledge guided grammar, learn models solving multiple tasks jointly, and infer based on spatial, temporal and casual information. We make three major contributions in this dissertation:

First, we introduce interpretable representations to incorporate high-level constraints defined by domain knowledge guided grammar. Specifically, we propose: i) Spatial and Temporal Attributed Parse Graph model (ST-APG) encoding compositionality and attribution for multi-view people tracking, enhancing trajectory associations across space and time, ii) Scene-centric Parse Graph to represent a coherent understanding of information obtained from cross-view scenes for multi-view knowledge fusion, iii) Fashion Grammar for constraining configurations of human appearance and clothing in human parsing, iv) Pose Grammar for describing physical and physiological relations among human body parts in human pose

estimation, and v) Causal And-Or Graph (C-AOG) to represent the causal-effect relations between an object's fluent changes and involved activities in tracking interacting objects.

Second, we formulate multiple related tasks into a joint learning, inference and reasoning framework for mutual benefits and better configurations, instead of solving each task independently. Specially, we propose: i) a joint parsing framework for iteratively tracking people locations and estimating people attributes, ii) a joint inference framework modeled by deep neural networks for passing messages from direct, top-down and bottom-up directions in the task of human parsing, and iii) a joint reasoning framework to reason object's fluent changes and track the object in videos, iteratively searching for a feasible causal graph structure.

Third, we mitigate the problem of data scarcity and data-hungry model learning using a learning-by-synthesis framework. Given limited training samples, we consider either propagate supervisions to unpaired samples or synthesizing virtual samples that minimize discrepancies with the realistic data. Specifically, we develop a pose sample simulator to augment training samples in virtual camera views for the task of 3D pose estimation, which improves our model cross-view generalization ability.

There are several interesting properties regarding the proposed frameworks: i) a novel perspective for problem formulation on joint inference and reasoning on space, time and causality, ii) overcoming the drawbacks of lack of interpretability and data hunger for end-to-end deep learning methods. Experiments show that our joint inference and reasoning framework outperforms existing approaches on many tasks and obtains more interpretable results.

The dissertation of Yuanlu Xu is approved.

Kai-Wei Chang

Ying Nian Wu

Demetri Terzopoulos

Song-Chun Zhu, Committee Chair

University of California, Los Angeles

2019

To my beloved family ...

who gave me all the courage and love along this path.

To my best friends ...

who accompanied me in my growing up years

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First, I would like to express my deep gratitude to my advisor, Prof. Song-Chun Zhu, for giving me this great opportunity to conduct researches in the Center of Vision, Cognition, Learning and Autonomy (VCLA). I am always deeply inspired by his high-level insight about computer vision and AI, passion about research, deep thinking of under-explored challenges during the journey. I would also like to thank my other committee members: Prof. Demetri Terzopoulos, Prof. Ying Nian Wu and Prof. Kai-Wei Chang for their help and supports.

Second, many thanks to our team members at VCLA. I feel like we are best friends and families along with the past five years of studying, working, living and playing together. I really enjoyed collaborating with them and always appreciate their supports during my doctoral study. Particularly, I'd like to thank Prof. Xiaobai Liu, Dr. Yang Liu, Prof. Lei Qin, Dr. Hang Qi, Dr. Wenguan Wang, Dr. Tony Tung and Hao-Shu Fang for their close collaborations with me. Many thanks also go to Prof. Tianfu Wu, Prof. Ping Wei, Dr. Xiaohan Nie, Dr. Joey Yu, Dr. Yixin Zhu, Tianmin Shu, Siyuan Qi and Siyuan Huang.

Last but not least, all the love goes to my family members, brothers and sisters. None of these would be possible without their countless love and support.

VITA

2014–2019 Research Assistant, VCLA, UCLA.

2018–2018 Research Intern, Oculus Research, Facebook Inc.

2014–2016 M.S. in Computer Science, UCLA.

2013–2014 Research Intern, NEC Laboratories, China.

2012–2014 M.S. in Computer Engineering, Sun Yat-Sen University, China.

2008–2012 B.E. in Software Engineering, Sun Yat-Sen University, China.

PUBLICATIONS

[1] **Yuanlu Xu**, Wenguan Wang, Xiaobai Liu, Jianwen Xie, Song-Chun Zhu, *"Learning Pose Grammar for Monocular 3D Pose Estimation"*, under review for IEEE TPAMI, 2019.

[2] **Yuanlu Xu**, Song-Chun Zhu, Tony Tung, *"Joint 3D Pose and Shape Estimation by Dense Render-and-Compare"*, under review for ICCV, 2019.

[3] Wenguan Wang*, **Yuanlu Xu***, Quanshi Zhang, Jianbin Shen, Song-Chun Zhu (* equally contributed), *"Deep Structured Neural Network with Joint and Interpretable Bottom-up and Top-down Inference"*, ArXiv Preprint, 2018.

[4] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, **Yuanlu Xu**, Song-Chun Zhu, *"Holistic 3D scene parsing and reconstruction from a single RGB image"*, ECCV, 2018.

[5] **Yuanlu Xu***, Lei Qin*, Xiaobai Liu, Jianwen Xie, Song-Chun Zhu (* equally contributed), *"A Causal And-Or Graph Model for Visibility Fluent Reasoning in Tracking Interacting Objects"*, CVPR, 2018.

[6] Wenguan Wang*, **Yuanlu Xu***, Jianbin Shen, Song-Chun Zhu, (* equally contributed) *"Attentive Fashion Grammar Network for Fashion Landmark Detection and Clothing Category Classification"*, CVPR, 2018.

[7] Hao-Shu Fang*, **Yuanlu Xu***, Wenguan Wang*, Xiaobai Liu, Song-Chun Zhu, (* equally contributed), *"Learning Pose Grammar to Encode Human Body Configuration for 3D Pose Estimation"*, AAAI 2018.

[8] Hang Qi*, **Yuanlu Xu***, Tao Yuan, Tianfu Wu, Song-Chun Zhu, (* equally contributed), *"Scene-centric Joint Parsing of Cross-view Videos"*, AAAI, 2018.

[9] Xiaobai Liu, **Yuanlu Xu**, Lei Zhu, Yadong Mu, *"A Stochastic Attribute Grammar for Robust Cross-View Human Tracking"*, IEEE TCSVT, 2018.

[10] Xiaobai Liu, Qi Chen, Lei Zhu, **Yuanlu Xu**, Liang Lin, *"Place-centric Visual Urban Perception with Deep Multi-instance Regression"*, ACMMM, 2017.

[11] **Yuanlu Xu**, Xiaobai Liu, Lei Qin, Song-Chun Zhu, *"Cross-view People Tracking by Scene-centered Spatio-temporal Parsing"*, AAAI, 2017.

[12] **Yuanlu Xu**, Xiaobai Liu, Yang Liu, Song-Chun Zhu, *"Multi-view People Tracking via Hierarchical Trajectory Composition"*, CVPR, 2016.

CHAPTER 1

Introduction

Though much progress has been made in 2D image/video based vision tasks, *e.g.*, object detection, tracking, human pose estimation, action recognition, it remains uncharted to leverage such estimations into the 3D world, due to the difficulty in data acquisition, ambiguities from monocular inputs and nuisance in natural images (*e.g.*, illumination, occlusion, texture). For example, as illustrated in Fig. 1.1, given the featured 2D RGB image, we could easily parse the content inside the image and leverage it into the 3D world. The holistic 3D scene and event parsing not only involves reconstructing 3D objects (*e.g.*, table, chair, people) and layouts (*e.g.*, wall, floor, ceiling), estimating human 3D poses and actions, but also engages high-level cognitive tasks about interactions, navigations, attentions and intentions.

In this dissertation, we focus on the task of understanding scene and event into the 3D world. Unlike common and popular 2D image/video based analytics (*e.g.*, recognition, detection, tracking, we first leverage information obtained from visual inputs with 3D spatial, structural and physical constraints, and then represent intra/inter-class as expressive and interpretable models (*e.g.*, grammar) to infer and reason the optimal configuration under the 3D world context.

We solve tasks related to the topic of 3D scene and event understanding under three kinds of input settings:

Directly estimating 3D objects, people and scene configurations from RGB image inputs. This is the most popular input type in the computer vision literature, as images can be easily captured and stored using widespread RGB cameras. Noticed monocular data is high ambiguous, we further consider two particular settings which provides richer information and enables high-level behavioral and cognitive analysis, *i.e.*, video and multi-view data.

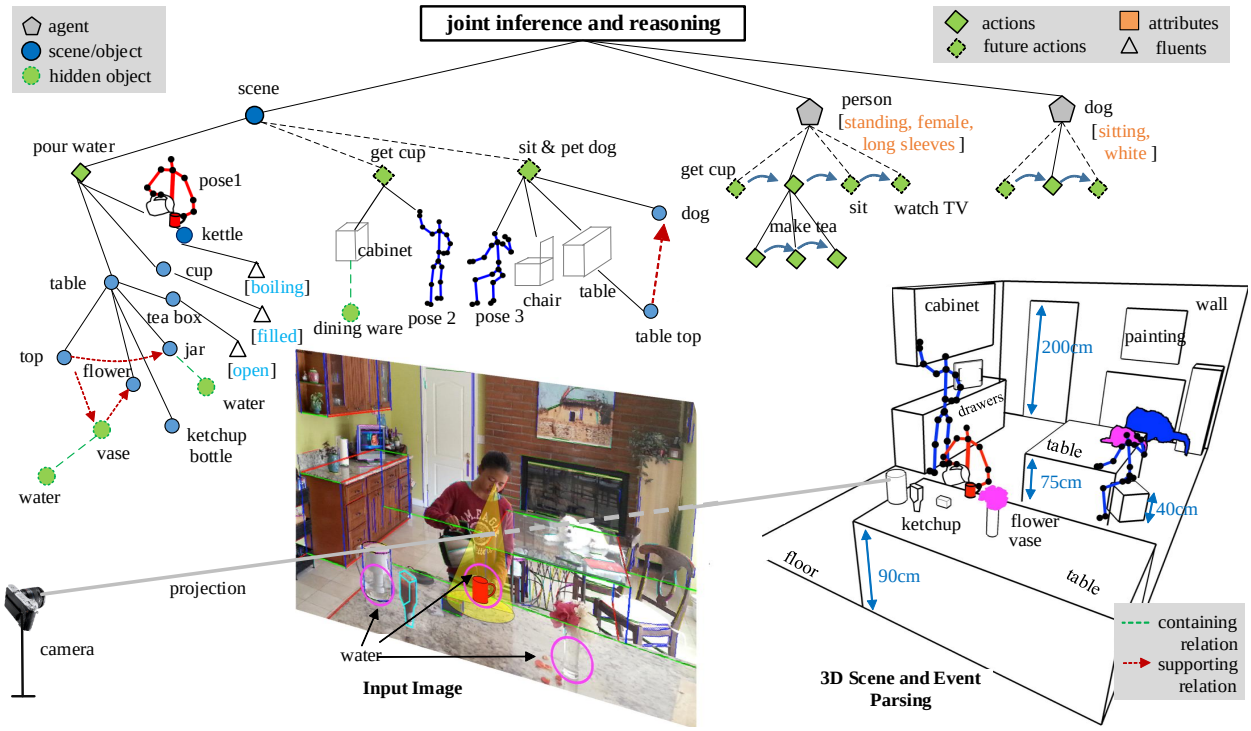


Figure 1.1: An illustration of 3D scene and event understanding.

Video data, in comparison with image data, encodes richer visual information and temporal consistency, which provides better roots for people to study many high-level tasks, *e.g.*, behavior, gait, relationship. Although video processing and analyzing has received more and more attentions from industrial and academic communities, works in both communities seem to focus on well defined low-level/middle-level tasks, *e.g.*, object detection, tracking and retrieval and lack the depth to study high-level tasks and how the high-level information affects and gets reflected on the low-level information. For example, many popular surveillance applications integrate multiple modules dealing with different tasks in a simple cascaded way and incapable of inferring low-level, middle-level and high-level information jointly.

We further consider cameras covering the same scene as a camera network and reconstruct 3D scene from multiple camera views using commonsense of scene geometry, and stitch a 3D scene from all cameras as global context. Then we parse objects, human pose, attributes actions, and group activities; project human and vehicle positions in 3D scene; and output their relations in 3D by cognitive reasoning in spatial and temporal parse graphs

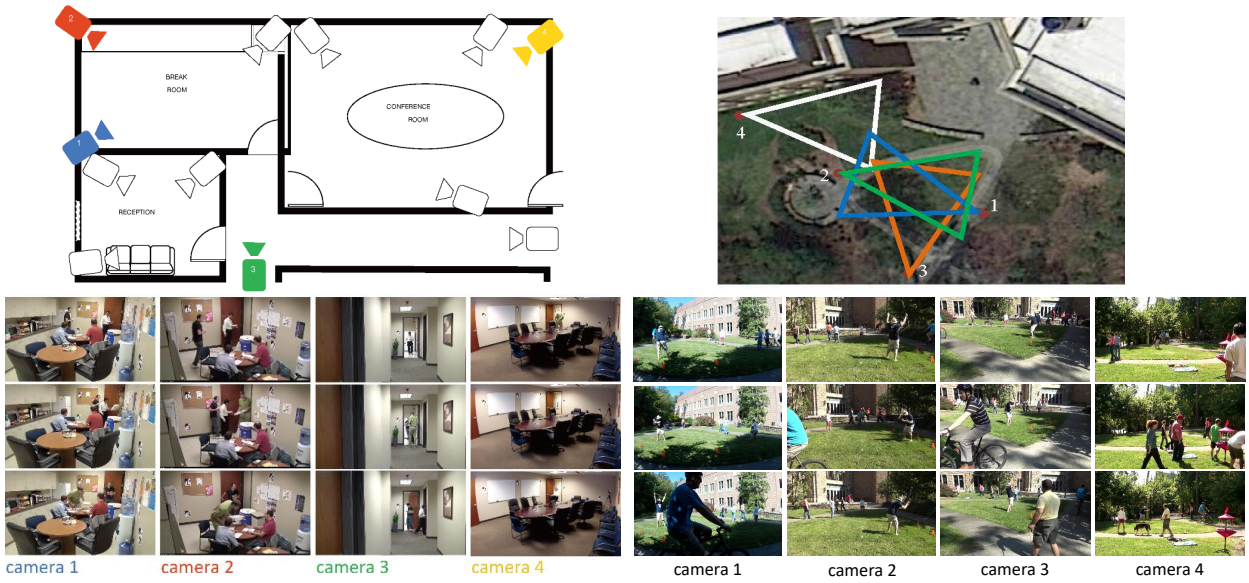


Figure 1.2: An illustration of indoor and outdoor scenes and corresponding multi-view camera networks.

with probabilities associated with nodes. For example, as shown in Fig. 1.2, an important public facility may be covered by several different cameras. There exist both overlapped regions (*e.g.*, the conference room) and non-overlapped regions (*e.g.*, the passageway). If we want to monitor and analyze what happens in this whole scenario, single camera can only provide limited information about certain agents. The information across different cameras, such as agents' identities, behaviors, are not associated together. Therefore, a joint inference across space and time is required to process information globally.

In this dissertation, we aim to solve basic perception tasks as well as advanced cognition tasks through a joint spatial-temporal inference and reasoning framework. As illustrated in Fig. 1.3, given image/video/multi-view data, we are interested in inferring low-level information (*e.g.*, trajectory, pose, attribute) and high-level information (*e.g.*, status, behavior) of a certain agent, and also the relations between this agent and other agents/objects in the scene. The analysis results from low-level are fine-tuned with constraints from high-level reasoning. For example, when the query agent misses, we want to figure out the reason that causes this. Is this agent occluded by some other agents? Does this agent exit and re-enter

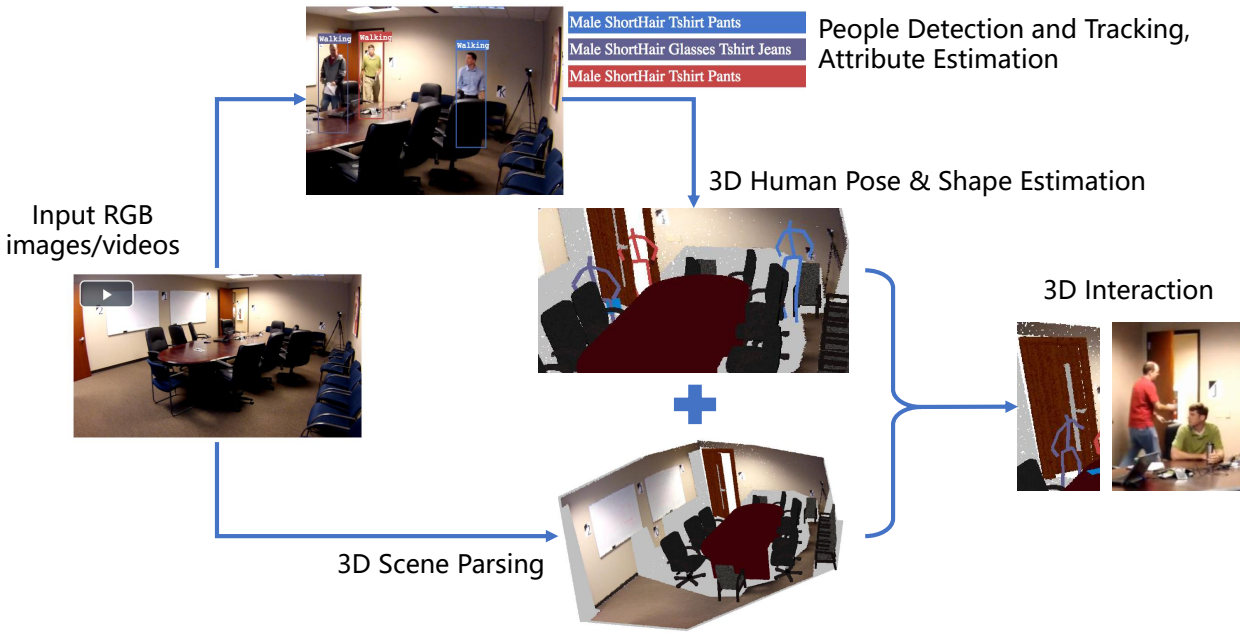


Figure 1.3: The proposed framework for 3D scene and event understanding.

the scene? Is this agent interacting with objects or other people? After figuring out the reason, the further step is to identify to what extend we can infer and recover the hidden information during the missing period. By commonsense, people know that different clues can be utilized and work in different cases.

In particular, we conduct studies in the following five topics in the rest chapters:

Multi-view People Tracking. We consider surveillance scenarios where there are multiple cameras monitoring an area (*e.g.*, parking-lot, garden) from different viewpoints. With streaming footages, we aim to recover the trajectories of all people in the scene. Different from single-view setting, a multi-view tracker is able to associate people in the long term and across camera views. Our algorithm aims to parse all people trajectories in the scene into a scene-centric representation which explicitly encodes various fine-grained attributes of humans in both spatial and temporal domains. Our representation encodes two principles: (i) compositionality, *i.e.*, decomposing a trajectory into sub-trajectories using multi-model information; (ii) attribution, *i.e.*, augmenting each trajectory elements with a set of fine-grained semantic/geometric attributes to enhance multi-view tracklet associations.

3D Scene Parsing. Based on the 3D trajectories obtained in multi-view people tracking, we further reconstruct and compose 3D scenes; infer the objects, human pose, actions, attributes and group activities in the global context of the scene; and output spatial and temporal parse graphs with probabilities associated with nodes. We focus on explicitly representing various constraints that reflect the appearance and geometry correlations among objects across multiple views and the correlations among different semantic properties of objects.

Human Pose and Attribute Estimation. This topic is in accordance with popular work of human parsing in the literature. We are inferring the trajectories, poses, appearance attributes of agents in images/videos. Noticed existing approaches often fail to directly encode interpretable structures and top-down information into their models due to the ambiguities of end-to-end learned deep neural networks, we specifically study expressive and interpretable representations that could organize different source of information.

3D Pose Estimation. Estimating 3D human poses from monocular RGB images has attracted growing interest in the past few years for its wide applications in robotics, autonomous vehicles, intelligent drones, *etc.* This is a challenging inverse task since it aims to reconstruct 3D spaces from 2D data and the inherent ambiguity is further amplified by other factors, *e.g.*, clothes, occlusions, background clutters. With the availability of large-scale pose datasets, *e.g.*, *Human3.6M* [IPO14], deep learning based methods have obtained encouraging success. We however, seek to encode domain-specific knowledge into current deep learning based detectors and improve performance and model robustness.

Event Understanding. This topic studies the relations between a specified agent and other agents/objects in the scene. For example, we may observe certain interactions such as getting-in/out, putting-in/out, throwing/fetching. In this track, we hope to understand how interactions affect the status of agents. Some typical and interesting problems are: tracking interacting objects, group activities/behaviors analysis, event understanding. Based on the initial results obtained in inference, we want to further refine them by reasoning. Supposed we have obtained the initial trajectory of an agent through basic spatio-temporal analysis, our purpose is to find out the abnormal parts. For example, the trajectory could be incomplete

and missing at certain moments. In this case, we need to infer the reason that causes this abnormality. Does the trajectory before and after the missing make sense? If so, we can further diagnose the casualty of the missing phenomenon. Finally, we can confirm whether the missing parts could be recovered.

CHAPTER 2

Multi-view People Tracking via Hierarchical Trajectory Composition

2.1 Introduction

Multi-view multi-object tracking has attracted lots of attentions in the literature [KGS09]. Tracking objects from multiple views is by nature a composition optimization problem. For example, a 3D trajectory of a human can be hierarchically decomposed into trajectories of individual views, trajectory fragments, and bounding boxes. While existing trackers have exploited the above principles more or less, they enforced strong assumptions over the validity of a particular cue, *e.g.* appearance similarity [ARS06], motion consistency [DCS13], sparsity [ML11, ZLA15], 3D localization coincidence [KS06], etc., which are not always correct. Actually, different cues may dominate different periods over object trajectories, especially for complicated scenes. In this chapter, we are interested in automatically discovering the optimal compositional hierarchy for object trajectories from various cues, in order to handle a wider variety of tracking scenarios.

As illustrated in Fig. 2.1, suppose we would like to track the highlighted subject and obtain its complete trajectory (e). The optimal strategy for tracking may vary over space and time. For example, in (a), since the subject shares the same appearance within certain time period, we apply an appearance based tracker to get a 2D tracklet; in (b) and (c), since the subject can be fully observed from two different views, we can group these two boxes into a 3D tracklet by testing the proximity of their 3D locations; in (d), since the subject is fully occluded in this view, we consider sampling its position from the 3D trajectory curve constrained by background occupancy.

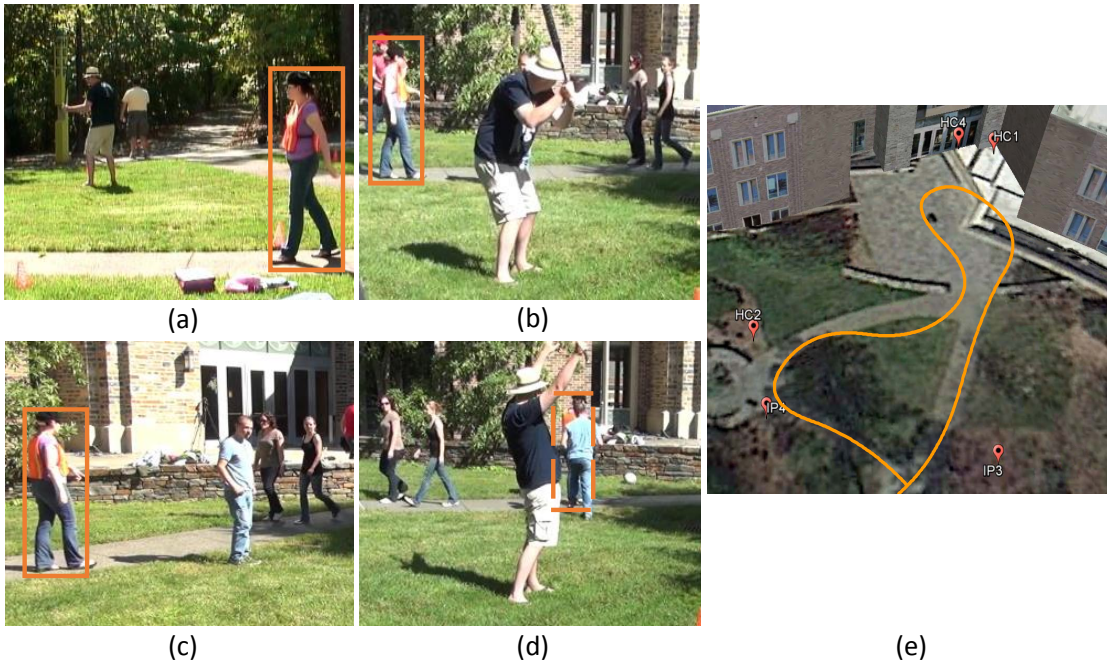


Figure 2.1: An illustration of utilizing different cues at different periods for the task multi--view multi-object tracking.

In this chapter, we formulate multi-view multi-object tracking as a structure optimization problem described by a hierarchical composition model. As illustrated in Fig. 2.2, our objective is to discover composition gradients of each object in the hierarchical graph. We start from structureless tracklets, *i.e.*, object bounding boxes, and gradually compose them into tracklets of larger size and eventually into trajectories. Each trajectory entity may be observed in single view or multiple views. The composition process is guided by a set of criteria, which describe the composition feasibility in the hierarchical structure.

Each criterion focuses on one certain cue and in fact is equivalent to a simple tracker, *e.g.*, appearance tracker [LLY11, XZW12], geometry tracker [SLX13], motion tracker [AS11], etc., which groups tracklets of the same view or different views into tracklets of larger sizes. Composition criteria lie in the heart of our method: feasible compositions can be conducted recursively and thus the criteria can be efficiently utilized.

To infer the compositional structure, we divest MCMC sampling-based algorithms due

to their heavy computation complexity. We approximate the hierarchy by a progressive composing process. The composition scheduling problem is solved by an iterative greedy pursuit algorithm. At each step, we first greedily find and apply the composition with maximum probability and then re-estimate parameters for the incremental part.

In the experiments, we evaluate the proposed method on a set of challenging sequences and the results demonstrate superior performance over other state-of-the-art approaches. Furthermore, we design a series of comparison experiments to systematically analyze the effectiveness of each criterion.

The main contributions of this work are two-fold. Firstly, we re-frame multi-view multi-object tracking as a hierarchical structure optimization problem and present three tracklet-based composition criteria to jointly exploit different kinds of cues. Secondly, we establish a new dataset to cover more challenges, to present richer visual information and to provide more detailed annotations than existing ones.

The rest of this chapter is organized as follows. We review the related work in Section 2.2, introduce the formulation of our approach in Section 2.3, and discuss the learning and inference procedures in Section 2.4. The experiments and comparisons are presented in Section 2.5, and finally comes a summary in Section 2.6.

2.2 Related Work

Our work is closely related to the following four research streams.

Multi-object tracking has been extensively studied in the last decades. In the literature, the tracking-by-detection pipeline [ZDS12, HLN13, PMR14, WLY14, DAS15, DTT15] attracts widespread attentions and acquires impressive results, thanks to the considerable progress in object detection [FGM10, SWJ13, RHG15], as well as in data association [ZLN08, PRF11, BFT11]. In particular, network flow based methods [PRF11, BFT11] organize detected bounding boxes into directed multiple Markov chains with chronological order and pursue the trajectory as finding paths. Andriyenko *et al.* [AS11] propose to track objects in

discrete space and use splines to model trajectories in continuous space. Our approach also follows this pipeline but considers bounding boxes as structureless elements. With preliminary associations to preserve locality, we can better explore the nonlocal properties [KGS13] of trajectories in the time domain. For example, tracklets with evident appearance similarities can be grouped together without considering the time interval.

Multi-view object tracking is usually addressed as a data association problem across cameras. The typical solutions include, homography constraints [KS06, ALD11], ground probabilistic occupancy [FBL08], network flow optimization [WHH09, BFT11, LPR12], marked point process [UB11], joint reconstruction and tracking [HWR13], multi-commodity network [SBF13] and multi-view SVM [ZYS15]. All these methods have certain strong assumptions and thus are restricted to certain specific scenarios. In contrast, we are interested in discovering the optimal composition structure to obtain complete trajectories in a wide variety of scenarios.

Hierarchical model receives heated endorsement for its effectiveness in modeling diverse tasks. In [HZ09], a stochastic grammar model was proposed and applied to solve the image parsing problem. After that, Zhao *et al.* [ZZ11] and Liu *et al.* [LCK14] introduced generative grammar models for scene parsing. Pero *et al.* [PBH13] further built a generative scene grammar to model the constitutionality of Manhattan structures in indoor scenes. Ross *et al.* presented a discriminative grammar for the problem of object detection [GFM11]. Grosse *et al.* [GSS12] formulated matrix decomposition as a structure discovery problem and solved it by a context-free grammar model. In this chapter, our representation can be analogized as a special hierarchical attributed grammar model, with similar hierarchical structures, composition criteria as production rules, and soft constraints as probabilistic grammars. The difference lies in that our model is fully recursive and without semantics in middle levels.

Combinatorial optimization receives considerable attentions in the surveillance literature [XLZ13]. When the solution space is discrete and the structure cannot be topologically sorted (*e.g.*, loopy graphs), there comes the problem of combinatorial optimization. Among all the solutions, MCMC techiques are widely acknowledged. For example, Khan *et*

al. [KS06] integrated the MCMC sampling within the particle filer tracking framework. Yu *et al.* [YMC07] utilized the single site sampler for associating foreground blobs to trajectories. Liu *et al.* [LLJ13] introduced a spatial-temporal graph to jointly solve the region labeling and object tracking problem by Swendsen-Wang Cut [BZ07]. In this chapter, though facing a similar combinatorial optimization problem, we propose a very efficient inference algorithm with acceptable trade-off.

2.3 Representation

In this section, we first introduce the compositional hierarchy representation, and then discuss the proposed problem formulation for multi-view multi-object tracking.

2.3.1 Hierarchical Composition Model

Given an input sequence containing videos shot by multiple cameras, we follow a default tracking-by-detection pipeline and apply [RHG15] to obtain detected bounding boxes. After that, we associate them into short trajectory fragments, *i.e.*, tracklets, similar to [HLN13, WWC14]. Tracklets preserve better local properties of appearance and motion as well as better robustness against errors and noises, compared with bounding boxes.

We denote a tracklet as O, which contains the appearance and geometry information over a certain period of time:

$$O = \{(a_i, l_i, t_i) : \ i = 1, 2, \ldots, |O|\}, \tag{2.1}$$

where a_i is the appearance feature, l_i the location information (*i.e.*, 2D bounding box and 3D ground position) and t_i the time stamp. Note that the 3D ground position is calculated by projecting the foot point of the 2D bounding box onto the world reference frame. For convenience, we denote the start time and end time of a tracklet by t^s and t^e, respectively. We further augment a set of states $x(O)$ for each tracklet O

$$x(O) = \{\omega_i : \ i = 1, \ldots, |O|\}, \tag{2.2}$$

where $\omega_i \in \{1, 0\}$ indicates the state of visibility/invisibility on the 3D ground plane at time

t_i. $x(0)$ describes the sparsity of a trajectory and can be utilized to enforce the consistency of object appearing and disappearing over time.

As shown in Fig. 2.2, we organize the scene as a compositional hierarchy $\mathbb{G}$ to recover the trajectory for each object in both single views and 3D ground. The compositional hierarchy $\mathbb{G}$ is denoted as

$$\mathbb{G} = (V_N, V_T, S, X), \tag{2.3}$$

where V_T denotes the set of terminal nodes, V_N indicates the set of non-terminal nodes, S is the root node representing the scene, and X represents the set of states of both terminal and non-terminal nodes.

A non-terminal node O is constructed by composing two nodes O_1 and O_2 together, that is

$$O \leftarrow f(O_1, O_2), \quad g_i(x(O)) = f_i(x(O_1), x(O_2)), \tag{2.4}$$

where $g_i(\cdot)$ and $f_i(\cdot)$ are associated operations on states. Note that $g_i(\cdot)$ and $f_i(\cdot)$ can assign states in either bottom-up or top-down direction, which act like functions of passing messages.

2.3.2 Bayesian Formulation

According to Bayes' rule, we can solve the problem of inferring the hierarchical composition model by maximizing a posterior, that is,

$$\mathbb{G}^* \;=\; \arg\max_{\mathbb{G}} p(\mathbb{G}|I) \;\propto\; \arg\max_{\mathbb{G}} p(I|\mathbb{G}) \cdot p(\mathbb{G}), \tag{2.5}$$

where I denotes the input video data.

Prior. Due to the property of hierarchy, we can further factorize the prior $p(\mathbb{G})$ as

$$p(\mathbb{G}) = \prod_{O_i \in V_N} p(x(O_i)) \prod_k p_k^{cp}(O_{i1}, O_{i2})^{\delta_i == k}, \tag{2.6}$$

where δ_i is an indicator for the type of criterion used in composition, and O_{i1} and O_{i2} are two children nodes of tracklet O_i.

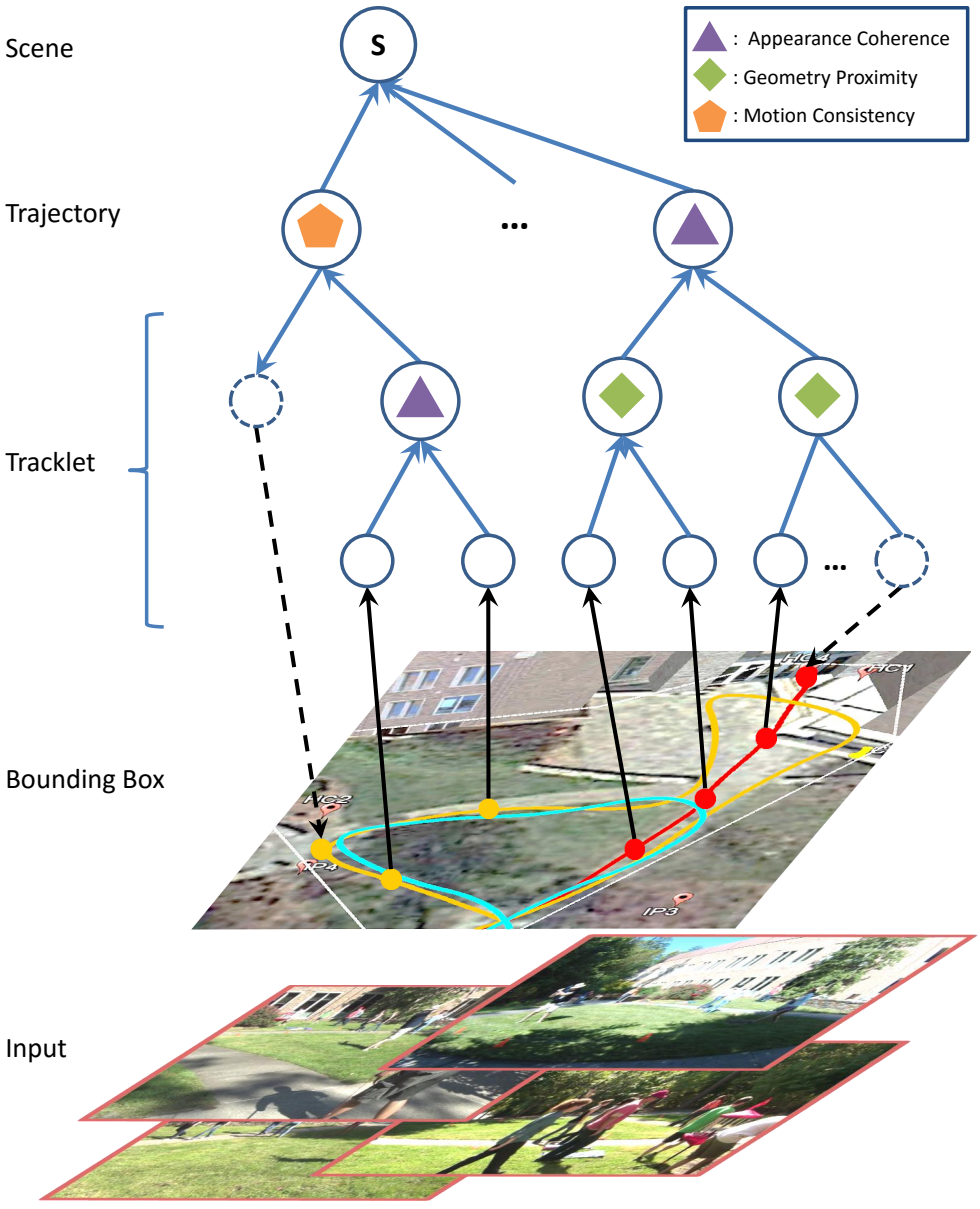


Figure 2.2: An illustration of the hierarchical compositional structure.

$p(x(O))$ is a unary probability defined on the state of O. We employ a simple Ising/Potts model to penalize the discontinuity of the trajectory, *i.e.*,

$$p(x(O)) \propto \exp\{ -\beta \sum_{i=1}^{|O|-1} \mathbf{1}(\omega_i \neq \omega_{i+1}) \}, \tag{2.7}$$

where β is a coefficient. $p(x(O))$ in fact constrain the number of times a trajectory switches between visible and invisible.

$p_k^{cp}(O_i, O_j)$ represents the composition probability using the k-th type of cue. We will discuss details about of composition criteria in Section 2.3.3.

Likelihood. The video data I is only dependent on the terminal nodes V_T and can be further decomposed as

$$\begin{aligned} p(I|\mathbb{G}) &= \left(\prod_{O_i \in V_T} \prod_{a_j \in O_i} p^{fg}(a_j) \right) \cdot \prod_{a_j \in I \setminus V_T} p^{bg}(a_j) \\ &= \prod_{O_i \in V_T} \prod_{a_j \in O_i} \frac{p^{fg}(a_j)}{p^{bg}(a_j)} \cdot \prod_{a_j \in I} p^{bg}(a_j), \end{aligned} \tag{2.8}$$

where $p^{fg}(\cdot)$ and $p^{bg}(\cdot)$ are foreground and background probabilities, respectively. The second term $\prod_{a_j} p^{bg}(a_j)$ measures the background probability over the entire video data and thus can be treated as a constant, and the first term measures the divergence between foreground and background, which can be analogous to a probabilistic foreground/background classifier. We use the detection scores to approximate this log-likelihood ratio.

2.3.3 Composition Criteria

In this section, we introduce details of the proposed composition criteria.

Appearance Coherence. Instead of using traditional descriptors (*e.g.*, SIFT, color histograms, MSCR) to measure the appearance discrepancy, we employ the powerful DCNN to model people's appearance variations. Notice that most DCNNs are trained over generic object categories and insufficient to provide fine-grained level of information about peoples identities [XMH14]. We therefore fine-tune the CaffeNet [JSD14] using people image samples with identity labels. The new DCNN consists of 5 convolutional layers, 2 max-pooling

layers, 3 fully-connected layers and a final 1000-dimensional output. The last two layers are discarded and replaced by random initializations. The output is new 1000 labels on people's identities. Note the training samples are augmented from unlabeled data and identity labels are obtained in an unsupervised way.

Similar to bag-of-words (BoW), our DCNN plays the role of a codebook, which codes a person image with common people appearance templates. We use this 1000-dimensional output as our appearance descriptor. Given two tracklets O_i and O_j, the appearance coherence constraint $p_1^{cp}(O_i, O_j)$ is defined as

$$p_1^{cp}(O_i, O_j) \propto \exp\{-\frac{\sum_{a_n \in O_i} \sum_{a_m \in O_j} \|a_n - a_m\|_2}{|O_i| \cdot |O_j|}\}. \tag{2.9}$$

$p_1^{cp}(O_i, O_j)$ actually measures the mean complete-link appearance dissimilarities among object bounding boxes belonging to two tracklets.

Geometry Proximity. Given tracklets from a single view or cross views, we first project them on the world reference frame to measure their geometric distances uniformly. However, considering tracklets with different time stamps and lengths, it is not a trivial task to determine whether the two given tracklets belong to the same object or not. The reason lies in: i) the time stamps of tracklet pairs might not be well aligned; ii) the localizations across views usually lead to remarkable amount of errors.

In order to address these issues, we introduce a kernel to measure these time series samples. The kernel $K(O_i, O_j)$ to measure the distance between two tracklets O_i and O_j is defined as the product of two kernel distances in space and time

$$K(O_i, O_j) = \sum_{(l_n, t_n) \in O_i} \sum_{(l_m, t_m) \in O_j} \frac{\phi_l(l_n, l_m) \cdot \phi_t(t_n, t_m)}{|O_i| \cdot |O_j|}, \tag{2.10}$$

where $\phi_l(l_n, l_m)$ and $\phi_t(t_n, t_m)$ are two RBF kernels between two points. We use different σ_l and σ_t values for the two kernels, respectively. This new kernel acts like a sequential convolution filter and takes both spatial and temporal proximities into consideration.

Given a set of training samples D,

$$D = \{(O_i, O_j, y_n) : \ n = 1, \ldots, |D|\}, \tag{2.11}$$

where $y_n \in \{1, 0\}$ indicates whether or not the two tracklets O_i and O_j belong to the same identity, we can train a kernel SVM with the energy function

$$\min_w \frac{1}{2} < w, w > + C \sum_n \max(0, 1 - y_n < w, K(O_i, O) >), \tag{2.12}$$

where C is a regularization factor.

We therefore interpret the normalized classification margin as the composition probability $p_2^{cp}(O_i, O_j)$.

Motion Consistency. We model the motion information of a tracklet O as a continuous function of its 3D ground positions l w.r.t. time t, *i.e.*, $l = \tau(t)$. We define a constraint on two tracklets that they can be interpreted with the same motion function. However, finding this motion pattern is a challenging problem. The reason lies in two-fold: i) inaccurate 3D positions due to perspective effects, detection errors and false alarms; ii) missing detections and object inter-occlusions in certain views, especially for crowded scenarios. In this chapter, we address these issues in the following two aspects.

Firstly, we employ the b-spline function to represent the motion pattern of the trajectory. B-spline functions can enforce high-order smoothness constraints, which enables learning from sparse and noisy data. Considering a tracklet O with 3D positions $\{l_i : i = 1, \ldots, |O|\}$, starting time t^s and ending time and t^e, the spline function $\tau(t)$ uses some quadratic basis functions $B_k(t)$, and represents the motion path as a linear combination of $B_k(t)$:

$$\begin{aligned} &\tau(t) = \sum_k \alpha_k B_k(t), \\ &s.t. \;\; \tau''(t^s) = \tau''(t^e) = 0, \end{aligned} \tag{2.13}$$

where $\tau''(t)$ denotes the second derivative of $\tau(t)$. The constraints enforce zero curvature at the starting and the ending point.

Secondly, we take advantages of the multi-view setting and derive feasible regions for object 3D positions to further confine the fitted motion curve. As illustrated in Fig. 2.3, given bounding boxes of a single object in the views (a), (b) and (c), we first perform exhaustive search to find the two anchor points (yellow dots in the image) along two sides of the foot position of each object. An anchor point is defined as a position where the surrounding

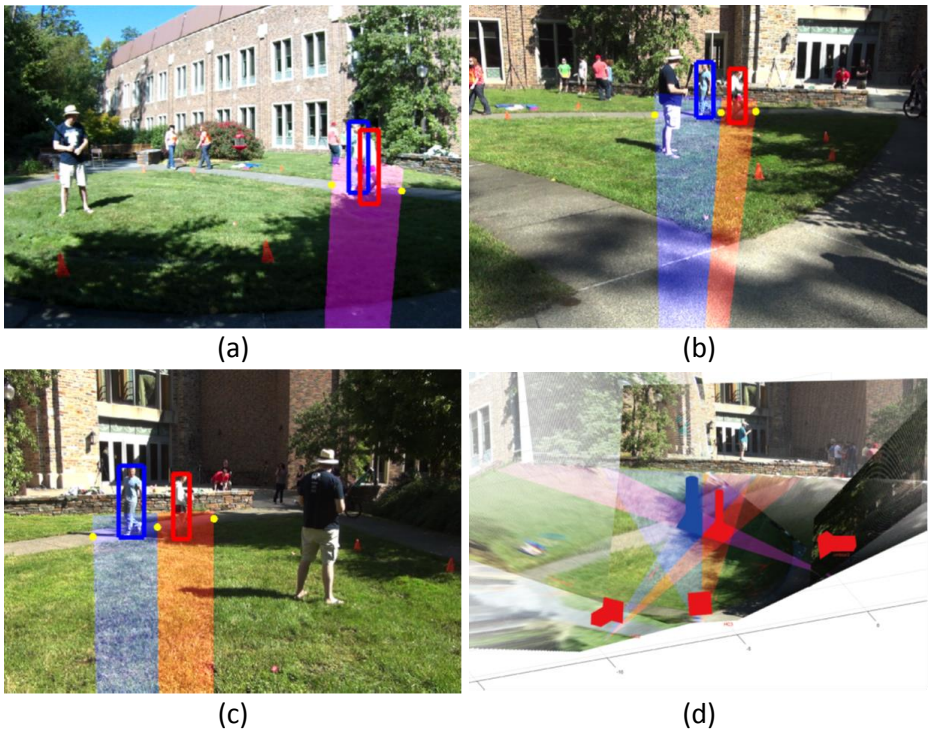


Figure 2.3: An illustration of finding feasible regions (polygons) for interacting people.

8×8 area contains most of background regions. Note that we generate background masks by GMM background modeling.

Once obtaining all the anchor points for an object, we can find the union area Ω, *i.e.*, a polygon on the world ground plane, as shown the shaded area in (d). These polygons serve as additional localization feasibility constraints on the motion pattern. That is, the spline fitting is formulated as minimizing the following objective function:

$$\min_{\alpha_k, B_k} E(O_i, O_j) = \sum_{(l_n, t_n) \in O_i \cup O_j} \left(l_n - \sum_k \alpha_k B_k(t_n) \right)^2, \quad s.t. \quad \alpha_k B_k(t_n) \in \Omega_n. \tag{2.14}$$

This is a constrained convex programming problem considering that all polygons are convex. We refer the readers to find more details about b-spline and robust fitting algorithms in [EM96].

The probability $p_3(O_i, O_j)$ is defined upon the averaged residuals for spline fitting, *i.e.*,

$$p_3^{cp}(O_i, O_j) \propto \exp\{-\frac{E(O_i, O_j)}{|O_i \cup O_j|}\}. \tag{2.15}$$

2.4 Learning and Inference

In this section, we first discuss the learning procedure for our constraints and then introduce how to infer the hierarchical compositional structure.

2.4.1 Learning Constraints

Appearance Coherence. Even for fine-tuning a DCNN, fair amount of training samples are required. We therefore augment the training data by external samples from public people detection datasets, *e.g.*, CaltechPedestrians, NICTA, ETH and TUD-Brussels. The augmented training set contains around 30,0000 samples of cropped people images. We resize all the samples to 128×256 and horizontally flip them to double the training set size. And then we extract dense HSV color histograms with 16 bins from 16×16 non-overlapping patches for each image. The computed histograms are concatenated into a 6144-dimensional feature vector. We perform K-means clustering on the data and obtain 1000 clusters. Each cluster is regarded as a class and we utilize them to fine tune our DCNN. In general, the fine-tuning process converges after 100000 iterations and costs about 8 hours.

Geometry Proximity. Given the training data and ground-truth of a scenario, we first generate initial tracklets and then associate them with the ground-truth. A tracklet is treated as a fragment of a ground-truth trajectory if more than 50% of its bounding boxes are correctly assigned (*i.e.*, hit/miss cutoff with 50% IoU ratio). The training data set D can thus be constructed using tracklets from the same trajectory as positive pairs samples and those from different trajectories as negative pairs. We learn the parameters of our kernel SVM for each pair of views (including self-to-self). The kernel parameters σ_l and σ_t are also tuned by cross-validation.

Note we also estimate the normalization constant for each constraint $p_k^{cp}(O_i, O_j)$ using

the training data.

2.4.2 Inferring Hierarchy

Our objective is to find a compositional hierarchy $\mathbb{G}$ by maximizing the posterior probability formulated in Equation (2.5). The optimization algorithm should accomplish two goals: i) composing hierarchical structures, and ii) estimating states for terminal and non-terminal nodes.

The main challenge in optimizing Equation (2.5) lies in the size of the solution space. For example, if there are n terminal nodes, even a single group can be formed in 2^{n-1} different ways, which is exponential. Although MCMC sampling-based algorithms [LLJ13, XLZ13] are favored to solve such kinds of combinatorial optimization problems, they are typically computationally expensive and difficult to converge, especially for our case, with thousands of terminal nodes and numerous possible compositions.

Hereby, we approximate the construction of the hierarchical structure by a progressive composing process. In the beginning, given a set of initial tracklets V_T, we initialize the state $\omega_i \in x(O)$ for each tracklet O as visible. We then enumerate all the tracklets over all composition criteria, and find two tracklets O_i and O_j with maximum probability to be composed into a new tracklet O_n, that is,

$$\max_{O_i, O_j, \delta_n} p(x(O_n)) \prod_k p_k^{cp}(O_i, O_j)^{\delta_n == k}, \tag{2.16}$$

where δ_n is an indicator for which cue is selected. We then group these two tracklets O_i and O_j together, and create their parent node O_n.

The states for this newly merged node O_n are re-estimated by

$$\begin{aligned} &x(O_n) = x(O_i) \cup x(O_j), \\ &t_n^s = \min(t_i^s, t_j^s), \quad t_n^e = \max(t_i^e, t_j^e), \\ &|x(O_n)| = t_n^e - t_n^s + 1. \end{aligned} \tag{2.17}$$

Note we set all the states of missing time stamps within the time scope $[t_n^s, t_n^e]$ to 0, *i.e.*, invisible. This encourages future filling-in operations.

If a composition performed based on motion consistency constraint, we then fill in the missing fragments by interpolations, and create a corresponding tracklet $O_m \in V_T$. The new tracklet O_m will be naturally incorporated into the hierarchical structure by subsequent compositions.

We continue this process iteratively. If the maximum composition probability reaches the lower limit, we terminate the algorithm and connect all the top non-terminal nodes to the root node S. Each sub-tree connected to the root node is essentially an object trajectory.

2.5 Experiment

In this section, we first introduce the datasets and the parameter settings, and then show our experimental results as well as component analysis of the proposed approach.

2.5.1 Datasets and Settings

We evaluate our approach on three public datasets:

(i) **EPFL dataset** [FBL08][1]. We adopt the Terrace sequence 1, Passageway sequence and Basketball sequence in our experiments. In general, each sequence consists of 4 different views and films 6-11 pedestrians walking or running around, lasting 3.5-6 minutes. Each view is shot at 25fps and in a relatively low resolution 360×288.

(ii) **PETS 2009 dataset** [FS09][2]. This dataset is widely used in evaluating tracking tasks and sequence S2/L1 is specially designed for multi-view-based tasks. With 3 surveillance cameras and 4 DV cameras, 10 pedestrians are recorded entering, passing through, staying and exiting the pictured area. The video is down-sampled to 720×576 and the frame rate is set to 7 FPS.

(iii) **CAMPUS dataset**. To cover more complete challenges not presented in existing databases, we design this dataset based on the idea of dense foreground (around 15-25 ob-

[1]Available at https://cvlab.epfl.ch/data/pom/

[2]Available at http://www.cvg.reading.ac.uk/PETS2009/a.html

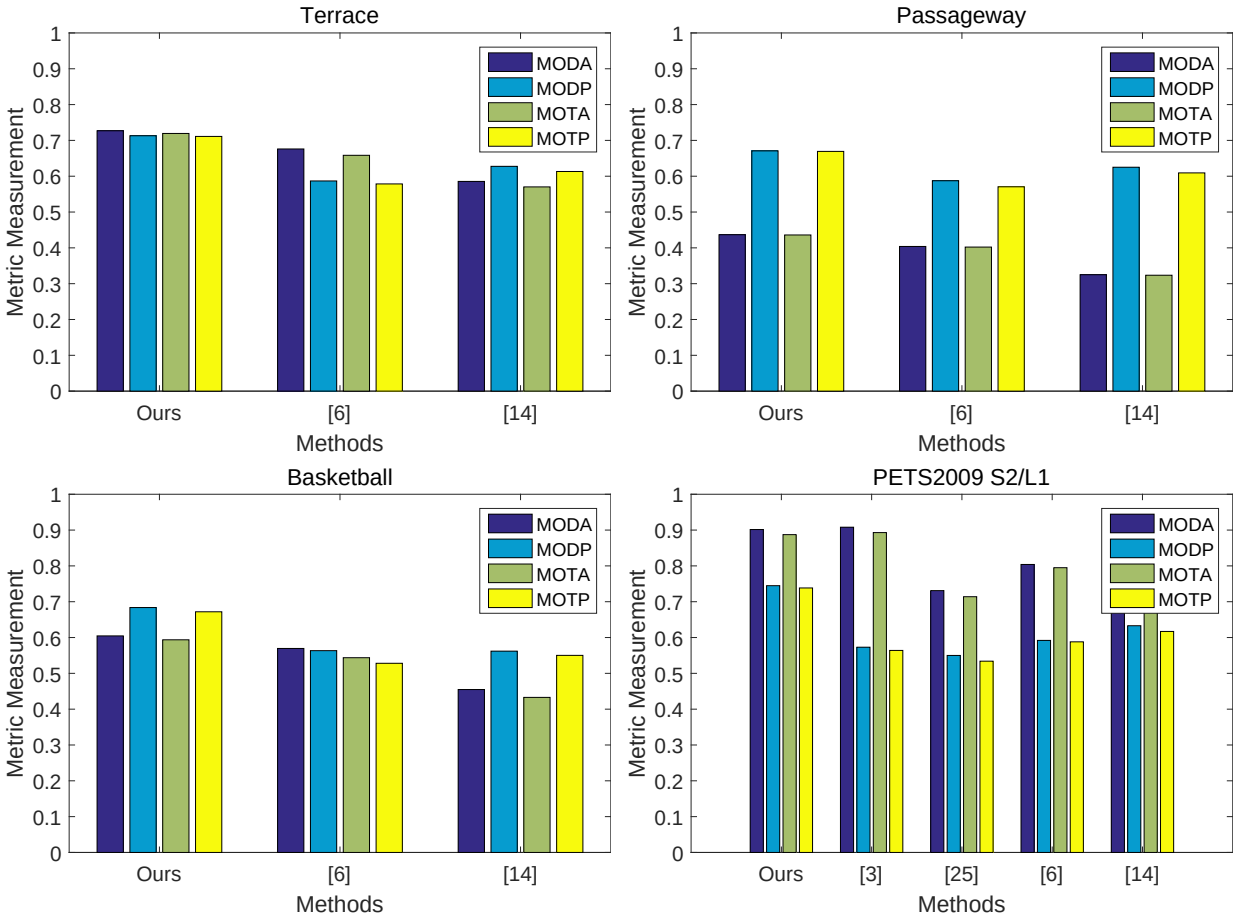


Figure 2.4: Comparison charts using CLEAR metrics on EPFL and PETS 2009 datasets.

jects, frequent conjunctions and occlusions), complex scenarios (objects conducting diverse activities, dynamic background, interactions between objects and background), various object scales (tracking targets sometimes either too tiny or huge to be accommodated in certain cameras). We incorporate 4 sequences into this dataset: Garden 1, Garden 2, Auditorium and Parking Lot. Each sequence is shot by 3-4 high-quality DV cameras mounted around 1.5-2 meters above ground and each camera covers both overlapping regions and non-overlapping regions with other cameras. The videos are recorded with frame rate 30 FPS and duration about 3-4 minutes. The resolution is preserved in 1920×1080, for better precision and richer information.

For all three datasets, videos in each sequence are synchronized. We fully annotate the

ground-truth trajectories for all the videos in all the sequences using [VPR13]. Note that we assign an unique ID for each object, whether it appears once or several times in the scene. Since the ultimate task of multi-view multi-object tracking is to discover the complete 3D trajectory of any targeted individual under a camera network, we believe uniquely assigned ID should be the ground-truth to fully evaluate the trackers, which poses higher requirements than conventional tracking tasks [KGS09]. In experiments, we use the beginning 10% video data for training and the rest for testing.

All the parameters are fixed in the experiments. For object detection, we use the PASCAL VOC fine-tuned ZF net, score threshold 0.3 and NMS threshold 0.3, which obtains proper trade-off between the efficiency and effectiveness. As for tracklet initialization, we construct a graph with edges only connected among successive frames and within limited scale changes. That is, sizes of two successive bounding boxes should not change more than 25% larger or smaller, in either height or width. We then run the successive shortest path algorithm [PRF11] to generate tracklets. Empirically, this produces short but identity consistent tracklets. $\beta = 0.05$ in the unary probability $p(x(O))$. The motion consistency constraint is conducted on tracklets with time interval no longer than 2 seconds, with the B-spline of order at most 3 and breaks at most 4. In the hierarchical composition, the lower limit is set to 0.2, which obtains good results.

2.5.2 Experimental Results

We employ the widely used CLEAR metrics [KGS09], Multiple Object Detection Accuracy (MODA), Detection Precision (MODP), Tracking Accuracy (MOTA) and Tracking Precision (MOTP) to measure three kinds of errors in tracking: false positives, false negatives and identity switches. Besides, we also report the percentage of *mostly tracked* (MT), *partly tracked* (PT) and *mostly lost* (ML) ground-truth (referring to [LHN09]), as well as the number of identity switches (IDSW) and fragments (FRAG). Hit/miss for the assignment of tracking output to ground-truth is set to a threshold of Intersection-over-Union (IoU) ratio 50%.

Sequence	Method	MODA(%)	MODP(%)	MOTA(%)	MOTP(%)	MT(%)	PT(%)	ML(%)	IDSW	FRAG
Garden1	Our-full	**49.30**	72.02	**49.03**	71.87	31.25	62.50	6.25	299	200
	Our-3	44.63	72.35	44.36	72.20	18.75	68.75	12.50	296	202
	Our-2	42.10	71.08	41.69	70.97	12.50	75.00	12.50	448	296
	Our-1	41.21	71.06	37.21	70.94	12.50	75.00	12.50	4352	4390
	[BFT11]	30.47	62.13	28.10	62.01	6.25	68.75	25.00	2577	2553
	[FBL08]	24.52	64.28	22.43	64.17	0.00	56.25	43.75	2269	2233
Garden2	Our-full	**27.81**	71.74	**25.79**	71.59	21.43	78.57	0.00	94	73
	Our-3	23.39	71.13	22.50	71.08	14.29	85.71	0.00	92	72
	Our-2	18.76	70.20	17.27	70.12	14.29	78.57	7.14	142	97
	Our-1	17.68	70.12	10.24	70.11	14.29	78.57	7.14	700	733
	[BFT11]	24.35	61.79	21.87	61.64	14.29	85.71	0.00	268	249
	[FBL08]	16.51	63.92	13.95	63.81	14.29	78.57	7.14	241	216
Auditorium	Our-full	**20.84**	69.26	**20.62**	69.21	33.33	55.56	11.11	31	28
	Our-3	18.83	68.99	18.62	68.95	22.22	61.11	16.67	30	28
	Our-2	18.02	68.32	17.29	68.25	16.67	66.67	16.67	104	94
	Our-1	17.78	68.33	14.11	68.28	16.67	66.67	16.67	523	536
	[BFT11]	19.46	59.45	17.63	59.29	22.22	61.11	16.67	264	257
	[FBL08]	17.90	61.19	16.15	61.02	16.67	66.67	16.67	249	235
ParkingLot	Our-full	**24.46**	66.41	**24.08**	66.21	6.67	66.67	26.67	459	203
	Our-3	19.23	66.50	18.84	66.38	0.00	53.33	46.67	477	191
	Our-2	12.85	65.70	12.23	65.61	0.00	46.67	53.33	754	285
	Our-1	10.86	65.77	8.74	65.72	0.00	46.67	53.33	2567	2600
	[BFT11]	14.73	58.51	13.99	58.36	0.00	53.33	46.67	893	880
	[FBL08]	11.68	60.10	11.00	59.98	0.00	46.67	53.33	828	812

Table 2.1: Quantitative results and comparisons on CAMPUS dataset. Our-1, Our-2, Our-3 are three benchmarks set up for component evaluation. See text for detailed explanations.

We compare the proposed approach with 4 state-of-the-arts methods: Probabilistic Occupancy Map (POM) [FBL08], K-Shortest Path (KSP) [BFT11], Branch-and-Price [LPR12] and Discrete-Continuous Optimization [AS12]. We adopt the public code of POM detection and implement the data association algorithms "DP with appearance" [FBL08] and KSP [BFT11] according to their descriptions. The reported metrics for comparing methods are quoted on PETS 2009 dataset from [ESF09] and computed on the rest by conducting experiments.

Quantitative evaluations on EPFL and PETS 2009 datasets is shown in Fig. 2.4 and CAMPUS dataset in Table 2.1, as well as qualitative results in Fig. 2.5. From the results, our method demonstrates superior performance over the competing methods. We can also observe the proposed method acquires significant margins on MODP, MOTP, IDSW and FRAG, which indicates two empirical conclusions: i) detection-based tracklet initialization is more beneficial to object overall localization than foreground-blob-based methods which mainly concerns ground positions; ii) when it comes to occlusions, multiple cues (*e.g.*, appearance, geometry, and motion) are all neccessary to keep the trajectory identity consistent, which has also been approved in [HHR13]. Competing methods do not work well on CAMPUS dataset mainly due to their strong dependence on clear visibility of ground plane and uniform object size.

Component Analysis. We set up three benchmarks to further analyze the benefits of each production rule on CAMPUS dataset. *Our-1* outputs the initial tracklets directly, *i.e.*, no composition performed; *Our-2* composes the hierarchy only using the appearance coherence criterion; *Our-3* further incorporates the geometry proximity criterion; *Our-full* employs all criteria proposed in this chapter. From the results, it is apparent that each constraint contributes to a better hierarchical composition model.

Efficiency. Our method is implemented in MATLAB and runs on a desktop with Intel I7 3.0GHz CPU, 32GB memory and Nvidia GTX780Ti GPU. Given a 1080P sequence, the runtime on average is 15-20 FPS for object detection, 1000-1500 FPS for tracklet initialization, and 2-4 FPS for optimizing the hierarchical structure. Overall, the proposed algorithm obtains 1-3 FPS, which is related to the object density of the sequence. With proper code

Figure 2.5: Results generated by the proposed method on CAMPUS, EPFL and PETS 2009 datasets.

migration and optimization, *e.g.*, batch processing, we believe the real-time processing can be achieved.

2.6 Summary

In this chapter, we study a novel formulation for multi-view multi-object tracking. We represent object trajectories as a compositional hierarchy and construct it with probabilistic constraints, which characterize the geometry, appearance and motion properties of trajectories. By exploiting multiple cues and composing them with proper scheduling, our method handles challenges in multi-view multi-object tracking well. Furthermore, we will explore more powerful inter-tracklet relations and better composition algorithms in the future.

CHAPTER 3

Cross-view People Tracking by Scene-centered Spatio-temporal Parsing

3.1 Introduction

In this chapter, we study a novel cross-view tracklet association algorithm for multi-view person tracking. We consider surveillance scenarios where there are 3-4 cameras looking at a target area (*e.g.*, parking-lot, garden) from different viewpoints. The task is to compute the scene-centered overall trajectory of all the people within the scene. In comparison with the single-view setting [LLJ13, AS11, WBK11, DAS15], it remains unclear how to associate people trajectories across views, especially when the cameras have wide baselines or large view changes.

• Large appearance variations. A person is assumed to have similar appearance across space and time. Nevertheless, large camera view and scale changes compromise such assumption. For example, Fig. 3.1 shows a garden covered by four cameras. From these camera view snapshots, the person in navy blue looks different in front and back view.

• Inaccurate geo-localization. A common way for solving the task is to calibrate camera parameters and utilize cross-camera ground homographs, with which a person detected in one viewpoint can be registered in another view. However, the registration results are often not accurate enough to separate humans in the proximity because of the calibration errors or the inaccuracy of footprint estimation. For example, in Fig. 3.1 (c-d), people's feet are occluded by the wall and so it is difficult to register the detected human feet positions in other views.

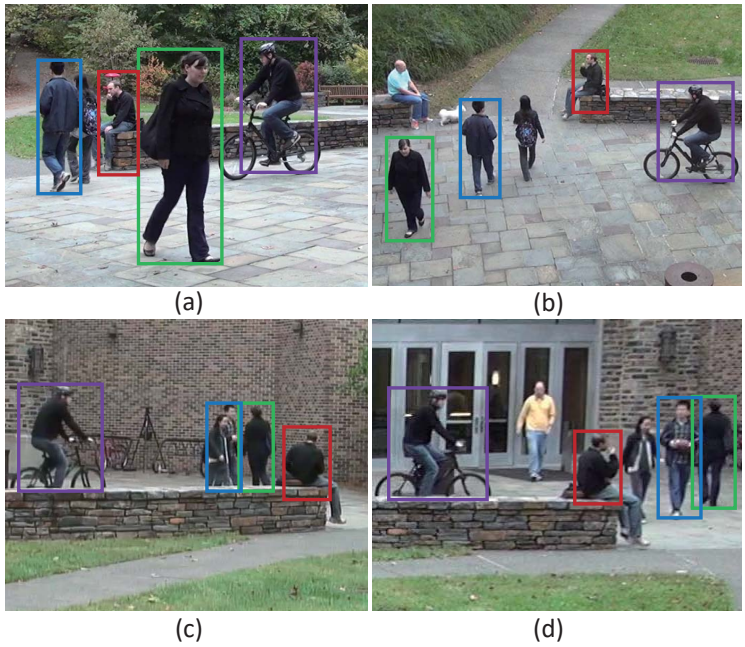


Figure 3.1: An example of cross-view data association for target tracking. (a)-(d) represents four different camera views of the same scene. Each color of the bounding box represents a unique person.

The main idea of our approach is to leverage semantic attributes, *e.g.*, facing orientations, poses and actions (standing, running, etc.), for cross-view tracklet association. Taking Fig. 3.1 for example, attributes of person can help prune the ambiguities in cross-view data association. Specifically, if the orientation of every human box can be correctly identified, we can associate the green box across views because there is only one person facing the building. In addition, since there is only one person sitting (red boxes) and one person on the bike (purple boxes), the pose and action recognition can be used to narrow down the association space. With the recent advances in computer vision and machine learning, these semantic attributes can be readily detected with a level of accuracy from a single view, serving as powerful cues for associating human boxes or trajectories across cameras.

We use Spatio-temporal Attributed Parse Graph (ST-APG) to integrate the semantic attributes with the people trajectories, and pose multi-view people tracking as spatio-temporal parsing problem. As illustrated in Fig. 3.2, the scene is decomposed into people trajectories and trajectories consists of tracklets with the same identity. A tracklet is a series of

human boxes grouped by spatial coherency and perceptual similarity. The parse graph is enriched with attributes across different levels. The scene is incorporated with the camera information while tracklets with four types of attributes: i) appearance; ii) geometry, *e.g.*, footprints; iii) motion, *e.g.*, facing direction and speed; iv) pose/action, *e.g.*, standing, sitting, walking, running, biking. These attributes can be recognized with a single image or a monocular video. We use these attributes to impose consistency constraints for cross-view tracklet associations. The constraints are used as additional energy term in the probabilistic formula, instead of hard constraints, to reduce errors made in bottom-up predictions.

To infer the ST-APG, we propose an efficient algorithm dealing with two sub-problems. I) We first employ a stochastic clustering algorithm [BZ05] to group the tracklets, which can efficiently traverse the combinatorial solution space. We explore two types of relationships among tracklets: i) being cooperative, *i.e.*, tracklets from different view are allowed to be grouped together according to their appearance and semantic attributes; ii) being conflicting, *e.g.*, tracklets with temporal overlaps in the same view, are conflicted to be grouped together. The conflicting relationships explicitly express the structure of the solution space. II) We use Dynamic Programming (DP) to estimate semantic attributes of the grouped tracklets. The trajectory is represented as a Markov Chain and DP are guaranteed to find the optimal solution. These two algorithms run iteratively until convergence.

We evaluate our approach on one public multi-view tracking dataset and collect a new multi-view dataset to cover daily activities (*e.g.*, touring, dining, working). We use 4 GOPRO cameras to capture synchronized videos for 3 scenarios, including food court, office reception and plaza, which provides rich actions and activities. Results and comparisons with popular trackers show that our method obtains impressive results and sets up a new state-of-the-art for multi-view tracking.

3.2 Related Work

The proposed work is closely related to the following research streams in computer vision and artificial intelligence.

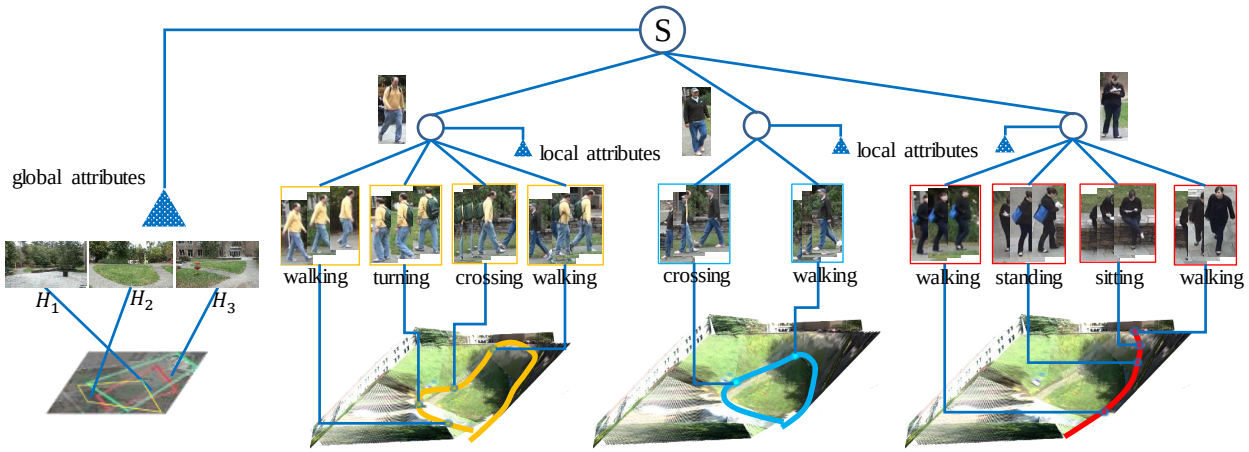


Figure 3.2: Illustration of Spatial and Temporal Attributed Parse Graph (ST-APG). The scene S is generated by 3D reconstruction and associated with certain global attributes (*e.g.*, homograph $H_1, \ldots, H_n$), and can be decomposed into trajectories belonging to different people. Each trajectory consists of multiple tracklets and is leveraged with local attributes (*i.e.*, blue triangles and words under tracklets).

Multi-view object tracking, like single-view tracking, is often formulated as a data association problem across cameras. A major question is to find cross-view correspondence at either pixel level [SZS03] or region-level [KS06, ALD11] or object-level [XLZ13, XMH14]. Typical data association methods are developed based on integer programming [JFL07], network flow [WHH09, BFT11], marked point process [UB11], multi-commodity network [SBF13], and multi-view SVM [ZYS15]. Notably, Porway and Zhu [PZ11] first introduced a cluster sampling method to explore both positive and negative relationships between samples, and Liu *et al.* [LLJ13] integrated a similar idea with motion information to construct a spatial-temporal graph for single-view tracking. In this chapter, we extend these two methods to further explore appearance, geometry, motion and pose/action relations between people tracklets in multi-view tracking.

Joint video parsing for solving multiple tasks simultaneously has been approved to be an effective way for boosting the performance of individual objectives. Wei *et al.* [WZZ17] presented a probabilistic framework for joint event, recognition, and ob-

ject localization. Shu *et al.* [SXR15] proposed to jointly infer groups, events, and human roles in aerial videos. Nie *et al.* [NXZ15] used human poses to improve action recognition. Park and Zhu used an stochastic grammar to jointly estimate human attributes, parts and poses [PNZ15]. Weng and Fu [WF11] utilized trajectories and key pose recognition to improve human action recognition. Yao *et al.* [YGF11] employed pose estimation to enhance human action recognition. Kuo and Nevatia [KN10] studied how person identity recognition can help multi-person tracking. In this chapter, we follow the same methodology to leverage semantic human attributes, including orientations, poses, and actions, to narrow the search space in cross-view data association.

Contributions. In comparison with previous methods, the contributions of this work is three-fold: i) a unified probabilistic framework of cross-view people tracking that can leverage multiple semantic attributes; ii) an efficient stochastic inference algorithm that can explore both positive and negative constraints between tracklets; iii) a comprehensive video benchmark regarding people's daily life which fosters research in this direction.z

3.3 Spatio-temporal Attributed Parse Graph

In a common multi-view setting, activities in a scene S are captured by multiple cameras $\{C_1, C_2, \ldots, C_n\}$ with overlapping field of view (FOV). Videos from these cameras are synchronized in time. Given such data, our goal is to discover the trajectories Γ of every person within the scene, that is,

$$\Gamma = \{\Gamma_i : i = 1, ..., K\}, \tag{3.1}$$

where K indicates the total number of people appearing in the scene over a time period.

We use tracklets (*i.e.*, trajectory fragments) as the basic unit. Tracklet is regarded as a mid-level representation to reduce the computation complexity, similar to super-pixels/voxels in segmentation. A tracklet τ consists of a short sequence of object bounding boxes, which can be denoted as

$$\tau = \{(b_k, t_k) : \ k = 1, 2, \ldots, |\tau|\}, \tag{3.2}$$

where b_k indicates the bounding box and t_k the corresponding frame number. Normally, the duration of the tracklet is short (less than 300 frames, usually 50-200 frames) and the person identity and motion within the tracklet is consistent.

Given a tracklet set $\Gamma = \{\tau_j, j = 1, 2, \ldots, N\}$, we can re-write the scene-center trajectory of a person Γ_i as

$$\Gamma_i = \{\, \tau_j : l(\tau_j) = l_i,\ j = 1, 2, \ldots, N \,\}, \tag{3.3}$$

where K indicates the total number of existing people in the scene. Each tracklet τ_j will be assigned with a label $l_i \in \{0, 1, \ldots, K\}$, which can be regarded as the person ID which it belongs to. We also add $l_i = 0$ to denote this tracklet belongs to background.

Therefore, the problem of multi-view tracking can be formulated as a tracklet grouping problem, *i.e.* clustering tracklets of the same person into scene-centered trajectories. We further associate these tracklets with attributes and represent the scene as a Spatio-temporal Attributed Parse Graph (ST-APG) M, as illustrated in Fig. 3.2. A ST-APG consists of four components:

$$M = (\, S,\, X(S),\, \Gamma,\, X(\Gamma) \,), \tag{3.4}$$

where $X(S)$ denotes the global attributes (*i.e.*, homographs $\{H_1, H_2, \ldots, H_n\}$ for each camera $\{C_1, C_2, \ldots, C_n\}$, $X(\Gamma)$ denotes the semantic attributes for tracklets. Therefore, solving multi-view people tracking is equivalent to finding the optimal ST-APG.

3.3.1 Semantic Attributes

Besides the identity label $l(\cdot)$, a tracklet τ_i is enriched with four kinds of attributes:

$$x(\tau_i) = (l(\tau_i), f(\tau_i),\, h(\tau_i),\, \vec{v}_i,\, \{a_{i,k}\}_{k=1}^{|\tau_i|}), \tag{3.5}$$

where $f(\tau_i)$ denotes the appearance attribute, $h(\tau_i)$ denotes the geometry attribute, $\vec{v}_i$ denotes the motion attribute of tracklet τ_i and $a_{i,k}$ the pose/action attribute at time $t_{i,k}$, *i.e.*, the k-th frame of tracklet τ_i.

Similar to the literature, we define the appearance attribute $f(\tau_i)$ as a feature descriptor, which implicitly models the visual evidence, *e.g.*, clothing, face, fair of a person. We also

define the geometry attribute $h(\tau_i)$ as the 2D object bounding boxes and projected footprints on the 3D ground plane. Besides appearance and geometry attributes, we further leverage two kinds of human semantic attributes to specifically handle the task of people tracking.

Motion Attributes. We assume the facing direction of a person is same as his/her motion direction. The average speed $\vec{v}_i$ is computed for each tracklet τ_i. However, 2D view-based motion not only suffers from the scale problem, but also is useless for cross-view comparisons. We thus transform the 2D view-based motion into the 3D real motion. Given the camera calibration, the foot point of each 2D bounding box is calculated and projected back onto the 3D ground. The speed and facing direction are thus computed and regarded as the motion attributes.

Pose/Action Attributes. To describe the actions and poses a_i of an individual, we apply a DCNN to categorize the classical human pose/action variations. We use the PASCAL VOC 2012 action dataset, augmented by our own collected images. The training set has 7 categories, including standing, sitting, bending, walking, running, riding bike, skateboarding, which covers people's common type of actions/poses in daily activities. The collected training set consists 5000 images. We thus fine tune a 7 layer CaffeNet, with 5 convolutional layers, 2 max-pooling layers, 3 fully-connected layers. The final output give us a 7d human pose/action confidence score and can be regarded as the local attribute probability $p(a_i)$.

Besides the unary pose/action confidence, we further learn a binary temporal consistency table $T(a_i, a_j)$ to describe the possible transitions between two successive pose/action attributes. The consistency table is learned from our newly collected multi-view dataset and apply in all experiments. There are around 1000 training samples in total. In learning, we initialize impossible transitions (*e.g.*, bending→running, sitting→riding) as 0 and else as 0.05.

3.4 Bayesian Formulation

According to Bayes rule, M can be solved by maximizing a posterior (MAP), that is,

$$\begin{aligned} M^* &= \arg\max_M p(M|\Gamma;\theta) \\ &= \arg\max_M \frac{1}{Z}\exp\{-\mathcal{E}(\Gamma|M;\theta) - \mathcal{E}(M;\theta)\}, \end{aligned} \tag{3.6}$$

where θ indicates the model parameters.

Likelihood term $\mathcal{E}(\Gamma|M;\theta)$ measures how well the observed data (video bundle) satisfies a certain object trajectory. Assuming the likelihood of each bundle is calculated independently given the partition, then $\mathcal{E}(\Gamma|M;\theta)$ can be written as

$$\mathcal{E}(\Gamma|M) = \sum_{\tau_i\in\Gamma} \mathcal{E}(\tau_i|M;\theta). \tag{3.7}$$

Each term $\mathcal{E}(\tau_i|M;\theta)$ measures how the tracklet τ_i discriminates from the background. Therefore we treat this term as the constraint of itself being consistent with a foreground trajectory of a certain person. We estimate $\mathcal{E}(\tau_i|M;\theta)$ as a Markov chain structure, where the unary term $\mathcal{E}(a_i)$ is the attributes confidence probability, and the pairwise term $\mathcal{E}(a_i, a_j)$ is the attribute consistency in two successive frames, that is

$$\mathcal{E}(\tau_i|M;\theta) = \sum_{k=1}^{|\tau_i|} \mathcal{E}(a_{i,k}) + \sum_{k=1}^{|\tau_i|-1} \mathcal{E}(a_{i,k},\, a_{i,k+1}). \tag{3.8}$$

Note the motion information is trivial for successive frames and we thus ignore this part.

In this chapter, we utilize **prior term** $\mathcal{E}(M;\theta)$ imposes constraints on people trajectories and their interactions. To do so, we develop four types of relations between two tracklets, as illustrated in Fig. 3.3. Given two tracklets, we consider both traditional visual relations (*i.e.*, appearance and geometry) and leveraged semantic attribute relations (*i.e.*, motion and pose/action).

Appearance similarity. This constraint assumes that the same person should share similar appearance across time and cameras. We adopt the appearance measurement proposed in [XLL16], which basically uses a DCNN as codebook and encodes human body

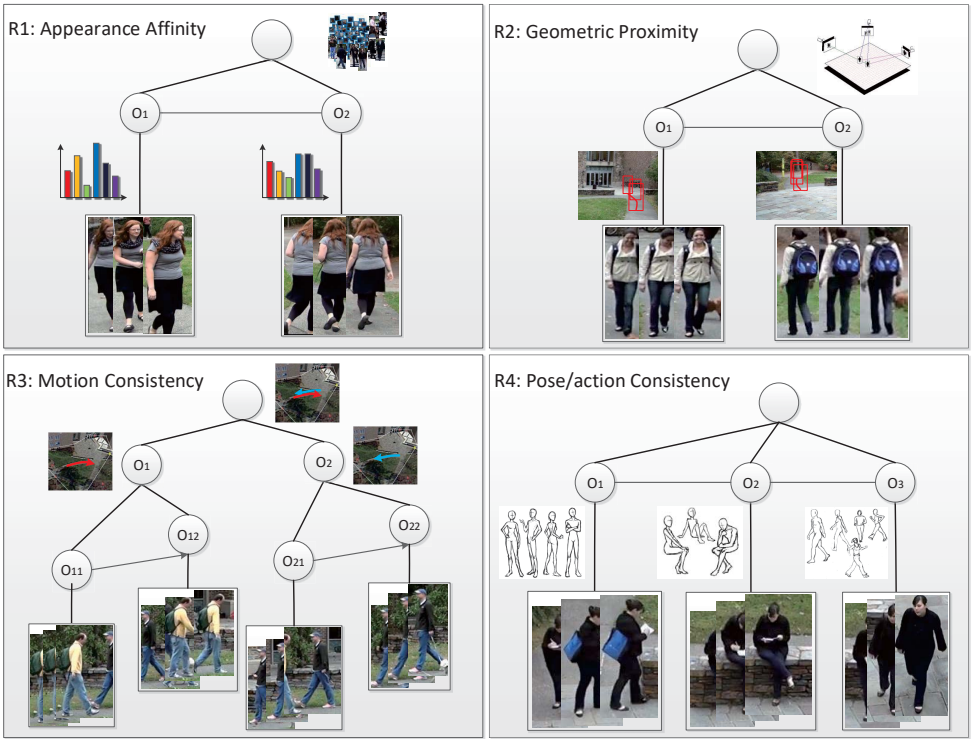


Figure 3.3: An illustration of four kinds of relations we utilize in this chapter.

appearance as a 1000d feature vector. We measure the appearance similarity rule by

$$\mathcal{E}_e^{app}(\tau_i, \tau_j) = \sum_{1}^{|\tau_i|} \sum_{1}^{|\tau_j|} \frac{\|f(\tau_i) - f(\tau_j)\|_2}{|\tau_i| \cdot |\tau_j|}, \tag{3.9}$$

where $f(\tau_i)$ denotes the encoded feature vector of τ_i.

Geometric proximity measures how far two tracklets are located. We project the foot points of two tracklets onto the scene 3D ground plane using the given 2D to 3D homograph, and then compute the proximity of two tracklets as

$$\mathcal{E}_e^{geo}(\tau_i, \tau_j) = D(h(\tau_i), h(\tau_j)). \tag{3.10}$$

$D(\cdot, \cdot)$ denotes the averaged Euclidean distance between foot points of τ_i and τ_j over all overlapped frames.

Motion consistency. Given two proximate tracklets, the motion direction actually provides a solid evidence to show whether these tracklets belong to a same person or two

persons crossing each other. Therefore, we can compute the angle between two motion directions. that is,

$$\mathcal{E}_e^{mov}(\tau_i, \tau_j) \;=\; \arccos \frac{\vec{v}_i \cdot \vec{v}_j}{|\vec{v}_i||\vec{v}_i|}. \tag{3.11}$$

If the angle is large, this probably indicates that two persons are moving in different directions.

Pose/action consistency. Noticing the pose/action of a same person across different views should also be consistent, we thus use the learned temporal consistency table $p(a_i, a_j)$ to describe the consistency between two actions/poses. The rule is computed as

$$\mathcal{E}_e^{act}(\tau_i, \tau_j) \;=\; \sum_{t_m \in \{t_{i,k}\} \cap \{t_{j,k}\}} \mathcal{E}(a_{i,m},\, a_{j,m}). \tag{3.12}$$

Note that we only consider such relation among overlapped frames of two tracklets τ_i and τ_j.

We further introduce an adjacency graph $G =< \Gamma, E >$ to describe connections among tracklets. Each tracklet $\tau_i \in \Gamma$ is treated as a graph vertex and each edge $e_{ij} =< \tau_i, \tau_j > \in E$ describes the relation between two adjacent (neighboring) tracklets τ_i and τ_j. In this chapter, two tracklets τ_i and τ_j are regarded as neighbors $\tau_i \in nbr(\tau_j)$ if only their temporal difference is no more than $\Delta_t = 30$ frames and no far than $\Delta_d = 5m$.

We regard edges generated by four types of constraints as cooperative edges E^+. The edge set E is further extended with conflicting edges E^-, that is, $E = E^- \cup E^+$. We enforce hard constraints to guarantee that i) two tracklets from the same view with temporal overlap will never be grouped together; ii) two adjacent tracklets with same identities will never have impossible pose/action transitions defined in temporal consistency table $T(a_i, a_j)$. Both types of relationships are utilized to help us group tracklets with similar characteristics together and with conflicting characteristics being dispelled.

Therefore, we can decompose the **prior term** $\mathcal{E}(M; \theta)$ into pairwise potentials between every two adjacent tracklets within G, that is,

$$\mathcal{E}(M; \theta) = \sum_{l_i = l_j, e_{ij} \in E^+} \mathcal{E}_e^+(\tau_i, \tau_j) + \sum_{l_i = l_j, e_{ij} \in E^-} \mathcal{E}_e^-(\tau_i, \tau_j), \tag{3.13}$$

where p_e^+ and p_e^- are the corresponding cooperative and conflicting edge probability defined above.

3.5 Inference

Given a scenario, finding the optimal ST-APG includes two sub-tasks: (1) partitioning tracklet set Γ into trajectories belonging to different people Γ_i, (2) inferring the semantic human attributes for each person. Noticing that sub-task (1) is a combinatorial optimization problem and jointly solving these two sub-tasks is infeasible, we therefore propose an inference algorithm to optimize these two sub-tasks iteratively. The inference process is illustrated in Fig. 3.4. For sub-task (1), we apply a stochastic clustering algorithm, *i.e.*, Swensden-Wang Cuts [BZ05], which could efficiently and effectively traverses through the grouping solution space. For sub-task (2), given grouped tracklets, we can use Dynamic Programming to update the semantic attributes of tracklets within every group (*i.e.*, person trajectory). These two algorithms are are iterated one after another until convergence.

3.5.1 Associating Tracklets by Stochastic Clustering

Traditional sampling algorithms usually suffer from the efficiency issues. On the contrary, cluster sampling algorithm overcomes this issue by randomly grouping clusters and resampling cluster as a whole. The algorithm consists of two steps:

(I) **Generating cluster set**. Given an adjacency graph $G =< \Gamma, E >$ and the current state M, we regard every edge e_{ij} in this graph as a switch. We turn on every edge e_{ij} probabilistically with its edge probability p_e. Afterwards, we regard candidates connected by "on" positive edges as a cluster V_{cc} and collect separate clusters to produce the cluster set.

(II) **Relabeling cluster set**. We randomly choose a cluster V_{cc} from the produced cluster set and randomly change the label of the selected cluster, which generates a new state M'. This is essentially changing the ID of a group of tracklets. This group of tracklets

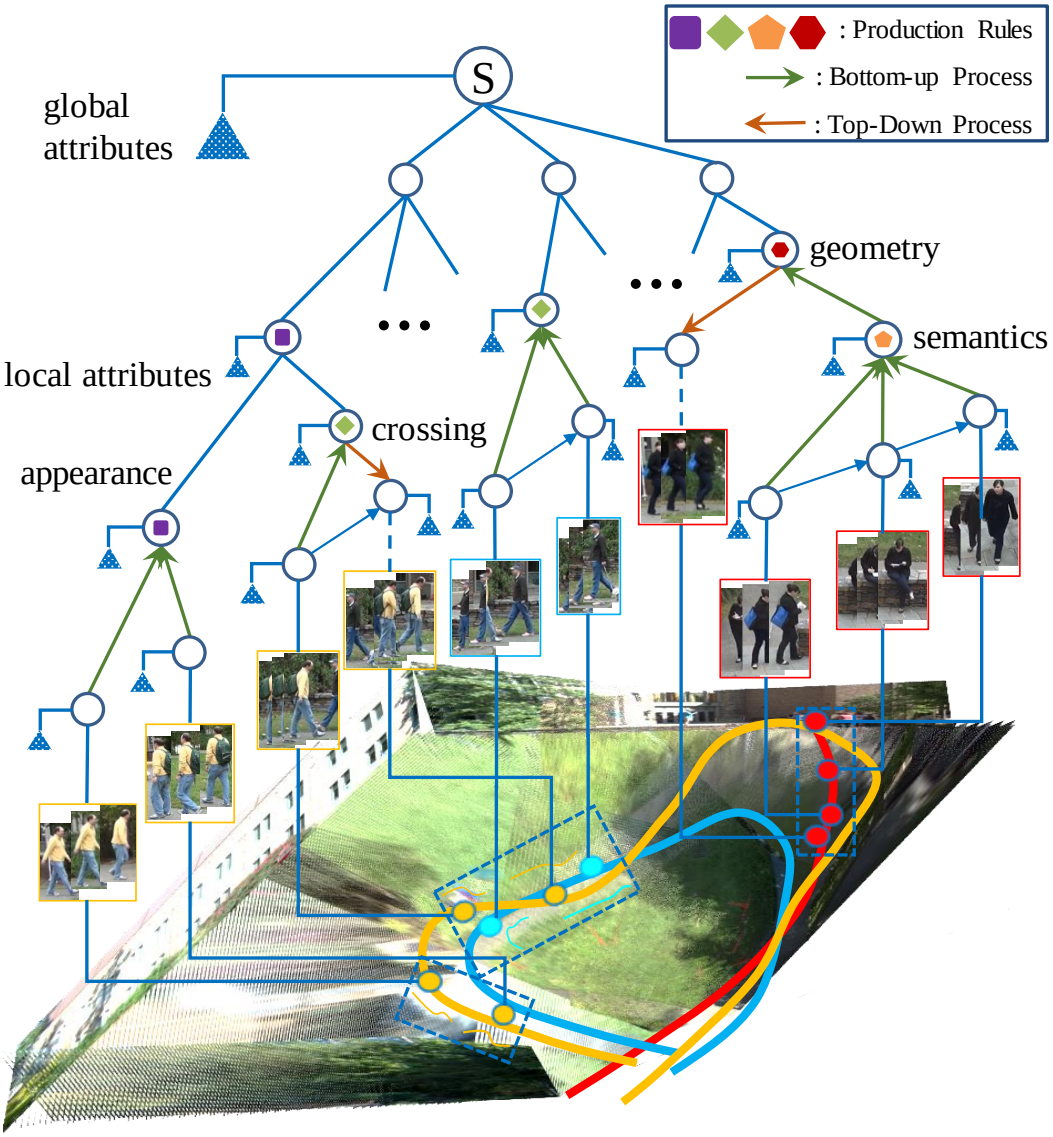


Figure 3.4: Illustration of our inference process. We parse the optimal parse graph in a joint bottom-up and top-down process. At each iteration, one of the five composition criteria is randomly selected and applied to update the current parse graph. The inference process either augments the current parse graph with bottom-up proposals or recover missing trajectory fragments from top-down guidance.

Algorithm 1: Sketch of our inference algorithm

Input: Tracklet set Γ, global attributes $X(S)$

Output: Spatio-temporal Attributed Parse Graph M

Assign semantic attributes for each tracklet τ_i by DP ;

Construct adjacency graph G by computing cooperative and conflicting relations among Γ ;

Initialize $K = |\Gamma|$, $l_i = i$;

repeat

  Generate a cluster V_{cc};

  Randomly relabel cluster V_{cc} and obtain a new state M' ;

  Accept the new state with acceptance rate $\alpha(M \to M')$;

  Re-run DP on each new trajectory to update semantic attributes ;

until *convergence*;

can either be merged into another trajectory, or set to background noises. Following the Markov chain Monte Carlo principal, we accept the transition from state M to new state M' with a rate $\alpha(\cdot)$ defined by the Metropolis-Hastings method [MRR53]:

$$\alpha(M \to M') = \min(1, \frac{p(M' \to M) \cdot p(M'|\Gamma)}{p(M \to M') \cdot p(M|\Gamma)}), \tag{3.14}$$

where $p(M' \to M)$ and $p(M \to M')$ are the state transition probability, $p(M'|\Gamma)$ and $p(M|\Gamma)$ the posterior defined in Equation.(3.6). This guarantees the stochastic algorithm can find better states and obtains reversible jumps between any two states.

Following instructions in [BZ05], the transition probability ratio can be calculated as

$$\frac{p(M' \to M)}{p(M \to M')} \propto \frac{p(V_{cc}|M')}{p(V_{cc}|M)} \propto \frac{\prod_{e \in E^{\star}_{M'}}(1 - p_e)}{\prod_{e \in E^{\star}_{M}}(1 - p_e)}, \tag{3.15}$$

where $E^{\star}$ denotes the sets of edges being turned off around V_{cc}, that is,

$$E^{\star} = \{e \in E : \tau_i \in V_{cc}, \tau_j \notin V_{cc}, l(\tau_i) = l(\tau_j)\}. \tag{3.16}$$

3.5.2 Assigning Semantic Attributes by DP

Given a trajectory, we first find trajectory gaps (*i.e.*, no bounding box presented) below 60 frames, we then apply a linear interpolation to fill-in the missing bounding boxes.

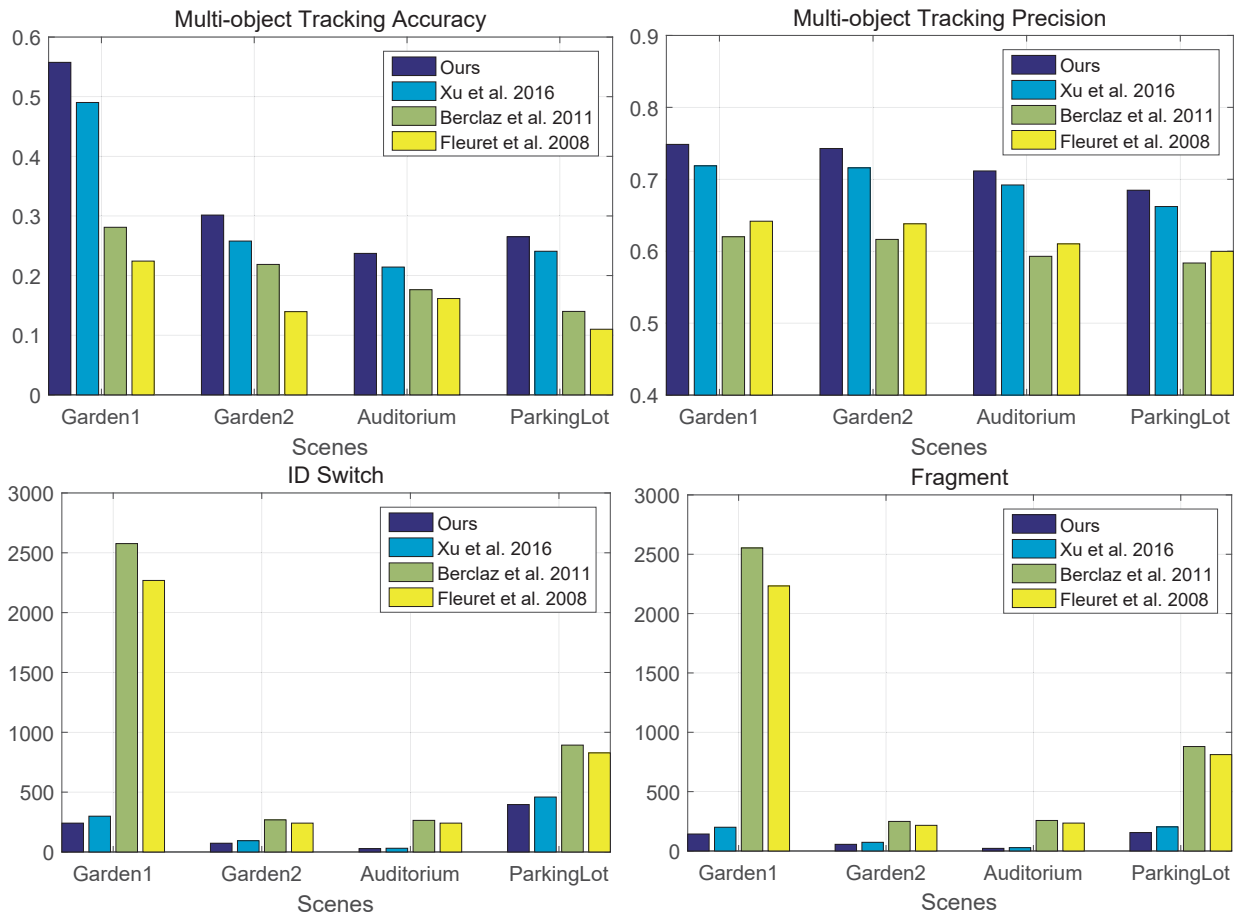


Figure 3.5: Comparison charts of major metrics on CAMPUS datasets.

After that, assigning the semantic attribute is similar to estimating the likelihood term $p(\tau_i|M)$. The whole trajectory is also treated as a Markov chain structure. We therefore apply the standard factor graph belief propagation (sum-product) algorithm to infer the semantic human attributes of a trajectory.

A short summary of our proposed inference algorithm is shown in Algorithm 1.

3.6 Experiment

To evaluate the proposed method, we compare with other state-of-the-arts on two datasets:

(1) **CAMPUS dataset** [XLL16]. This is a newly published dataset targeting multi-view

tracking. There are four sequences, *i.e.*, two gardens, parking lot, auditorium, each of which is shot by 3-4 1080P cameras. The recorded videos are 3-4 minutes long and with 30 FPS. This dataset contains people with huge pose variations and lots of actions (*e.g.*, running, riding bikes, sitting), providing richer semantic human attributes.

(2) **PPL-DA dataset**. We collect a new dataset aiming to cover people's daily activities. The new dataset consists of 3 public facilities: foot court, office reception, plaza. The scenes are recorded with 4 GoPro cameras, mounted on around 1.5 meters high tripods. The produced videos are also around 4 minutes long and in 1080P high quality. We further annotate the trajectories of every person inside the scene with cross-view consistent ID.

For both datasets, we incorporate 10% of the videos as augmented training set and the rest as testing set. The augmented data, together external dataset described in previous section, helps us learn the action labels and transitions. The learning process is only done once and applied to both datasets. All parameters are fixed in the experiment. We use Fast R-CNN [Gir15] to generate people's bounding boxes. The pruning threshold is set to 0.3. We apply Sequential Shortest Path (SSP) [PRF11] to initialize tracklets. The sampling is set to finish after 1000 iterations, which achieves decent results.

The proposed approach is compared with 3 state-of-the-arts methods: Probabilistic Occupancy Map (POM) [FBL08], K-Shortest Path (KSP) [BFT11] and Hierarchical Trajectory Composition (HTC) [XLL16]. The public implementations of POM and KSP are adopted. We further implement HTC on our own using the default parameters. For quantitative results, we apply multi-object tracking accuracy (TA), multi-object tracking precision (TP), mostly tracked/lost trajectories (MT/ML), identity switches (IDSW) and trajectory fragments (FRG). DA, DP, TA and TP mainly measure the percentage of true positives while MT/ML, IDSW and FRG mainly measure the completeness and identity consistency of the result trajectories. A higher value means better for TA, TP and MT while a lower value means better for ML, IDSW and FRG.

We report quantitative results on CAMPUS datasets in Fig. 3.5 and on PPL-DA dataset in Table 3.1. From the results, the proposed method obtains a significant improvement over

Seq-Court	TA(%)	TP(%)	MT(%)	ML(%)	IDSW	FRG
Our-full	**34.47**	**72.38**	**18.52**	**25.93**	**79**	**55**
Our-1	26.82	70.23	11.11	33.33	114	90
HTC	29.51	71.87	14.81	25.93	91	77
KSP	24.72	64.40	0.00	44.44	318	291
POM	22.26	65.39	0.00	51.85	296	269
Seq-Office	TA(%)	TP(%)	MT(%)	ML(%)	IDSW	FRG
Our-full	**47.38**	**73.70**	**42.86**	**0.00**	**45**	**31**
Our-1	39.79	68.99	28.57	0.00	71	64
HTC	41.17	70.65	28.57	0.00	66	59
KSP	39.62	58.01	28.57	0.00	83	76
POM	36.86	58.77	28.57	0.00	89	82
Seq-Plaza	TA(%)	TP(%)	MT(%)	ML(%)	IDSW	FRG
Our-full	**25.18**	**67.10**	**16.28**	**11.63**	**165**	**133**
Our-1	20.59	65.15	11.63	18.60	244	199
HTC	23.11	66.24	11.63	18.60	202	178
KSP	17.30	57.49	6.98	27.91	356	311
POM	16.71	57.87	4.65	32.56	339	295

Table 3.1: Quantitative results and comparisons on PPL-DA dataset. Our-1 and Our-full are two variants of the proposed framework. See text for detailed explanations.

Figure 3.6: Sampled qualitative results of our proposed method on CAMPUS and PPL-DA datasets.

the competing methods. An interesting observation is that tracking by associating bounding boxes (*i.e.*, KSP, POM) yields much worse results than tracking by associating tracklets (*i.e.*, Ours, HTC).

We set up a baseline **Our-1** to further analyze the effectiveness of leveraged semantic attributes. **Our-1** only uses appearance and geometry information for multi-view tracking. From the results we can observe that when people with various actions present, the proposed method is able to exploit this visual information and significantly improves the tracking results. However, when lack of such variations (*e.g.*, Auditorium, ParkingLot, Plaza), the proposed method can only utilize people motion information and obtains slightly better results. some qualitative results are visualized in Fig. 3.6.

We implement the proposed method with MATLAB and test it on a workstation with I7 3.0GHz CPU, 32GB memory and GTX1080 GPU. For a scene shot by 4 cameras and lasting for around 4 minutes, our algorithm obtains 5 frames per second on average. With further code optimization and batch-based data parallelization, our proposed method can run in real-time.

3.7 Summary

In this chapter, we propose a novel multi-view multi-object tracking approach. Tracking people is leveraged with rich semantic attributes and therefore the association of tracklets are further constrained. By incorporating the motion attributes, pose attributes and action attributes, our algorithm outperforms the competing methods only using appearance and geometry information. In the future, we will continue to explore more high-level information (*e.g.*, people interactions, group information) among tracklets and more efficient inference algorithms.

CHAPTER 4

3D Scene Understanding by Scene-centric Joint Parsing

4.1 Introduction

During the past decades, remarkable progress has been made in many vision tasks, *e.g.*, image classification, object detection, pose estimation. Recently, more comprehensive visual tasks probe deeper understanding of visual scenes under interactive and multi-modality settings, such as visual Turing tests [GGH15, QWL15] and visual question answering [AAL15]. In addition to discriminative tasks focusing on binary or categorical predictions, emerging research involves representing fine-grained relationships in visual scenes [KZG17, ABY16] and unfolding semantic structures in contexts including caption or description generation [YYL10], and question answering [TML14, ZGB16].

In this chapter, we present a framework for uncovering the semantic structure of scenes in a cross-view camera network. The central requirement is to resolve ambiguity and establish cross-reference among information from multiple cameras. Unlike images and videos shot from single static point of view, cross-view settings embed rich physical and geometry constraints due to the overlap between fields of views. While multi-camera setups are common in real-word surveillance systems, large-scale cross-view activity dataset are not available due to privacy and security reasons. This makes data-demanding deep learning approaches infeasible.

Our joint parsing framework computes a hierarchy of spatio-temporal parse graphs by establishing cross-reference of entities among different views and inferring their semantic

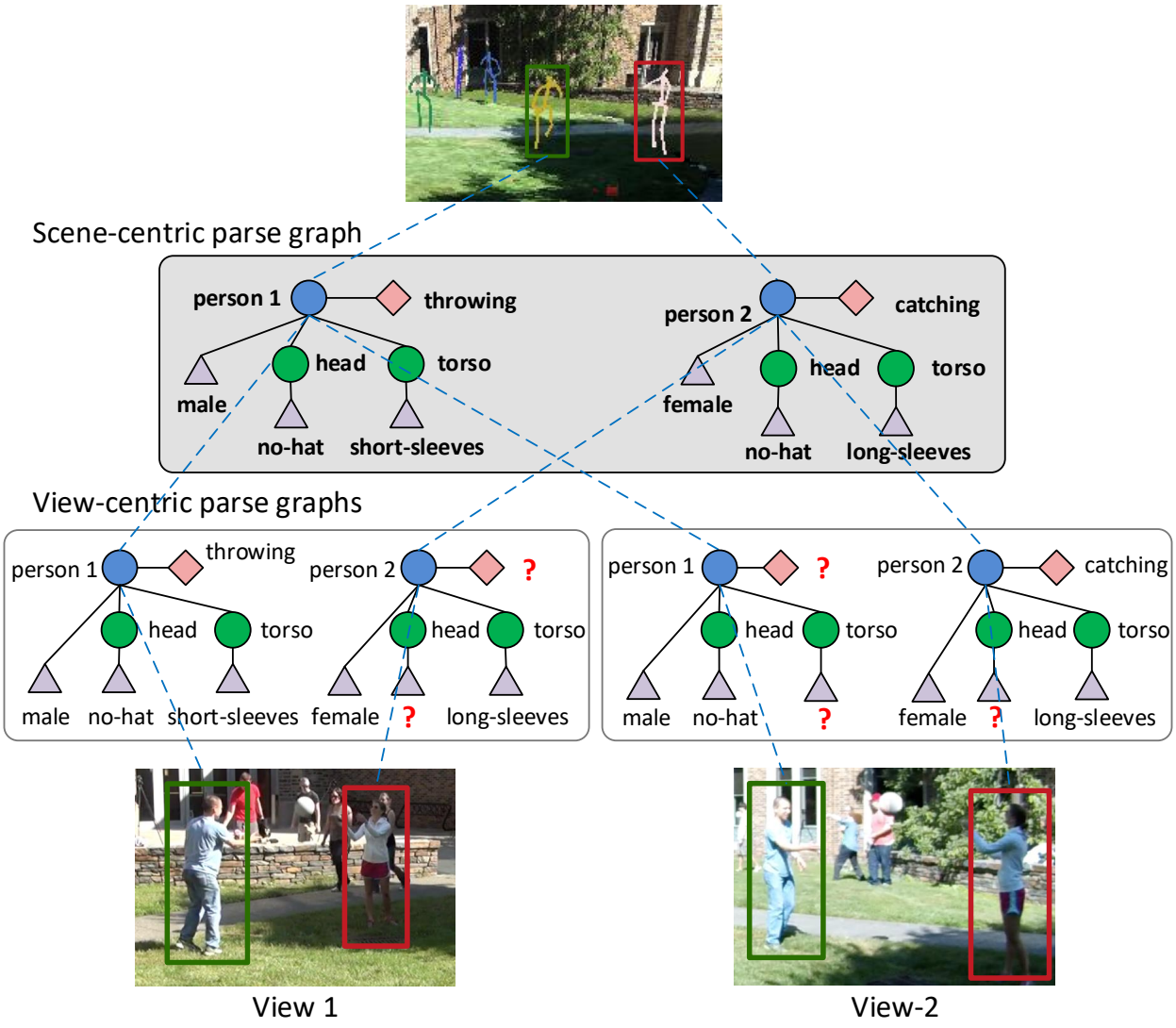


Figure 4.1: An example of the spatio-temporal semantic parse graph hierarchy in a visual scene captured by two cameras.

attributes from a scene-centric perspective. For example, Fig. 4.1 shows a parse graph hierarchy that describes a scene where two people are playing a ball. In the first view, person 2's action is not grounded because of the cluttered background, while it is detected in the second view. Each view-centric parse graph contains local recognition decisions in an individual view, and the scene centric parse graph summaries a comprehensive understanding of the scene with coherent knowledge.

The structure of each individual parse graph fragment is induced by an ontology graph that regulates the domain of interests. A parse graph hierarchy is used to represent the correspondence of entities between the multiple views and the scene. We use a probabilistic model to incorporate various constraints on the parse graph hierarchy and formulate the joint parsing as a MAP inference problem. A MCMC sampling algorithm and a dynamic programming algorithm are used to explore the joint space of scene-centric and view-centric interpretations and to optimize for the optimal solutions. Quantitative experiments show that scene-centric parse graphs outperforms the initial view-centric proposals.

Contributions. The contributions of this work are three-fold: (i) a unified hierarchical parse graph representation for cross-view person, action, and attributes recognition; (ii) a stochastic inference algorithm that explores the joint space of scene-centric and view-centric interpretations efficiently starting with initial proposals; (iii) a joint parse graph hierarchy that is an interpretable representation for scene and events.

4.2 Related Work

Our work is closely related to three research areas in computer vision and artificial intelligence.

Multi-view video analytics. Typical multi-view visual analytics tasks include object detection [LS10, UB11], cross-view tracking [BFT11, LPR12, XLL16, XLQ17], action recognition [WNX14], person re-identification [XLZ13, XMH14] and 3D reconstruction [HWR13]. While heuristics such as appearances and motion consistency constraints have been used to regularize the solution space, these methods focus on a specific multi-view vision task

whereas we aim to propose a general framework to jointly resolve a wide variety of tasks.

Semantic representations. Semantic and expressive representations have been developed for various vision tasks, *e.g.*, image parsing [HZ09], 3D scene reconstruction [PBH13, LZZ14], human-object interaction [KS16], pose and attribute estimation [WZZ17]. In this chapter, our representation also falls into this category. The difference is that our model is defined upon cross-view spatio-temporal domain and is able to incorporate a variety of tasks.

Interpretability. Automated generation of explanations regarding predictions has a long and rich history in artificial intelligence. Explanation systems have been developed for a wide range of applications, including simulator actions [VFM04, LCV05, CLV06], robot movements [LCC12], and object recognition in images [BM14, HAR16]. Most of these approaches are rule-based and suffer from generalization across different domains. Recent methods including [RSG16] use proxy models or data to interpret black box models, while our scene-centric parse graphs are explicit representations of the knowledge by definition.

4.3 Representation

A scene-centric spatio-temporal parse graph represents humans, their actions and attributes, interaction with other objects captured by a network of cameras. We will first introduce the concept of ontology graph as domain definitions, then we will describe parse graphs and parse graph hierarchy as view-centric and scene-centric representations respectively.

Ontology graph. To define the scope of our representation on scenes and events, an ontology is used to describe a set of plausible objects, actions and attributes. We define an ontology as a graph that contains nodes representing objects, parts, actions, attributes respectively and edges representing the relationships between nodes. Specifically, every object and part node is a concrete type of object that can be detected in videos. Edges between object and part nodes encodes "part-of" relationships. Action and attribute nodes connected to an object or part node represent plausible actions and appearance attributes the object can take. For example, Fig. 4.2 shows an ontology graph that describes a domain including

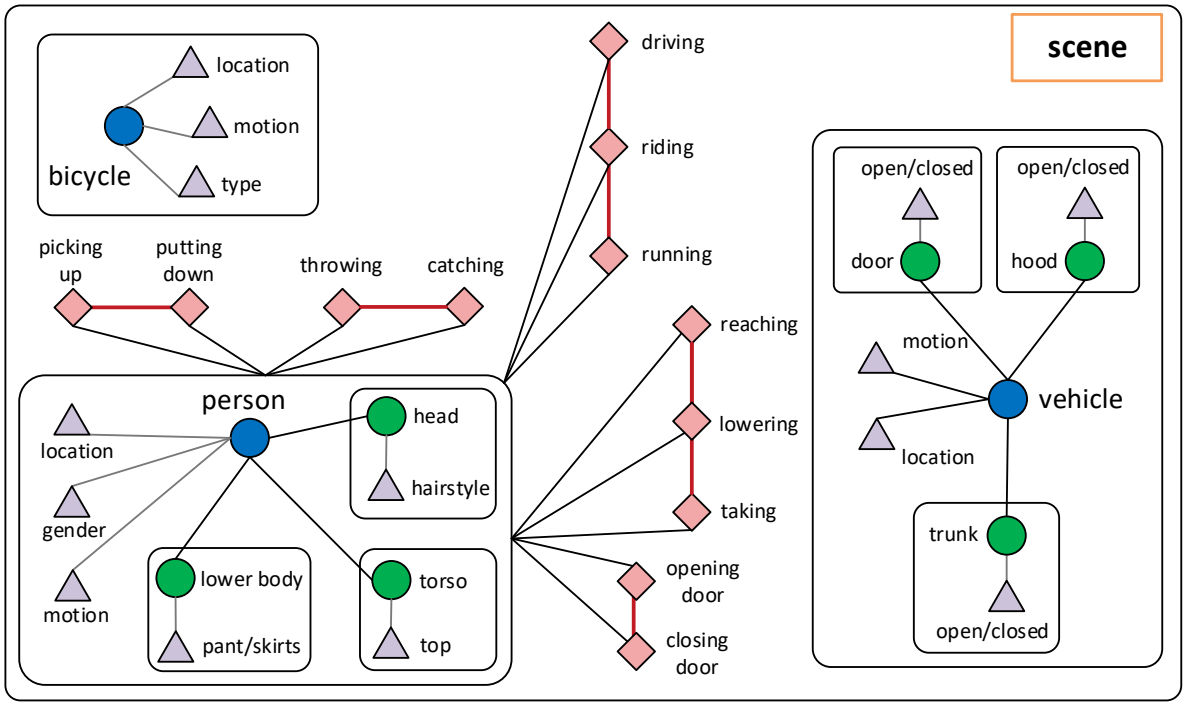


Figure 4.2: An illustration of the proposed ontology graph describing objects, parts, actions and attributes.

people, vehicles, bicycles. An object can be decomposed into parts (*i.e.*, green nodes), and enriched with actions (*i.e.*, pink nodes) and attributes (*i.e.*, purple diamonds). The red edges among action nodes denote their incompatibility. The ontology graph can be considered a compact AOG [LZZ14, WZZ17] without the compositional relationships and event hierarchy. In this chapter, we focus on a restricted domain inspired by [QWL15], while larger ontology graphs can be easily derived from large-scale visual relationship datasets such as [KZG17] and open-domain knowledge bases such as [LS04].

Parse graphs. While an ontology describes plausible elements, only a subset of these concepts can be true for a given instance at a given time. For example, a person cannot be both "standing" and "sitting" at the same time, while both are plausible actions that a person can take. To distinguish plausible facts and satisfied facts, we say a node is *grounded* when it is associated with data. Therefore, a subgraph of the ontology graph that only contains grounded nodes can be used to represent a specific *instance* (*e.g.* a specific person) at a specific time. In this chapter, we refer to such subgraphs as *parse graphs*.

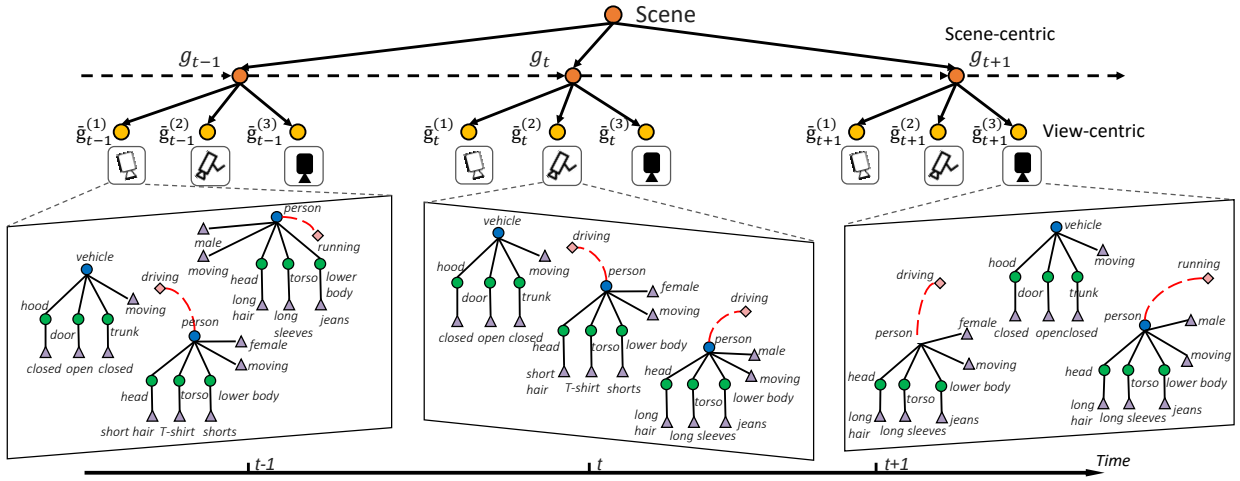


Figure 4.3: The proposed spatio-temporal parse graph hierarchy. (Better viewed electronically and zoomed).

Parse graph hierarchy. In cross-view setups, since each view only captures an incomplete set of facts in a scene, we use a spatio-temporal hierarchy of parse graphs to represent the collective knowledge of the scene and all the individual views. To be concrete, a view-centric parse graph $\tilde{g}$ contains nodes grounded to a video sequence captured by an individual camera, whereas a scene-centric parse graph g is an aggregation of view-centric parse graphs and therefore reflects a global understanding of the scene. As illustrated in Fig. 4.3, for each time step t, the scene-centric parse graph g_t is connected with the corresponding view-centric parse graphs $\tilde{g}_t^{(i)}$ indexed by the views, and the scene-centric graphs are regarded as a Markov chain in the temporal sequence. In terms of notations, in this chapter we use a tilde notation to represent the view-centric concepts $\tilde{x}$ corresponding to scene-centric concepts x.

4.4 Probabilistic Formulation

Given the input frames from video sequences $I = \{I_t^{(i)}\}$ captured by a network of M cameras, the task of joint parsing is to infer the spatio-temporal parse graph hierarchy G

$$G =< \Phi, g, \tilde{g}^{(1)}, \tilde{g}^{(2)}, \ldots, \tilde{g}^{(M)} >, \tag{4.1}$$

where Φ is an object identity mapping between scene-centric parse graph g and view-centric parse graphs $\tilde{g}^{(i)}$ from camera i. Φ defines the structure of parse graph hierarchy. In this section, we discuss the formulation assuming a fixed structure, while defer the discussion of how to traverse the solution space to section 4.5.

We formulate the inference of parse graph hierarchy as a MAP inference problem in a posterior distribution $p(G|I)$ as follows

$$G^* = \arg\max_G p(I|G) \cdot p(G). \tag{4.2}$$

Likelihood. The likelihood term models the grounding of nodes in view-centric parse graphs to the input video sequences. Specifically,

$$\begin{aligned} p(I|G) &= \prod_{i=1}^{M}\prod_{t=1}^{T} p(I_t^{(i)}|\tilde{g}_t^{(i)}) \\ &= \prod_{i=1}^{M}\prod_{t=1}^{T}\prod_{v\in V(\tilde{g}_i^{(t)})} p(I(v)|v), \end{aligned} \tag{4.3}$$

where $\tilde{g}_t^{(i)}$ is the view-centric parse graph of camera i at time t and $V(\tilde{g}_t^{(i)})$ is the set of nodes in the parse graph. $p(I(v)|v)$ is the node likelihood for the concept represented by node v being grounded on the data fragment $I(v)$. In practice, this probability can be approximated by normalized detection and classifications scores [PRF11].

Prior. The prior term models the compatibility of scene-centric and view-centric parse graphs across time. We factorize the prior as

$$p(G) = p(g_1)\prod_{t=1}^{T-1} p(g_{t+1}|g_t)\prod_{i=1}^{M}\prod_{t=1}^{T} p(\tilde{g}_t^{(i)}|g_t), \tag{4.4}$$

where $p(g_1)$ is a prior distribution on parse graphs that regulates the combination of nodes, and $p(g_t|g_{t-1})$ is a transitions probability of scene-centric parse graphs across time. Both probability distributions are estimated from training sequences. $p(\tilde{g}_t^{(i)}|g_t)$ is defined as a Gibbs distribution that models the compatibility of scene-centric and view-centric parse

graphs in the hierarchy (we drop subscripts t and camera index i for brevity).

$$\begin{aligned} p(\tilde{g}|g) &= \frac{1}{Z}\exp\{-\mathcal{E}(g,\tilde{g})\} \\ &= \frac{1}{Z}\exp\{-w_1\mathcal{E}_S(g,\tilde{g}) - w_2\mathcal{E}_A(g,\tilde{g}) \\ &\qquad - w_3\mathcal{E}_{Act}(g,\tilde{g}) - w_4\mathcal{E}_{Attr}(g,\tilde{g})\}, \end{aligned} \tag{4.5}$$

where energy $\mathcal{E}(g,\tilde{g})$ is decomposed into four different terms described in detail in the subsection below. The weights are tuning parameters that can be learned via cross-validation. We consider view-centric parse graphs for videos from different cameras are independent conditioned on scene-centric parse graph under the assumption that all cameras have fixed and known locations.

4.4.1 Cross-view Compatibility

In this subsection, we describe the energy function $\mathcal{E}(g,\tilde{g})$ for regulating the compatibility between the occurrence of objects in the scene and an individual view from various aspects. Note that we use a tilde notation to represent the node correspondence in scene-centric and view-centric parse graphs (*i.e.*, for a node $v \in g$ in a scene-centric parse graph, we refer to the corresponding node in a view-centric parse graph as $\tilde{v}$).

Appearance similarity. For each object node in the parse graph, we keep an appearance descriptor. The appearance energy regulates the appearance similarity of object o in the scene-centric parse graph and $\tilde{o}$ in the view-centric parse graphs.

$$\mathcal{E}_A(g,\tilde{g}) = \sum_{o\in g} ||(\phi(o) - \phi(\tilde{o})||_2, \tag{4.6}$$

where $\phi(\cdot)$ is the appearance feature vector of the object. At the view-level, this feature vector can be extracted by pre-trained convolutional neural networks; at the scene level, we use a mean pooling of view-centric features.

Spatial consistency. At each time point, every object in a scene has a fixed physical location in the world coordinate system while appears on the image plane of each camera according to the camera projection. For each object node in the parse graph hierarchy, we

keep a scene-centric location $s(o)$ for each object o in scene-centric parse graphs and a view-centric location $s(\tilde{o})$ on the image plane in view-centric parse graphs. The following energy is defined to enforce the spatial consistency:

$$\mathcal{E}_S(g, \tilde{g}) = \sum_{o \in g} ||s(o) - h(s(\tilde{o}))||_2, \tag{4.7}$$

where $h(\cdot)$ is a perspective transform that maps a person's view-centric foot point coordinates to the world coordinates on the ground plane of the scene with the camera homography, which can be obtained via the intrinsic and extrinsic camera parameters.

Action compatibility. Among action and object part nodes, scene-centric human action predictions shall agree with the human pose observed in individual views from different viewing angles:

$$\mathcal{E}_{Act}(g, \tilde{g}) = \sum_{l \in g} -\log p(l|\tilde{p}), \tag{4.8}$$

where l is an action node in scene-centric parse graphs and $\tilde{p}$ are positions of all human parts in the view-centric parse graph. In practice, we separately train a action classifier that predicts action classes with joint positions of human parts and uses the classification score to approximate this probability.

Attribute consistency. In cross-view sequences, entities observed from multiple cameras shall have a consistent set of attributes. This energy term models the commonsense constraint that scene-centric human attributes shall agree with the observation in individual views:

$$\mathcal{E}_{Attr}(g, \tilde{g}) = \sum_{a \in g} \mathbf{1}(a \neq \tilde{a}) \cdot \xi, \tag{4.9}$$

where $\mathbf{1}(\cdot)$ is an indicator function and ξ is a constant energy penalty introduced when the two predictions mismatch.

4.5 Inference

The inference process consists of two sub-steps: (i) matching object nodes Φ in scene-centric and view-centric parse graphs (*i.e.* the structure of parse graph hierarchy) and (ii) estimating

optimal values of parse graphs $\{g, \tilde{g}^{(1)}, \ldots, \tilde{g}^{(M)}\}$.

The overall procedure is as follows: we first obtain view-centric objects, actions, and attributes proposals from pre-trained detectors on all video frames. This forms the initial view-centric predictions $\{\tilde{g}^{(1)}, \ldots, \tilde{g}^{(M)}\}$. Next we use a Markov Chain Monte Carlo (MCMC) sampling algorithm to optimize the parse graph structure Φ. Given a fixed parse graph hierarchy, variables within the scene-centric and view-centric parse graphs $\{g, \tilde{g}^{(1)}, \ldots, \tilde{g}^{(M)}\}$ can be efficiently estimated by a dynamic programming algorithm. These two steps are performed iteratively until convergence.

4.5.1 Inferring Parse Graph Hierarchy

We use a stochastic algorithm to traverse the solution space of the parse graph hierarchy Φ. To satisfy the detailed balance condition, we define three reversible operators $\Theta = \{\Theta_1, \Theta_2, \Theta_3\}$ as follows.

Merging. The merging operator Θ_1 groups a view-centric parse graph with an other view-centric parse graph by creating a scene-centric parse graph that connects the two. The operator requires the two operands to describe two objects of the same type either from different views or in the same view but with non-overlapping time intervals.

Splitting. The splitting operator Θ_2 splits a scene-centric parse graph into two parse graphs such that each resulting parse graph only connects to a subset of view-centric parse graphs.

Swapping. The swapping operator Θ_3 swaps two view-centric parse graphs. One can view the swapping operator as a shortcut of merging and splitting combined.

We define the proposal distribution $q(G \to G')$ as an uniform distribution. At each iteration, we generate a new structure proposal Φ' by applying one of the three operators Θ_i with respect to probability 0.4, 0.4, and 0.2, respectively. The generated proposal is then accepted with respect to an acceptance rate $\alpha(\cdot)$ as in the Metropolis-Hastings algorithm [MRR53]:

$$\alpha(G \to G') = \min\left(1, \frac{q(G' \to G) \cdot p(G'|x)}{q(G \to G') \cdot p(G|x)}\right), \tag{4.10}$$

where $p(G|x)$ the posterior is defined in Equation (4.2).

4.5.2 Inferring Parse Graph Variables

Given a fixed parse graph hierarchy, we need to estimate the optimal value for each node within each parse graph. As illustrated in Fig. 4.3, for each frame, the scene-centric node g_t and the corresponding view-centric nodes $\tilde{g}_t^{(i)}$ form a star model, and the whole scene-centric nodes are regarded as a Markov chain in the temporal order. Therefore the proposed model is essentially a Directed Acyclic Graph (DAG). To infer the optimal node values, we can simply apply the standard factor graph belief propagation (sum-product) algorithm.

4.6 Experiments

4.6.1 Setup and Datasets

We evaluate our scene-centric joint-parsing framework in tasks including object detection, multi-object tracking, action recognition, and human attributes recognition. In object detection and multi-object tracking tasks, we compare with published results. In action recognition and human attributes tasks, we compare the performance of view-centric proposals without joint parsing and scene-centric predictions after joint parsing as well as additional baselines. The following datasets are used to cover a variety of tasks.

The **CAMPUS** dataset [XLL16][1] contains video sequences from four scenes each captured by four cameras. Different from other multi-view video datasets focusing solely on multi-object tracking task, videos in the CAMPUS dataset contains richer human poses and activities with moderate overlap in the fields of views between cameras. In addition to the tracking annotation in the CAMPUS dataset, we collect new annotation that includes 5 action categories and 9 attribute categories for evaluating action and attribute recognition.

[1]Available at https://bitbucket.org/merayxu/multiview-object-tracking-dataset

The **TUM Kitchen** dataset [TBB09][2] is an action recognition dataset that contains 20 video sequences captured by 4 cameras with overlapping views. As we only focusing on the RGB imagery inputs in our framework, other modalities such as motion capturing, RFID tag reader signals, magnetic sensor signals are not used as inputs in our experiments. To evaluate detection and tracking task, we compute human bounding boxes from motion capturing data by projecting 3D human poses to the image planes of all cameras using the intrinsic and extrinsic parameters provided in the dataset. To evaluate human attribute tasks, we annotate 9 human attribute categories for every subject.

In our experiments, both the CAMPUS and the TUM Kitchen datasets are used in all tasks. In the following subsection, we present isolated evaluations.

4.6.2 Evaluation

Object detection & tracking. We use FasterRCNN [RHG15] to create initial object proposals on all video frames. The detection scores are used in the likelihood term in Equation (4.3). During joint parsing, objects which are not initially detected on certain views are projected from object's scene-centric positions with the camera matrices. After joint parsing, we extract all bounding boxes that are grounded by object nodes from each view-centric parse graph to compute multi-object detection accuracy (DA) and precision (DP). Concretely, the accuracy measures the faction of correctly detected objects among all ground-truth objects and the precision is computed as fraction of true-positive predictions among all output predictions. A predicted bounding box is considered a match with a ground-truth box only if the intersection over union (IoU) score is greater than 0.5. When more than one prediction overlaps with a ground-truth box, only the one with the maximum overlap is counted as true positive.

When extracting all bounding boxes on which the view-centric parse graphs are grounded and grouping them according to the identity correspondence between different views, we obtain object trajectories with identity matches across multiple videos. In the evaluation,

[2]Available at https://ias.in.tum.de/software/kitchen-activity-data

CAMPUS-S1	DA (%)	DP (%)	TA (%)	TP (%)	IDSW	FRAG
Fleuret *et al.*	24.52	64.28	22.43	64.17	2269	2233
Berclaz *et al.*	30.47	62.13	28.10	62.01	2577	2553
Xu *et al.*	49.30	72.02	56.15	72.97	320	141
Ours	**56.00**	72.98	**55.95**	72.77	310	138
CAMPUS-S2	DA (%)	DP (%)	TA (%)	TP (%)	IDSW	FRAG
Fleuret *et al.*	16.51	63.92	13.95	63.81	241	214
Berclaz *et al.*	24.35	61.79	21.87	61.64	268	249
Xu *et al.*	27.81	71.74	28.74	71.59	1563	443
Ours	**28.24**	71.49	**27.91**	71.16	1615	418
CAMPUS-S3	DA (%)	DP (%)	TA (%)	TP (%)	IDSW	FRAG
Fleuret *et al.*	17.90	61.19	16.15	61.02	249	235
Berclaz *et al.*	19.46	59.45	17.63	59.29	264	257
Xu *et al.*	49.71	67.02	49.68	66.98	219	117
Ours	**50.60**	67.00	**50.55**	66.96	212	113
CAMPUS-S4	DA (%)	DP (%)	TA (%)	TP (%)	IDSW	FRAG
Fleuret *et al.*	11.68	60.10	11.00	59.98	828	812
Berclaz *et al.*	14.73	58.51	13.99	58.36	893	880
Xu *et al.*	24.46	66.41	24.08	68.44	962	200
Ours	**24.81**	66.59	**24.63**	68.28	938	194
TUM Kitchen	DA (%)	DP (%)	TA (%)	TP (%)	IDSW	FRAG
Fleuret *et al.*	69.88	64.54	69.67	64.76	61	57
Berclaz *et al.*	72.39	63.27	72.20	63.51	48	44
Xu *et al.*	86.53	72.12	86.18	72.37	9	5
Ours	**89.13**	72.21	**88.77**	72.42	12	8

Table 4.1: Quantitative comparisons of multi-object tracking on CAMPUS and TUM Kitchen datasets.

Methods	CAMPUS					
	Run	PickUp	PutDown	Throw	Catch	Overall
view-centric	0.83	0.76	0.91	0.86	0.80	0.82
baseline-vote	0.85	0.80	0.71	0.88	0.82	0.73
baseline-mean	0.86	0.82	1.00	0.90	0.87	0.88
scene-centric	0.87	0.83	1.00	0.91	0.88	**0.90**

Methods	TUM Kitchen								
	Reach	Taking	Lower	Release	OpenDoor	CloseDoor	OpenDrawer	CloseDrawer	Overall
view-centric	0.78	0.66	0.75	0.67	0.48	0.50	0.50	0.42	0.59
baseline-vote	0.80	0.63	0.77	0.71	0.72	0.73	0.70	0.47	0.69
baseline-mean	0.79	0.61	0.75	0.69	0.67	0.67	0.66	0.45	0.66
scene-centric	0.81	0.67	0.79	0.71	0.71	0.73	0.70	0.50	**0.70**

Table 4.2: Quantitative comparisons of human action recognition on CAMPUS and TUM Kitchen datasets.

we compute four major tracking metrics: multi-object tracking accuracy (TA), multi-object track precision (TP), the number of identity switches (IDSW), and the number of fragments (FRAG). A higher value of TA and TP and a lower value of IDSW and FRAG indicate the tracking method works better. We report quantitative comparisons with several published methods [XLL16, BFT11, FBL08] in Table 4.1. From the results, the performance measured by tracking metrics are comparable to published results. We conjecture that the appearance similarity is the main drive for establish cross-view correspondence while additional semantic attributes proved limited gain to the tracking task.

Action recognition. View-centric action proposals are obtained from a fully-connected neural network with 5 hidden layers and 576 neurons which predicts action labels using human pose. For the CAMPUS dataset, we collect additional annotations for 5 human action classes: Run, PickUp, PutDown, Throw, and Catch in total of 8,801 examples. For the TUM Kitchen dataset, we evaluate on the 8 action categories: Reaching, TakingSomething, Lowering, Releasing, OpenDoor, CloseDoor, OpenDrawer, and CloseDrawer. We measure both individual accuracies for each category as well as the overall accuracies across all categories.

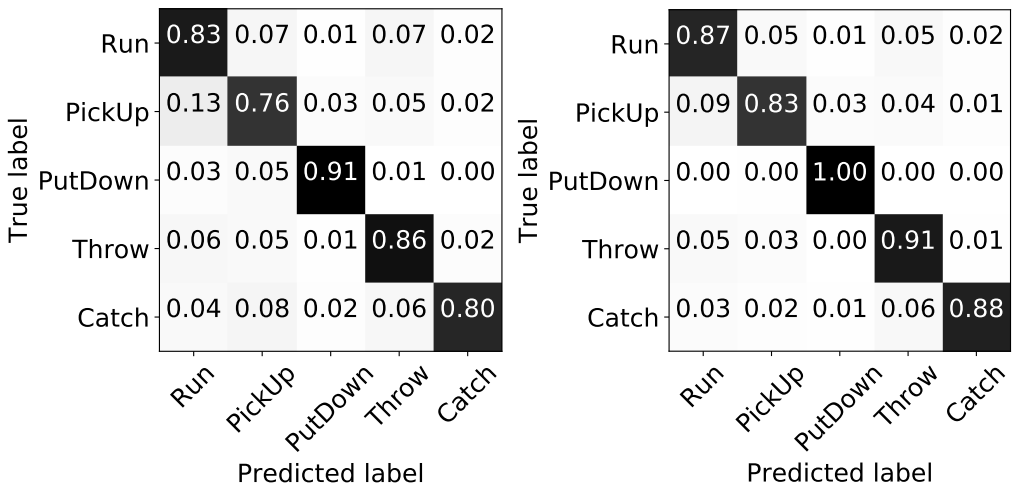


Figure 4.4: Confusion matrices of action recognition on view-centric proposals (left) and scene-centric predictions (right).

	Methods	Gender	Long hair	Glasses	Hat	T-shirt	Long sleeve	Shorts	Jeans	Long pants	mAP
CAMPUS	view-centric	0.59	0.77	0.56	0.76	0.36	0.59	0.70	0.63	0.35	0.59
	baseline-mean	0.63	0.82	0.55	0.75	0.34	0.64	0.69	0.63	0.34	0.60
	baseline-vote	0.61	0.82	0.55	0.75	0.34	0.65	0.69	0.63	0.35	0.60
	scene-centric	0.76	0.82	0.62	0.80	0.40	0.62	0.76	0.62	0.24	**0.63**
	Methods	Gender	Long hair	Glasses	Hat	T-shirt	Long sleeve	Shorts	Jeans	Long pants	mAP
TUM Kitchen	view-centric	0.69	0.93	0.32	1.00	0.50	0.89	0.91	0.83	0.73	0.76
	baseline-mean	0.86	1.00	0.32	1.00	0.54	0.96	1.00	0.83	0.81	0.81
	baseline-vote	0.64	1.00	0.32	1.00	0.32	0.93	1.00	0.83	0.76	0.76
	scene-centric	0.96	0.98	0.32	1.00	0.77	0.96	0.94	0.83	0.83	**0.84**

Table 4.3: Quantitative comparisons of human attribute recognition on CAMPUS and TUM Kitchen datasets.

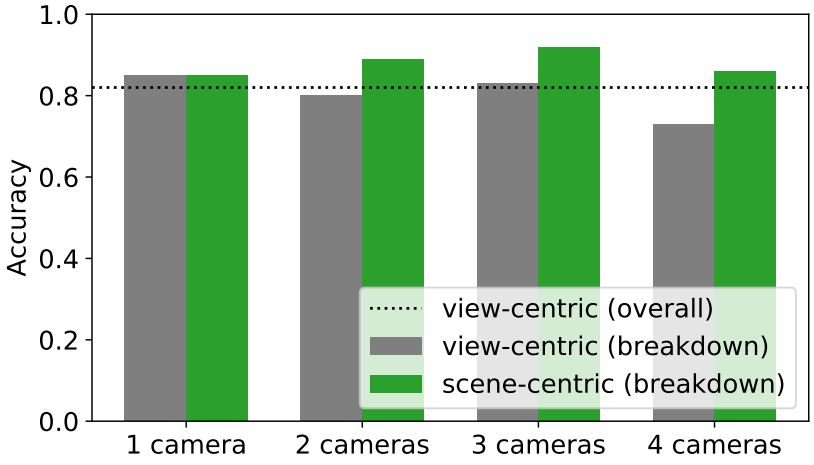


Figure 4.5: The breakdown of action recognition accuracy according to the number of camera views in which each entity is observed.

Table 4.2 shows the performance of scene-centric predictions with view-centric proposals, and two additional fusing strategies as baselines. Concretely, the *baseline-vote* strategy takes action predictions from multiple views and outputs the label with majority voting, while the *baseline-mean* strategy assumes equal priors on all cameras and outputs the label with the highest averaged probability. When evaluating scene-centric predictions, we project scene-centric labels back to individual bounding boxes and calculate accuracies following the same procedure as evaluating view-centric proposals. Our joint parsing framework demonstrates improved results as it aggregates marginalized decisions made on individual views while also encourages solutions that comply with other tasks. Fig. 4.4 compares the confusion matrix of view-centric proposals and scene-centric predictions after joint parsing for CAMPUS dataset. To further understand the effect of multiple views, we break down classification accuracies by the number of cameras where persons are observed (Fig. 4.5). Observing an entity from more cameras generally leads to better performance, while too many conflicting observations may also cause degraded performance. Fig. 4.6 shows some success and failure examples.

Human attribute recognition. We follow the similar procedure as in the action recognition case above. Additional annotations for 9 different types of human attributes are collected for both CAMPUS and TUM Kitchen dataset. View-centric proposals and score

Figure 4.6: Success (1st row) and failure examples (2nd row) of view-centric (labels overlaid on the images) and scene-centric predictions (labels beneath the images) of action and attribute recognition tasks. For failure examples, true labels are in the bracket. "Occluded" means that the locations of objects or parts are projected from scene locations and therefore no view-centric proposals are generated. Better viewed in color.

are obtained from an attribute grammar model as in [PNZ15]. We measure performance with average precisions for each attribute categories as well as mean average precision (mAP) as in human attribute literatures. Scene-centric predictions are projected to bounding boxes in each views when calculating precisions. Table 4.3 shows quantitative comparisons between view-centric and scene-centric predictions. The same baseline fusing strategies as in the action recognition task are used. The scene-centric prediction outperforms the original proposals in 7 out of 9 categories while remains comparable in others. Notably, the CAMPUS dataset is harder than standard human attribute datasets because of occlusions, limited scales of humans, and irregular illumination conditions.

4.6.3 Runtime

With initial view-centric proposals precomputed, for a 3-minute scene shot by 4 cameras containing round 15 entities, our algorithm performs at 5 frames per second on average. With further optimization, our proposed method can run in real-time. Note that although the

proposed framework uses a sampling-based method, using view-based proposals as initialization warm-starts the sampling procedure. Therefore, the overall runtime is significantly less than searching the entire solution space from scratch. For problems of a larger size, more efficient MCMC algorithms may be adopted. For example, the mini-batch acceptance testing technique [CSP16] has demonstrated several order-of-magnitude speedups.

4.7 Summary

We represent a joint parsing framework that computes a hierarchy of parse graphs which represents a comprehensive understanding of cross-view videos. We explicitly specify various constraints that reflect the appearance and geometry correlations among objects across multiple views and the correlations among different semantic properties of objects. Experiments show that the joint parsing framework improves view-centric proposals and produces more accurate scene-centric predictions in various computer vision tasks.

We briefly discuss advantages of our joint parsing framework and potential future directions from two perspectives.

4.7.0.1 Explicit Parsing

While the end-to-end training paradigm is appealing in many *data-rich* supervised learning scenarios, as an extension, leveraging loosely-coupled pre-trained modules and exploring commonsense constraints can be helpful when large-scale training data is not available or too expensive to collect in practice. For example, many applications in robotics and human-robot interaction domains share the same set of underlying perception units such as scene understanding, object recognition, etc. Training for every new scenarios entirely could end up with exponential number of possibilities. Leveraging pre-trained modules and explore correlation and constraints among them can be treated as a factorization of the problem space. Therefore, the explicit joint parsing scheme allows practitioners to leverage pre-trained modules and to build systems with an expanded skill set in a scalable manner.

4.7.0.2 Interpretable Interface

Our joint parsing framework not only provides a comprehensive scene-centric understanding of the scene, moreover, the scene-centric spatio-temporal parse graph representation is an interpretable interface of computer vision models to users. In particular, we consider the following properties an explainable interface shall have apart from the correctness of answers:

- *Relevance*: an agent shall recognize the intent of humans and provide information relevant to humans' questions and intents.

- *Self-explainability*: an agent shall provide information that can be interpreted by humans as how answers are derived. This criterion promotes humans' trust on an intelligent agent and enables sanity check on the answers.

- *Consistency*: answers provided by an agents shall be consistent throughout an interaction with humans and across multiple interaction sessions. Random or non-consistent behaviors cast doubts and confusions regarding the agent's functionality.

- *Capability*: an explainable interface shall help humans understand the boundary of capabilities of an agent and avoid blinded trusts.

Potential future directions include quantifying and evaluating the interpretability and user satisfaction by conducting user studies.

CHAPTER 5

Human Parsing by Joint Bottom-up and Top-down Inference

5.1 Introduction

Neural networks are currently revolutionizing computer vision. Their wide-ranging success has proven their strong representation power and end-to-end learning ability. However, they may not directly encode interpretable structures and top-down information. For example, it's difficult to incorporate the knowledge of human body decomposability into networks since the intrinsic mechanism of a network is often hard to explain. Alternatively, graphical models are powerful to build structured representations, which is the incentive for their prevalence in computer vision. Such structured representations could reflect task-specific relations and constraints. For example, in cloth landmark localization (see Fig. 5.1), nodes represent atomic components (*e.g.*, collars, hems, *etc.*), and edges describe node inter-relations (*e.g.*, kinematic dependencies among cloth landmarks). Graphical models allow domain experts to inject their high-level knowledge, but often require significant feature engineering.

We propose a deep structured network, named α-β-γ network, which augments the hierarchical graphical representation with the learning capability of neural network, and pursue to connote three information flows, straight pass (*i.e.*, α process), bottom-up process (*i.e.*, β process) and top-down process (*i.e.*, γ process), in hierarchical models. As illustrated in Fig. 5.1, when predicting the location of *upper-body cloth* of a person, we consider three kinds of information: image regions directly revealing itself, the decompositional relation from parent *full-body cloth*, and the compositional relations from children l.&r. collar and l.&r. hem. Different information flows confer different portions of contributions to the final

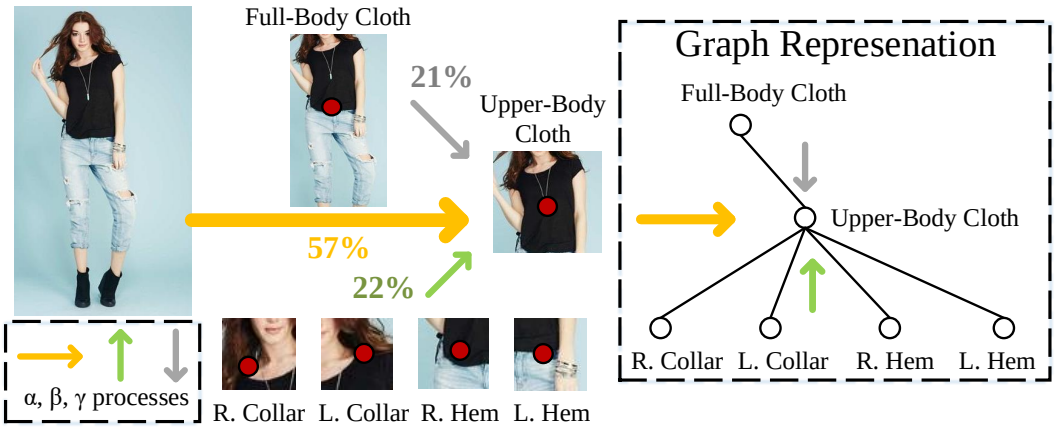


Figure 5.1: **Illustration of joint bottom-up and top-down inference**. Given a hierarchy of human cloth, three types of information (*i.e.*, α, β, and γ processes) contribute to the final prediction of *upper body cloth* node. With α-β-γ network, these three processes can be explicitly learned in an end-to-end manner with post-hoc interpretability.

prediction. There are some interesting properties about α-β-γ network:

• **Encodings of joint bottom-up and top-down inference**. Our structured network models graph nodes as CNNs and takes into account the dependencies (*e.g.*, composition, decomposition, and contextual relation) within the hierarchical graph. It provides a principled algorithm for learning hierarchical graphical models jointly with neural networks. The proposed model approaches three basic inference processes [WZ11b] with end-to-end learning: α process directly generates predictions based on image features, β process makes predictions by binding child node(s) in bottom-up fashion and γ process utilizes contextual information from parent node(s) in top-down style. We show that, with the suggested model, the hierarchical graph with above three processes can be efficiently learned in a stochastic way.

• **Post-hoc interpretability**. A major benefit of our network lies in the post-hoc interpretability. Taking human pose detection as an example, it is intuitive that people usually directly observe a certain node (*e.g.*, human arm), without occlusion. When the arm node is partially occluded, people rely more on bottom-up process that considers the information from those non-occluded child nodes (*e.g.*, hand). When the arm node is heavily occluded or becomes indistinguishable, people would still recognize this node with the high-level prior

knowledge of human body articulation. As seen, α, β and γ processes are straightforward and interpretable, which leads to more interpretability compared with previous structured deep learning methods. Thus our model is able to provide information that can be interpreted by humans as how results are inferred and combined from top-down/bottom processes. Additionally, such self-interpretability is also measurable, which can be evaluated as the agreement between inference processes and respective human performance.

We conduct experiments on two tasks, *i.e.*, cloth landmark localization and human pose detection, to verify the effectiveness and generalization of our model. The selected two tasks by nature implies complex hierarchical structures. Results show that our method outperforms competing methods *with the similar model complexity*. From the experimental results, we further observe that: (i) α process is generally stronger than β and γ processes; (ii) α process is favored for low-level nodes (*e.g.*, cloth landmarks, human joints), while β and γ processes are preferred for high-level nodes (*e.g.*, full-body cloth, or upper-body pose); (iii) combining three inference processes is beneficial to final predictions.

Contributions. The contributions are three-field: i) a deep network representing hierarchical graphical structures; ii) explicit encodings of three inference processes with end-to-end learning; iii) post-hoc interpretability.

The rest of this chapter is organized as follows. We first review the related work in § 5.2, then discuss the representation and formulation of our model in § 5.3 and § 5.4, respectively. We further elaborate the learning process in § 5.5. We report experiments and comparisons in § 5.6, and finally summarize this chapter in § 5.7.

5.2 Related Work

We give a categorized overview of the related literature, yet not limited to specific tasks. In general, there are three main characteristics differentiating our work from existing techniques: being flexible to any deep networks, outlining a unified framework for modeling the bottom-up/top-down processes with end-to-end training, and being fully trainable and better interpretable with explicit inference processes. In general, our work is closely related to

two streams of research in the literature:

Hierarchical graphical models have an enormous impact in computer vision, as they are powerful for expressing and capturing inherent structures, contextual information and high-level human knowledge. Their applications span from low-level problems, *e.g.*, hierarchical clustering, image restoration, to high-level tasks, *e.g.*, object parsing, human-object interaction. Commonly used models include MRF/CRF [JFY09], part-based models [FGM10, LLA16], and And-Or Graph [KMY06, WZ11b, SWJ13]. For inference, bottom-up process passes information in a feed-forward manner while top-down process in a feed-back fashion over the hierarchy. In this work, we extend deep learning algorithm to hierarchical graphical model for end-to-end learning bottom-up and top-down processes jointly and automatically.

Deep learning with graphical models has recently received growing interests. Many recent works [CSY15, ZJR15, ZSG15, CPK16] focus on incorporating CRF into networks with end-to-end training. Others extend RNN [MJ99], or LSTM [SLM11] from chain structures to tree or graph structures [TSM15, JZS16, LLS17, CLX16]. However, they largely address the bottom-up process over structured architectures or work in a mixed fashion of bottom-up and top-down manners. In comparison, our model formulates the bottom-up and top-down inference processes in an explicit way. It is also a more principled and interpretable framework for modeling complex graph structures, rather than previous models limited to MRF assumptions or implicit mechanics.

Some works explore the **top-down mechanism in neural networks** and demonstrate success in their specific tasks. More specifically, bottom-up/top-down network architectures are proposed for leveraging both low-level and high-level features from different layers in semantic segmentation [LSD15, NHH15]. Some investigations [CLY15, HR16] focus on inspiring information flow between feed-forward and feedback loops in networks. However, these works (i) often perform inference over DNN hierarchy, without considering semantic hierarchical structures and relations in graph models; (ii) very few touch how to *explicitly* learn the bottom-up and top-down processes over a hierarchical graph. Additionally, the proposed model is more favored due to its post-hoc interpretability that specifies how its

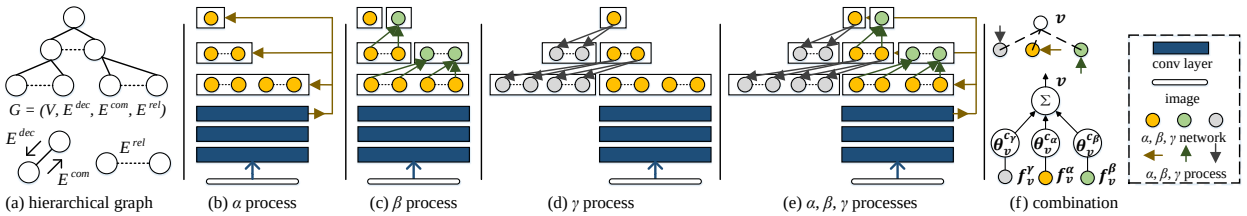


Figure 5.2: **The proposed α-β-γ network**. (a) The encoded hierarchical graph $\mathcal{G}$. (b)-(d) α-, β-, and γ- networks encoding α, β, γ process. (e) Joint inference based on neural network. (f) Fusion of three information flows. See text for detailed explanations.

outputs are inferred from different information flows.

5.3 Representation

Given a graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$, where nodes $\mathcal{V}$ represent problem components, and edges $\mathcal{E}$ capture the relationships between nodes (see Fig. 5.2(a)). Since we concentrate on hierarchical graphical model, we decompose nodes $\mathcal{V}$ into L layers: $\mathcal{V}=\mathcal{V}^1\cup\ldots\cup\mathcal{V}^L$, where $\mathcal{V}^l$ indicates the set of the nodes in l-th layer and the root node locates in the first layer (l=1). Edges can be further decomposed into three categories: $\mathcal{E}=\mathcal{E}^{com}\cup\mathcal{E}^{dec}\cup\mathcal{E}^{rel}$. $\mathcal{E}^{com}$ and $\mathcal{E}^{dec}$ are sets of vertical edges connecting parent nodes with their child nodes, which represent hierarchical constraints of composition and decomposition. Note vertical edges work in both bottom-up and top-down directions (*i.e.*, undirected edges), we use $\mathcal{E}^{com}$ and $\mathcal{E}^{dec}$ denote edging directing upwards and downwards, respectively. $\mathcal{E}^{rel}$ refers to the set of horizontal edges connecting among siblings with the same parent, which describes contextual relations in hierarchy. As suggested in [WZ11b], three inference processes, termed α, β and γ processes, can be derived for each node $v\in\mathcal{V}$.

α **process** detects node v directly based on image features. The α process is the basic inference, which can work alone (without taking advantage of surrounding context). Most structured networks [TJL14, CSY15, LSH16, CPK16] in literature are proposed in this line. It can be viewed as either bottom-up or top-down. By bottom-up, it means that discriminative models. By top-down, it means that generative models are used.

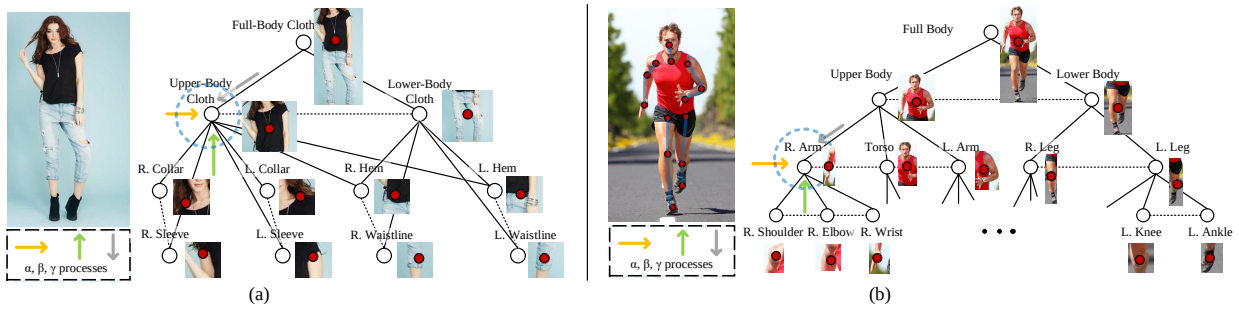


Figure 5.3: **Graphical representations for cloth landmark localization (a) and human pose estimation (b)**, where blue circles illustrate the α, β, γ processes of *upper-body cloth* node and *right arm* node, respectively.

β **process** computes node v by binding the detected child nodes in bottom-up fashion, where the child nodes' α processes are activated. An intuitive interpretation of this inference is to infer an occluded node, like human head, from its detected sub-nodes, say eye node or mouth node. Most component or part based models [LLS17] belong to this process.

γ **process** predicts node v top-down from its parent nodes whose α processes are activated. The parent node passes contextual information, such as we can detect human head node even we only see the outline of the person. Most of the context-based methods belong to this process.

We propose a deep learning algorithm to learn above processes over the graph $\mathcal{G}$. In the high level, each node v is parameterized as a stack of CNNs by means of learning capacity and differentiable property. Each node accepts the information from other nodes, in bottom-up (β process) or top-down (γ process) manner as input, or directly uses the deep learning features from an underlying network (α process) for inference. In this way, we build a structured and fully differentiable network, which efficiently models $\mathcal{G}$ and explicitly learns inferences with powerful back-propagation.

5.4 Problem Formulation

According to Bayes rule, $\mathcal{G}$ can be solved by maximizing a posterior (MAP), that is,

$$\mathcal{G}^* = \arg\max_{\mathcal{G}} p(\mathcal{G}|I;\theta) \propto \arg\max_{\mathcal{G}} p(I|\mathcal{G};\theta)\cdot p(\mathcal{G};\theta), \tag{5.1}$$

where θ indicates the model parameters.

Likelihood $p(I|\mathcal{G};\theta)$ measures how well the observed image data I satisfies the hierarchical model $\mathcal{G}$. We assume each node in $\mathcal{G}$ only corresponds to a certain region of image, thus the likelihood $p(I|\mathcal{G};\theta)$ can be decomposed as:

$$p(I|\mathcal{G};\theta)=\prod_{v\in\mathcal{V}} p(I_{\Lambda_v}|v;\theta) = p^{bg}(I_\Lambda)\prod_{v\in\mathcal{V}}\frac{p^{fg}(I_{\Lambda_v}|v;\theta_v^\alpha)}{p^{bg}(I_{\Lambda_v})}, \tag{5.2}$$

where Λ denotes image lattice and Λ_v denotes the image region occupied by node v, $p^{bg}(\cdot)$ and $p^{fg}(\cdot)$ denote background and foreground probability, respectively. Similar to [SK04, CZY09], $p^{bg}(I_\Lambda)$ can be assumed as a constant and the likelihood ratio $g(\cdot) = p^{fg}(\cdot)/p^{bg}(\cdot)$ can be regarded as a logistic regression. This ratio represents the straight pass inference (*i.e.*, α **process**). For each node, the α process consists of two sub-steps: (i) extracting features ϕ_I from raw images and (ii) making direct predictions based on the extracted features.

Prior $p(\mathcal{G};\theta)$ imposes constraints on the hierarchy, measuring the compatibilities among composition edges $\mathcal{E}^{com}$, decomposition edges $\mathcal{E}^{dec}$ and contextual edges $\mathcal{E}^{rel}$:

$$\begin{aligned} p(\mathcal{G};\theta)&=\prod_{v\in\mathcal{V}} p(v;\theta_v^c)\cdot p(nb(v)|v),\\ &=\prod_{v\in\mathcal{V}} p(v;\theta_v^c)\cdot p(ch(v)|v;\theta_v^\beta)\cdot p(pr(v)|v;\theta_v^\gamma)\cdot p(sb(v)|v), \end{aligned} \tag{5.3}$$

where $nb(v)$, $ch(v)$, $pr(v)$ and $sb(v)$ denote neighbors, children, parents, siblings of node v, respectively. Note that, for some nodes, the composition or decomposition edges (*i.e.*, β or γ processes) might not exist. Terminal leaf nodes only have α and γ processes, while the root node only has α and β processes. For clarity, we omit such cases, as they do not affect the method description.

Prior term $p(v;\theta_v^c)$ measures to what extent we should trust different information sources (*i.e.*, information flows from α,β,γ processes). For each v, the fusion term $p(v;\theta_v^c)$ is defined

as a weighted combination of α, β, γ processes:

$$\begin{aligned} p(v;\theta_v^c) &= [\theta_v^{c_\alpha}, \theta_v^{c_\beta}, \theta_v^{c_\gamma}], \\ s.t.\ \theta_v^{c_\alpha} \geq 0,\ \theta_v^{c_\beta} \geq 0,\ \theta_v^{c_\gamma} \geq 0,\ \theta_v^{c_\alpha} + \theta_v^{c_\beta} + \theta_v^{c_\gamma} &= 1. \end{aligned} \tag{5.4}$$

Prior term $p(ch(v)|v;\theta_v^\beta)$ represents the bottom-up inference (*i.e.*, β **process**), which considers information flow upward from descendants. Each node v is fed with the information flow from its child nodes $ch(v)$, which composes composition edges $\mathcal{E}_v^{com}$.

Prior term $p(pr(v)|v;\theta_v^\gamma)$ represents the top-down inference (*i.e.*, γ **process**). Each node v is fed with the information flow from its parent nodes $pr(v)$ in γ process, which conveys the high-level information in a top-down manner. This describes decomposition edges $\mathcal{E}_v^{dec}$.

So far, we have discussed the formulation of nodes with vertical connections in the graph $\mathcal{G}$, which allows us to utilize information from straight pass (*i.e.*, α process), bottom-up process (*i.e.*, β process) and top-down process (*i.e.*, γ process). This generally covers composition relations $\mathcal{E}^{com}$ and decomposition relations $\mathcal{E}^{dec}$. Last but not least, our model should be able to capture contextual relations $\mathcal{E}^{rel}$.

Prior term $p(sb(v)|v)$ describes horizontal edges $\mathcal{E}^{rel}$ among siblings, which could represent many possible contextual relations, such as object-object interactions, dependency grammars and kinematic relations. In this , we consider contextual relations are encoded in prior terms $p(ch(v)|v;\theta_v^\beta)$ and $p(pr(v)|v;\theta_v^\gamma)$, which are joint distributions for child nodes and parent nodes given node v, respectively, while prior work usually assumes conditional independence among siblings. Thus we choose to implicitly model contextual relations in our α-β-γ network, which will be elaborated in next section.

In summary, our model encodes four probability distributions for each node v, parameterized by

$$\theta = \{(\theta_v^\alpha, \theta_v^\beta, \theta_v^\gamma, \theta_v^c) :\ v \in \mathcal{V}\}. \tag{5.5}$$

5.5 Learning

For each $v \in \mathcal{V}$, we further derive three sub-networks, namely α-, β-, and γ- network, for learning α, β, and γ processes.

α-process. The α-network f_v^α, parameterized by θ_v^α, is learned for node v. It takes cropped images under corresponding lattice Λ_v as inputs and prediction score maps as outputs:

$$g(I_{\Lambda_v}|v;\theta_v^\alpha) = f_v^\alpha(\phi_{I_{\Lambda_v}};\theta_v^\alpha), \tag{5.6}$$

where the image features $\phi_{I_{\Lambda_v}}$ are extracted from a underlying network. As shown in Fig. 5.2(b), the final score map can be obtained by applying logistic *sigmoid* activation function.

β-process. As shown in Fig. 5.2(c), the β-network f_v^β for node v utilizes the information of its child nodes $ch(v)$ in α process, and outputs prediction score as the result of β process:

$$\begin{aligned} &p(ch(v)|v;\theta_v^\beta) \propto f_v^\beta(\phi_{ch(v)};\theta_v^\beta), \\ &\phi_{ch(v)} = \mathbf{P}_{avg}(\,\{f_{v'}^\alpha(v') : v' \in ch(v)\}\,), \end{aligned} \tag{5.7}$$

where we use channel-wise average-pooling operation $\mathbf{P}_{avg}$ for combining the output scores from child nodes. Such operation is important for transforming features from a variable number of predictions from child nodes to a fixed-size feature representation. Note that any commutative operations can be used as alternatives (*e.g.*, sum-pooling, max-pooling).

γ-process. γ process works on the knowledge transferred from the parent node which is activated in α process. For node v, γ-network f_v^γ takes the information $\phi_{pr(v)}$ from parent nodes $pr(v)$ as input, and generates prediction as the output of γ process (see Fig. 5.2(d)):

$$\begin{aligned} &p(pr(v)|v;\theta_v^\gamma) \propto f_v^\gamma(\phi_{pr(v)};\theta_v^\gamma), \\ &\phi_{pr(v)} = \mathbf{P}_{avg}(\,\{f_{v'}^\alpha(v') : v' \in pr(v)\}\,). \end{aligned} \tag{5.8}$$

Fusion of α, β, γ processes. As illustrated in Fig. 5.2(f), the final prediction is made by a weighted combination of outputs generated from α-, β-, and γ- networks, parameterized by $\theta_v^c = [\theta_v^{c_\alpha}, \theta_v^{c_\beta}, \theta_v^{c_\gamma}]$. We represent the combination weights as 1×1 convolution layer con-

Nodes	Configuration
α-network	conv(256,(3,3)) → conv(128,(5,5)) → conv(64,(3,3)) → conv(32,(3,3))
β-network	conv(32, (7,7)) → conv(32, (7,7)) → conv(32,(7,7)) → conv(32,(7,7)) → conv(32,(5,5))
γ-network	conv(32, (7,7)) → conv(32, (7,7)) → conv(32,(7,7)) → conv(32,(7,7)) → conv(32,(5,5))

Table 5.1: **Configurations of α-, β-, and γ- networks**. Keras notations (channels, kernel) are used to define the conv layers.

Methods	3rd Layer								2nd Layer		1st Layer
	L.Collar	R.Collar	L.Sleeve	R.Sleeve	L.Waistline	R.Waistline	L.Hem	R.Hem	U.Body	L.Body	F.Body
FashionNet CVPR'16 [LLQ16]	.0784	.0803	.0975	.0923	.0874	.0821	.0802	.0893	-	-	-
DFA ECCV'16 [LYL16]	.048	.048	.091	.089	-	-	.071	.072	-	-	-
DLAN AAAI'17 [YLL17]	.0531	.0547	.0705	.0735	.0752	.0748	.0693	.0675	-	-	-
FashionGrammar CVPR'18 [WXS18]	.0463	.0471	.0627	.0614	.0635	.0692	.0635	.0527	-	-	-
α-network	.0457	.0450	.0619	.0628	.0623	.0705	.0643	.0530	.1220	.1186	.0935
β-network	-	-	-	-	-	-	-	-	.1100	.1105	.0820
γ-network	.0503	.0512	.0721	.0713	.0643	.0821	.0703	.0627	.1002	.1013	-
α-β-γ network w/o share	.0441	**.0415**	.0606	.0615	.0620	.0702	**.0624**	.0515	.0994	.0986	.0790
α-β-γ network	**.0435**	.0426	**.0597**	**.0612**	**.0614**	**.0690**	.0631	**.0511**	**.0989**	**.0977**	**.0778**

Table 5.2: **Comparison of normalized error (NE) on FLD dataset**. Lower values are better. The best score is marked in **bold**.

necting three channels (without bias term) and enforce the non-negativity and normalization constraints in Equation (5.4) to preserve model interpretability.

α-β-γ **network**. As shown in Fig. 5.2(e), the joint framework composes all the above components into a unified network, which can be learned in an end-to-end manner. Given ground-truth $\hat{v}$ for each node in the hierarchy $\hat{\mathcal{V}}$ with total K training samples, the α-β-γ network can be learned as:

$$\theta^* = \arg\max_{\theta} \prod_{k=1}^{K} p(\hat{\mathcal{V}}_k | I_k, \theta) = \arg\min_{\theta} \sum_{k=1}^{K} \sum_{\hat{v} \in \hat{\mathcal{V}}_k} L(\hat{v}|\theta_v), \tag{5.9}$$

where $L(\hat{v}|\theta_v)$ is the prediction loss for node v. The losses are defined as per the respective tasks, which are elaborated in next section. Our whole model is differentiable, and thus all the parameters θ of the graph model (in Equation (5.5)) can be trained in a stochastic way.

Parameter sharing. Noticing that CNNs are naturally inherited to describe relations among all nodes in higher layer $\mathcal{V}^l$ and all nodes in lower layer $\mathcal{V}^{l+1}$. We thus employ

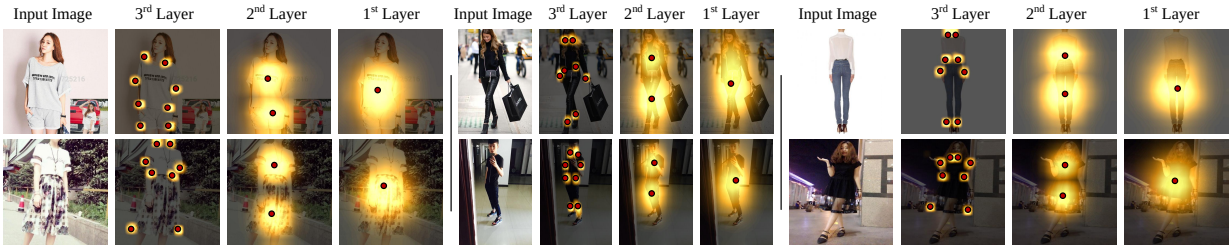


Figure 5.4: **Results of cloth landmark localization**. We show the prediction scores of each layer in our hierarchical graph, where the brighter pixel indicates higher prediction values, and the red circle indicates the location of highest score of each node.

parameter sharing among siblings to encourage information exchange, instead of modeling $\mathcal{E}^{rel}$ explicitly. We partition nodes $\mathcal{V}^l$ in l-th layer into N unconnected groups: $\mathcal{V}^l = \mathcal{V}^l_1 \cup \ldots \cup \mathcal{V}^l_N$, according to their sibling relations. Then we enforce parameter sharing among nodes from same groups, instead of learning distinct parameters for each node. Taking Fig. 5.3(b) as an example, there exist kinematic relations (represented as dotted lines) among human body parts: *r. shoulder* $\leftrightarrow$ *r. elbow* and *r. elbow* $\leftrightarrow$ *r. wrist*, where three nodes are siblings with the same parent node *r. arm*. We model these three nodes with the same parameterization, which represents the knowledge sharing among them. Parameter sharing not only enables our network to capture complex inter-sibling relations and allows siblings to bootstrap each other capabilities, but also brings higher flexibility and better training efficiency on a large hierarchical structure. Overall, our entire model (including the underlying network) is fully differentiable, thus can be trained in end-to-end manner.

5.6 Experiments

We validate our α-β-γ network on two vision tasks: cloth landmark localization and human pose estimation. Then, we study post-hoc interpretability.

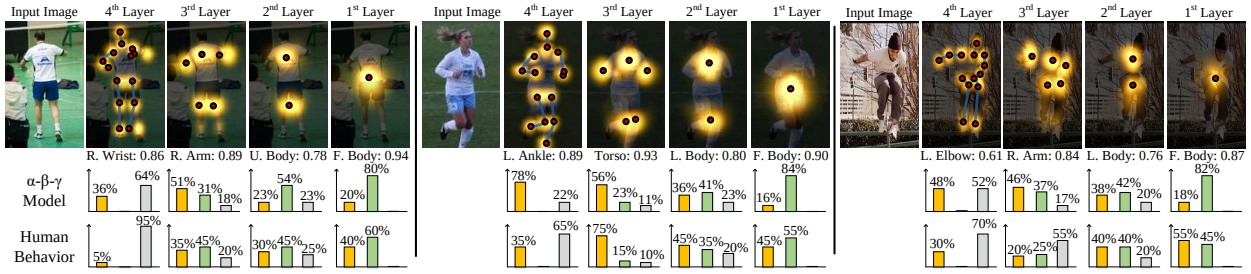


Figure 5.5: **Results of human pose estimation on LSP dataset**. In the first row, we show the predictions of each layer. For each layer, we select one node to demonstrate its prediction score. Then, in the second and third rows, we present the contributions of α, β, and γ processes over such node, which estimated from our model and human behavior.

5.6.1 Cloth Landmark Localization

Fashion landmarks are functional keypoints defined on clothes, such as corners of neckline, cuff [LYL16], which are effective representation for visual fashion understanding. Cloth landmark detection is a good example with inherent structures and obvious components, yet challenging due to background clutters, deformations, and scales.

Dataset. We use Fashion Landmark Detection (FLD) [LYL16][1], which contains totally 123, 016 clothes images. For each image, 8 fashion landmarks (l.&r. collar, l.&r. sleeve, l.&r. waistline, l.&r. hem) are annotated. For each image, cloth bounding box is also annotated.

Network architecture. A three-layer graph $\mathcal{G}$ is derived for representing human cloth (Fig. 5.3 (a)). We build our structured network following $\mathcal{G}$. The first five convolutional stacks of ResNet50 [HZR16] are opted as our underlying network. For preserving detailed spatial information, we modify the last two blocks by changing the strides to 1. Specifications of α-, β-, and γ- networks are listed in Table 5.1. Note that 1×1 convolution layer with *sigmoid* layer is applied to produce final predictions. The principle behind such design is mainly for pursuing large enough receptive field and simplicity. The input images are resized into 224×224. Thus if sliding our network over the input image, we could obtain a 28×28 prediction map of each nodes for their specific tasks.

[1]Available at http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion/LandmarkDetection.html

Methods	4th Layer							3rd Layer			2nd Layer		1st Layer
	Head &Neck	Shoulder (L.&R.)	Elbow (L.&R.)	Wrist (L.&R.)	Hip (L.&R.)	Knee (L.&R.)	Ankle (L.&R.)	Arm (L.&R.)	Leg (L.&R.)	Head &Torso	U.Body	L.Body	F.Body
[WL13]	89.1	78.5	62.5	52.3	85.2	69.6	65.9	-	-	-	-	-	-
[CY14]	91.8	78.2	71.8	65.5	73.3	70.2	63.4	-	-	-	-	-	-
[TJL14]	90.6	79.2	67.9	63.4	69.5	71.0	64.2	-	-	-	-	-	-
[FZL15]	92.4	75.2	65.3	64.0	75.7	68.3	70.4	-	-	-	-	-	-
[YOL16]	90.6	78.1	73.8	68.8	74.8	69.9	58.9	-	-	-	-	-	-
[HR16]	93.4	83.2	77.3	72.1	87.6	79.6	76.8	-	-	-	-	-	-
[WRK16]	94.1	86.0	78.9	76.0	88.7	82.3	77.4	-	-	-	-	-	-
[LLA16]	88.4	76.5	70.6	66.3	75.6	68.7	67.5	67.4	63.2	73.4	74.7	76.2	78.9
[PNZ18]	90.7	79.8	76.8	68.1	78.9	70.2	75.4	76.9	75.1	82.5	84.9	82.0	81.5
α-network	94.4	**86.4**	79.3	75.2	89.1	82.6	77.1	78.4	76.2	83.4	89.1	85.5	84.7
β-network	-	-	-	-	-	-	-	77.6	78.3	83.5	90.7	92.5	91.7
γ-network	92.5	82.1	76.5	71.1	85.7	78.3	72.2	80.3	81.0	82.4	90.2	90.3	-
α-β-γ w/o. share	95.1	85.0	80.6	**78.5**	90.3	83.2	78.6	80.7	79.3	84.2	91.3	93.4	92.4
α-β-γ full	**95.6**	85.3	**81.6**	77.3	**91.3**	**83.7**	**80.5**	**82.5**	**81.3**	**86.5**	**93.4**	**94.9**	**92.8**

Table 5.3: **Comparison of PCKh metric on LSP dataset**. Higher values are better. The best score is marked in **bold**.

Training. For all nodes, ground-truth heatmaps are generated by convolving binary annotation maps with a small Gaussian kernel. For those higher-layer nodes without annotation, such as *upper-body cloth* or *full-body cloth*, we generate their annotation according to child nodes' configurations. This annotation process is similar to [LLA16]. For node v, we would have an output prediction score map $S \in [0,1]^{28\times28}$ and its corresponding ground-truth map $\hat{S} \in [0,1]^{28\times28}$ for a 224×224 training image. Then we adopt Kullback-Leibler Divergence to measure the loss:

$$L(\hat{v}|\theta_v) = D_{KL}(S_v, \hat{S}_v) = \mathbf{1}(\hat{v}) \cdot \sum\nolimits_k^{28\times28} \hat{s}_k \cdot \log \frac{\hat{s}_k}{s_k}. \tag{5.10}$$

where the indicator function $\mathbf{1}(\cdot)$ is employed for remedying missing ground truth locations of the landmarks, in the sense that the error is not propagated back when a landmark is occluded (according to the visibility annotation). Here we drop subscript v for s_k and $\hat{s}_k$ for simplicity.

Performance comparison. We compare α-β-γ network with four deep learning based fashion landmark detectors: [LLQ16, LYL16, YLL17, WXS18]. For all the methods, stan-

dard train/validation/test settings $(83,033/19,992/19,991)$ in FLD dataset are used for fair comparisons. We adopt normalized error (NE) metric suggested by FLD dataset for evaluation. NE refers to the ℓ_2 distance between predicted landmarks and ground-truth in the normalized coordinate space (*i.e.*, divided by the width/height of the image). We report the results in Table 5.2, where the baselines: α-*network*, β-*network*, and γ-*network* indicate the results obtained from α, β, and γ processes independently. α-β-γ *w/o. share* corresponds to the results of α-β-γ network without parameter sharing, equivalent to ignoring the horizontal relations $\mathcal{E}_{rel}$ in graph $\mathcal{G}$. As seen, the proposed structured network outperforms other competitors. Some qualitative results can be found in Fig. 5.4.

Discussion. The improvement would be attributed to the integration of deep learning and graph model. Unstructured models like FashionNet and DFA are hard to model the inherent structures of fashion cloth, which offers strong contextual information about cloth landmark locations. Our solution is more favored due to its structural modeling with underlying graphical representation and powerful joint bottom-up and top-down inference. We can further observe that, α inference performs better for those low-level nodes (*e.g.*, l. collar, r. waistline), while β and γ processes are more informative for high-level nodes like lower-body cloth or full-body cloth. Compared with those explicit junctions, the nodes in higher layers are often companied with more ambiguities, in which sense more complex bottom-up/top-down inference processes are preferred. For α-β-γ *network w/o share*, we can observe a drop of performance. This demonstrates the importance of structure information, and thus verifies our design. Besides, parameter sharing would bring extra advantage of better generalization.

5.6.2 Human Pose Estimation

In this section, we present our structured network for another vision task, human pose estimation, which is a popular vision task requiring both powerful detection of human body parts and effective modeling of relationship among parts.

We compare our methods with several previous pose estimators using graphical structures

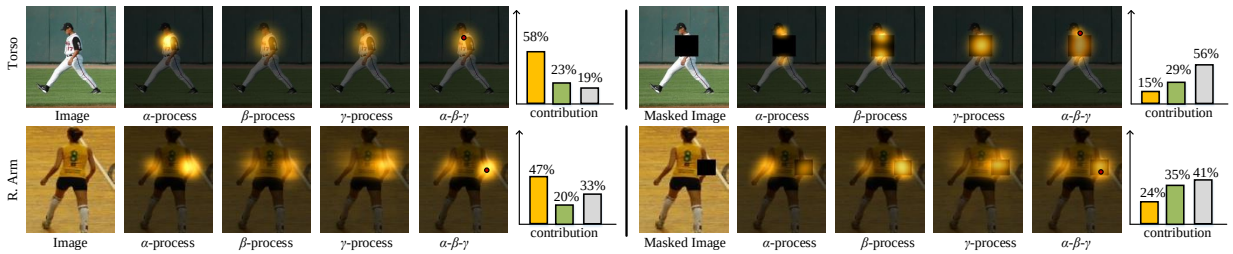


Figure 5.6: **Examining interpretability with masked examples**. See text for more details.

or pure networks, which show the benefits of capturing the interactions between body parts with graphical models. In this experiment, we demonstrate our structured network is able to generate more accurate pose estimations via explicitly and jointly considering bottom-up and top-down information.

Dataset. We use the standard pose estimation benchmark: LSP dataset [JE10][2], containing 11,000 images for training and 1,000 images for testing. The images are of people in various sport poses.

Network architecture. In Fig. 5.3 (b), human pose is represented as a 4-level hierarchical structured network. The bottom level of our hierarchy is comprised of the 14 atomic parts corresponding to the annotated joints. The third level consists of 5 composite parts formed by grouping parts belonging to each of the limbs, a composite part for the head and torso. The second-layer nodes refer to the upper-human body and lower-human body, and the root node presents full-human body. Such settings are similar to previous graphical models [LLA16, RMH14]. The base network of our model is built upon [WRK16]. For consistency, we adopt the same network architectures of α-, β-, and γ- networks as in Table 5.1.

Training. We follow the standard protocol in the area of pose estimation. For each node, a ground-truth confidence map is created by putting Gaussian peaks at ground-truth locations of corresponding part. We infer the ground-truth locations of higher-level parts following the annotation procedure in fashion landmark detection. We also resize the input

[2]Available at http://sam.johnson.io/research/lsp.html

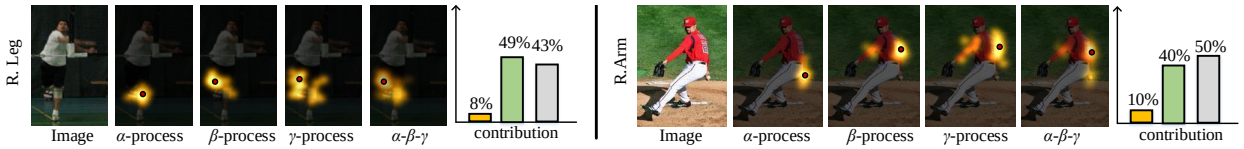


Figure 5.7: **Illustration of bottom-up and top-down inference**. We select one node in the $3rd$ layer of human pose graph and show the predictions from α-, β-, and γ- processes, and draw the distribution of contribution of above processes in the final prediction.

images into 224×224. Then we use ℓ_2 distance to measure the loss, which is widely used in human pose estimation:

$$L(\hat{v}|\theta_v) = D_{\ell_2}(S_v, \hat{S}_v) = \sum\nolimits_k^{28\times 28} \|s_k - \hat{s}_k\|_2, \tag{5.11}$$

where the $S \in [0,1]^{28\times 28}$ and $\hat{S} \in [0,1]^{28\times 28}$ denote the output score map and the ground-truth, respectively.

Performance comparison. For evaluation, we use the PCKh metric [APG14], which is a modification of the Percentage Correct Keypoints (PCK) metric with a matching threshold. We compare the performance of our method with several pose estimators. We also investigate the performance of individual inference processes, and simplified model without horizontal relations. As seen in Table 5.3, our model outperforms other competitors. We visualize detection results in Fig. 5.5.

Discussion. The proposed model offers a powerful tool that has the complementary strengths of neural network and graphics models. It is not limited to CRF-like assumptions, which are widely used in previous graphical pose models. Therefore, α-β-γ network is able to better represent rich internal relations among human body parts. When comparing the performance of individual inference process and our full model, we again get the similar observations that α, β, and γ processes are favored under different scenarios and the integration of three processes would improve final performance.

5.6.3 Study of Post-hoc Interpretability

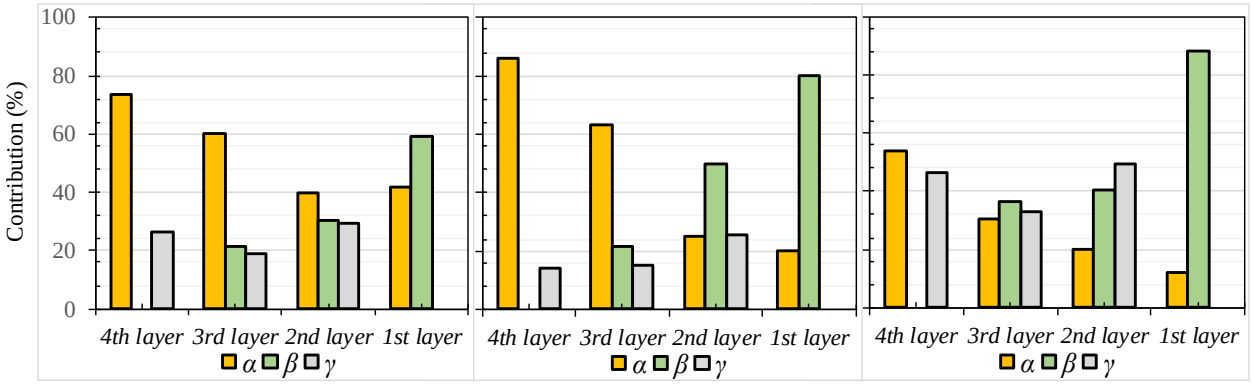
We further explore the post-hoc interpretability conferred by our model, specifically, how the contribution made by different inference processes coincides with human knowledge. The contribution of a process is defined as the ratio between its own weighted prediction and the final score. Taking α process as an example, the contribution of α process for node v can be formulated as:

$$C^{\alpha}(v) = \frac{\theta_v^{c_\alpha} \cdot f_v^{\alpha}(v)}{\theta_v^{c_\alpha} \cdot f_v^{\alpha}(v) + \theta_v^{c_\beta} \cdot f_v^{\beta}(v) + \theta_v^{c_\gamma} \cdot f_v^{\gamma}(v)}. \tag{5.12}$$

We first perform a user study to measure the agreement between human behavior and our model. A corpus of 20 participants (9 female) with diverse backgrounds are recruited to participate in our studies. 100 images were randomly selected from the test set of LSP dataset. For each node, participants were asked to label the most informative inference process. Generally, for the cases that a node can be directly recognized, it is labeled as α label. In the situation that compositional or contextual information are needed, β or γ are annotated accordingly. We average the votes from all the participants as human consensus. Fig. 5.8 (a-b) plot the contribution distributions of the three processes annotated from human and learned via our model, showing that α process is important in low-level nodes, while β and γ processes are relatively strong in high-level nodes. This observation is also verified in previous experiments, that α, β and γ processes are effective in different layers. We also find the contribution distribution of our model is close to human consensus.

In Fig. 5.7, we select one node in the $3rd$ layer of human pose graph and show the prediction scores from α, β, and γ processes and the final score from the fusion of above three processes. We also present the distribution of contribution of above processes in the final prediction. As seen, the combination of α, β, and γ inference processes would get the best results. Quantitatively, our model obtains 61.3%, 50.7% and 31.5% *average precision* (AP) of α, β, and γ processes with human consensus over all nodes, respectively.

We further conduct a counter-factual experiment using data manipulation. For each image, we generate a mask (20×20) to cover certain nodes. Afterwards, we obtain manipulated images with occlusions on body parts and re-estimate human poses (see Fig. 5.6). The statis-



(a) (b) (c)

Figure 5.8: **Numerical study of contributions of three inference processes**: (a) human behavior; (b) performance of α-β-γ network; and (c) α-β-γ network with masked images. We average the scores from same-layer nodes.

tics regarding contributions of three processes are reported in Fig. 5.8 (c). Interestingly, we find that, when a node suffers occlusion, β and γ processes provide more supports in final predictions.

5.7 Summary

In this chapter, we propose a deep structured network for combining hierarchical graph representations with deep learning. The α-β-γ network is capable of modeling rich structures, incorporating top-down/bottom-up inference learned in end-to-end manner. It gains better interpretability via separating explicit inferences from the underlying implicit mechanics of neural network. Performance and interpretability of the proposed model are well demonstrated through extensive experiments on fashion landmark detection and human pose estimation.

CHAPTER 6

Human Parsing using Fashion Grammar

6.1 Introduction

With the rapid development of electronic commerce and the boom of online shopping, visual clothing analysis has attracted lots of interests in computer vision. More recently, benefited from the availability of large-scale fashion datasets, deep learning based models gained astonishing success in this area, such as clothing item retrieval [HHL15, HFC15], and fashion image classification [SFM15, LLQ16, LKZ17], to name a few.

In this chapter, we address two key problems in visual fashion analysis, namely fashion landmark localization and clothing category classification. The success of previous deep learning based fashion models [HFC15, LLQ16, LXS15, CBR17] has proven the potential of applying neural network in this area. However, few of them attacked how to inject high-level human knowledge (such as geometric relationships among landmarks) into fashion models. In this chapter, we propose a fashion grammar model that combines the learning power of neural network and domain-specific grammars that capture the kinematic and symmetric relations between clothing landmarks. For modeling the massage passing process over fashion grammars, we introduce a novel network architecture, Bidirectional Convolutional Recurrent Neural Network (BCRNN), which is flexible to our tree-structured models and generates more reasonable landmark layouts with global grammar constraints. Crucially, our whole deep grammar model is fully differentiable and can be trained in end-to-end manner.

This work also proposes two important attention mechanisms for boosting fashion image classification. The first one is *fashion landmark-aware*, which leverages the strong representation ability of fashion landmarks and can be learned in supervised manner. This attention

is able to generate landmark-aligned clothing features, which makes our model look for the informative semantic parts of garments. The second attention is *clothing category-driven* and trained in goal-driven way. Such attention mechanism learns to directly enhance task-related features and thus improves the classification performance. The attentions provide the model with more robust clothing representations and filter out useless information.

Comprehensive evaluations on two large-scale datasets [LLQ16, LYL16] demonstrate that our fashion grammar model outperforms the state-of-the-arts. Additionally, we experimentally demonstrate that our BCRNN based fashion grammars and attention modules give non-trivial improvements.

Contribution. Our main contribution is three-fold: i) We develop a deep grammar network to encode a set of knowledge over fashion clothes. The fashion knowledge, represented in grammar format, explicitly expresses the relations (*i.e.*, kinematics, and symmetry) of fashion landmarks, and serve as basis for constructing our fashion landmark detection module. ii) We present Bidirectional Convolutional Recurrent Neural Network (BCRNN) for approaching message passing over the suggested fashion grammars. The chain-structure topology of BCRNNs efficiently represents the rich relations of clothes, and our fashion model is fully differentiable which can be trained in end-to-end manner. iii) We introduce two attention mechanisms, one is landmark-aware and domain-knowledge-involved, and the other one directly focuses on the category relevant image regions and can be learned in goal driven manner.

6.2 Related Work

Visual fashion understanding has drawn lots of interests recently, due to its wide spectrum of human-centric applications such as clothing recognition [CGG12, LLQ16, HG17, HWH17, ASG17], retrieval [WZ11a, HHL15, LSL12, YHB13], recommendation [KYB14, SFM15, LLQ16, HWJ17], parsing [YKO12, YLL14] and fashion landmark detection [LLQ16, LYL16]. *Earlier fashion models* [CGG12, KYB14, WZ11a, LSL12] are mostly relied on handcrafted features (*e.g.*, SIFT, HOG) and seek for powerful clothing representations,

such as graph models [CXL06], contextual information [SWH11, HHL15], general object proposals [HHL15], human parts [SWH11, LSL12], bounding boxes [CHF15] and semantic masks [YHB13, YKO12, YLL14, LXS15, GLZ17].

With the availability of large-scale fashion datasets [SFM15, LLQ16, LYL16], *deep learning based models* [HFC15, LLQ16, LYL16, LXS15, LKZ17, CBR17] were proposed and outperformed prior work by a large margin. In particular, Huang *et al.* [HFC15] introduced a Dual Attribute-aware Ranking Network (DARN) for clothing image retrieval. Liu *et al.* [LLQ16] proposed a branched neural network, for simultaneously performing clothing retrieval, classification, and landmark detection. More recently, in [LYL16], a deep learning based model was designed as a combination of three cascaded networks for gradually refining fashion landmark estimates. Yan *et al.* [YLL17] combined selective dilated convolution and recurrent spatial transformer for localizing cloth landmarks in unconstrained scenes. The success of those deep learning based fashion models demonstrate the strong representation power of neural network. However, they barely explore the rich domain-specific knowledge of clothes. In comparison, we propose a deep fashion grammar network that incorporates both powerful learning capabilities of neural networks and high-level semantic relations in visual fashion.

Grammar models in computer vision are powerful tool for modeling high-level human knowledge in specific domains, such as the decompositions of scenes [HZ09, LCK14, QZH18], semantic relations between human and objects [ZZ11, QHW17], dependencies between human parts [XLZ13, FXW18], and the compatibility relations between human attributes over human hierarchy [XLL16, XLQ17, PNZ18]. They are a natural choice for modeling rich relations and diverse structures in this world. Grammar models allow an expert inject domain-specific knowledge into the algorithms, thus avoiding local ambiguities and hard decisions [AT07, PNZ18]. In this chapter, we first propose two fashion grammars that account for dependent and symmetric relations in clothes. In particular, we ground these knowledge in a BCRNN based deep learning model which can be end-to-end trained with back-propagation.

Attention mechanism in computer vision has been popular in the tasks of image cap-

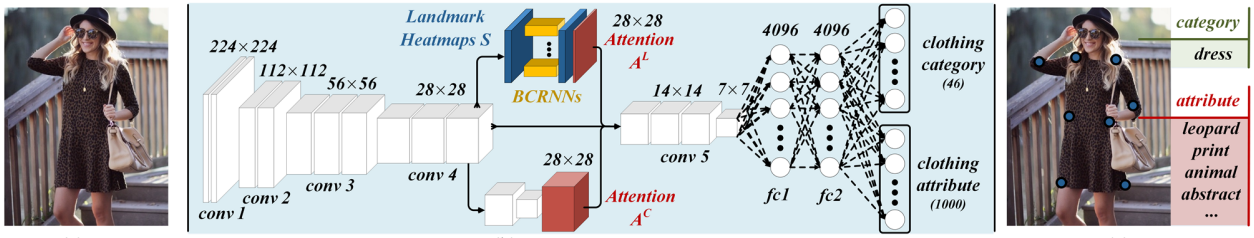


Figure 6.1: **Illustration of the proposed Attentive Fashion Grammar Network.** (a) Input fashion image. (b) Network architecture of our deep fashion model. A set of BCRNNs (yellow cubes) are established for capturing kinematics and symmetry grammars as global constraints for detecting clothing landmarks (blue cubes), detailed in §6.3.1. Fashion landmark-aware attention A^L and clothing category-driven attention A^C (red cubes) are further incorporated for enhancing clothing features and improving clothing category classification and attribute estimation (§6.3.2). (c) Results for clothing landmark detection, category classification and attribute estimation.

tion [XBK15], Visual Question Answering (VQA) [SSH16, YHG16], object detection [CLY15, XXY15] and image recognition [WS18, CYW16, WJQ17, JSZ15]. Those methods show that top-down attention mechanism is effective as it allows the network to learn which regions in an image to attend to solve their tasks. In this chapter, two kinds of attentions, namely category-directed and landmark-aware attentions, are proposed. As far as we know, no attention mechanism has been applied to the feed-forward network structure to achieve state-of-the-art results in visual fashion understanding tasks. Besides, in contrast to previous part-based fashion models [SWH11, LSL12, LLQ16, LYL16] with hard deterministic constraints in feature selection, our attentions act as soft constraints and can be learned in a stochastic way from data.

6.3 Our Approach

We first describe our fashion grammar network for fashion landmark detection (§6.3.1). Then we introduce two attention mechanisms for clothing image classification (§6.3.2).

6.3.1 Fashion Grammar Network for Fashion Landmark Detection

Problem Definition. Clothing landmark detection aims to predict the positions of K functional key points defined on the fashion items, such as the corners of neckline, hemline, and cuff. Given an image I, the goal is to predict cloth landmark locations L:

$$L = \{L_k : k = 1, \ldots, K\},\ L_k \in \mathbb{R}^2, \tag{6.1}$$

where L_k can be any pixel locations (u, v) in an image.

Previous fashion landmark methods [LLQ16, LYL16] formulate this problem as regression. They train a deep learning model and use a function $f(I;\theta) \in \mathbb{R}^{2K}$ which for an image I directly regresses to a landmark vector. They minimize the mean square error over N training samples:

$$f^* = \min_f \frac{1}{N} \sum\nolimits_{n=1}^{N} \|f(I^n;\theta) - L^n\|_2. \tag{6.2}$$

However, recent studies in pose estimation [TJL14, PCZ15] demonstrate this regression is highly non-linear and very difficult to learn directly, due to the fact that only one single value needs to be correctly predicted.

In this work, instead of regressing landmark positions L directly, we learn to predict a confidence map of positional distribution (*i.e.*, heatmap) for each landmark, given the input image. Let $S_k \in [0,1]^{w \times h}$ and $\hat{S}_k \in [0,1]^{w \times h}$ denote the predicted heatmap and the ground-truth heatmap (with size of $w \times h$) for the k-th landmark, respectively, our fashion network is learned as a function $f'(I;\theta') \in [0,1]^{w \times h \times K}$, via penalizing following pixel-wise mean squared differences,

$$\begin{aligned} f^* &= \min_{f'} \frac{1}{N} \sum\nolimits_{n=1}^{N} \sum\nolimits_{k=1}^{K} \mathcal{L}(f'(I^n;\theta'), L_k^n), \\ \mathcal{L}(f'(I^n;\theta'), L_k^n) &= \sum\nolimits_{u=1}^{w} \sum\nolimits_{v=1}^{h} \|S_k^n(u,v) - \hat{S}_k^n(u,v)\|_2. \end{aligned} \tag{6.3}$$

The ground-truth heatmap $\hat{S}_k$ is obtained by adding a 2D Gaussian filter at the ground-truth location L_k.

Fashion Grammar. We consider a total of eight landmarks (*i.e.*, K=8), namely, *left/right collar*, *left/right sleeve*, *left/right waistline*, and *left/right hem*, following previous settings [LLQ16, LYL16]. The natural of clothes that rich inherent structures are involved

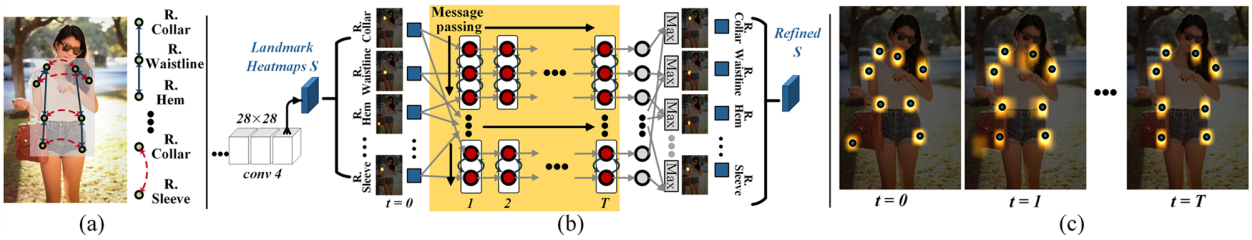


Figure 6.2: (a) **Illustration of our fashion grammars**, where green circles indicate ground-truth cloth landmarks, blue and red lines correspond to kinematics and symmetry grammars, respectively. (b) **Illustration of our message passing over fashion grammars**, where the blue rectangles indicate heatmaps of landmarks, and the red circles indicate BCRNN units. Within a certain BCRNN, we perform message passing over fashion grammars (one time, two directions). With stacked of BCRNNs, the messages are iteratively updated and refined landmark estimations are generated. (c) **Illustration of the refined estimations by message passing** over our fashion grammars. With the efficient message passing over grammar topology, our fashion network is able to predict more kinematically and symmetrically possible landmark layouts with high-level constraints.

in this task, motivates us to reason the positions of landmarks in a global manner. Before going deep into our grammar network, we first detail our grammar formulations that reflect high-level knowledge of clothes. Basically, we consider the two types of fashion grammars:

- *Kinematics grammar* $\mathcal{R}^{\mathcal{K}}$ describes kinematic relations between clothing landmarks.

We define 4 kinematic grammars to represent the constraints among kinematically connected clothing parts:

$$\begin{aligned}
&\mathcal{R}_1^{\mathcal{K}} : l.\ \ collar \leftrightarrow l.\ \ waistline \leftrightarrow l.\ \ hem, \\
&\mathcal{R}_2^{\mathcal{K}} : l.\ \ collar \leftrightarrow l.\ \ sleeve, \\
&\mathcal{R}_3^{\mathcal{K}} : r.\ \ collar \leftrightarrow r.\ \ waistline \leftrightarrow r.\ \ hem, \\
&\mathcal{R}_4^{\mathcal{K}} : r.\ \ collar \leftrightarrow r.\ \ sleeve.
\end{aligned} \tag{6.4}$$

Such grammar focuses on the clothing landmarks that connected in a human-parts kinematic chain, which satisfies human anatomical and anthropomorphic constraints.

- *Symmetry grammar* $\mathcal{R}^{\mathcal{S}}$ describes bilateral symmetric property of clothes. Symmetry

of clothes is defined as the right and left sides of the cloth being mirrored reflections of each other. We consider 4 symmetric relations between clothing landmarks:

$$\begin{aligned}
&\mathcal{R}_1^{\mathcal{S}} : l.\ \textit{collar} \leftrightarrow r.\ \textit{collar},\\
&\mathcal{R}_2^{\mathcal{S}} : l.\ \textit{sleeve} \leftrightarrow r.\ \textit{sleeve},\\
&\mathcal{R}_3^{\mathcal{S}} : l.\ \textit{waistline} \leftrightarrow r.\ \textit{waistline},\\
&\mathcal{R}_4^{\mathcal{S}} : l.\ \textit{hem} \leftrightarrow r.\ \textit{hem}.
\end{aligned} \tag{6.5}$$

Message Passing over Fashion Grammar. As illustrated in Fig. 6.2 (a), our proposed grammars upon cloth landmarks constitute a graph, where vertices specifying cloth landmark heatmaps and edges describing possible connections among vertices. To infer the optimal landmark configuration, message passing [YOL16, COL16] is favored on such loopy structures. To simulate this process, we make an approximation by performing message passing on each grammar independently and merging the output afterwards to disentangle the loopy structure.

More specifically, within the chain structure of grammar $\mathcal{R}$, the passing process is performed iteratively for each node i, consisting of two phases: the message passing phase and the readout phase. The message passing phase runs for T iterations and is defined w.r.t. message function $M(\cdot)$ and vertex update function $U(\cdot)$. In each iteration, hidden states h_i of node i is updated by computing messages coming from its neighbors j, that is,

$$\begin{aligned}
m_i &\leftarrow \sum\nolimits_{j \in \mathcal{N}(i)} M(h_j),\\
h_i &\leftarrow U(m_i),
\end{aligned} \tag{6.6}$$

where $\mathcal{N}(i)$ denotes neighbors of vertex i specified in the grammar $\mathcal{R}$.

The second phase, *i.e.*, the readout phase, infers the marginal distribution (*i.e.*, heatmaps) for each node i using h_i and readout function $\Gamma(\cdot)$, namely,

$$y_i = \Gamma(h_i). \tag{6.7}$$

Implementation with Recurrent Neural Network. For implementing above message passing process over grammar topology, we introduce Bidirectional Convolutional Re-

current Neural Network (BCRNN) (see Fig. 6.3), which is achieved by extending classical fully connected RNNs with convolution operation [SCW15, SP97].

In a high level, the bi-directionality and recurrent nature of BCRNN are favored to simulate the message passing over the grammar neighborhood system. Additionally, with the convolution operation, our model could preserve the spatial information of convolutional feature map and is able to produce pixel-wise heatmap prediction.

All the proposed grammars consist of short chain structures (*i.e.*, at most 3 vertices involved) [GSR17], connoting that every node i in the grammar can at most have two neighbors (*i.e.*, previous node i-1 and post node i+1). Specifically, given a BCRNN, message functions $M(\cdot)$ for node i (in forward/backward directions) are represented as

$$\begin{aligned} m_i^f &= M^f(h_{i-1}^f) = W^f * h_{i-1}^f, \\ m_i^b &= M^b(h_{i+1}^b) = W^b * h_{i+1}^b, \end{aligned} \tag{6.8}$$

where $*$ denotes the convolution operator, $M^f(\cdot)$ and $M^b(\cdot)$ denote the forward and backward message function, h^f and h^b refer to the hidden states inferred from forward and backward neighbors, respectively. The hidden state h_i is thus updated accordingly

$$\begin{aligned} h_i^f &= U(m_i^f) = \tanh(m_i^f + b_h^f) \ , \\ h_i^b &= U(m_i^b) = \tanh(m_i^b + b_h^b) \ , \end{aligned} \tag{6.9}$$

where b_h^f and b_h^b refer to the bias term used in forward and backward inference, respectively. The readout function $\Gamma(\cdot)$ is defined as

$$y_i = \Gamma(h_i) = \sigma(W^x * x_i + h_i^f + h_i^b), \tag{6.10}$$

where σ is the soft-max function, x_i is the input generated by the base convolution network.

We illustrate the implementation of message passing mechanism in Fig. 6.2 (b). By implementing massage passing with BCRNN, our network maintains the fully differentiability and obtains decent results by exchanging information along the fashion grammars.

Network Architecture. Our fashion network is based on VGG-16 architecture [SZ15]. First, we employ features from *conv4-3* layer (the last convolution layer of the forth block)

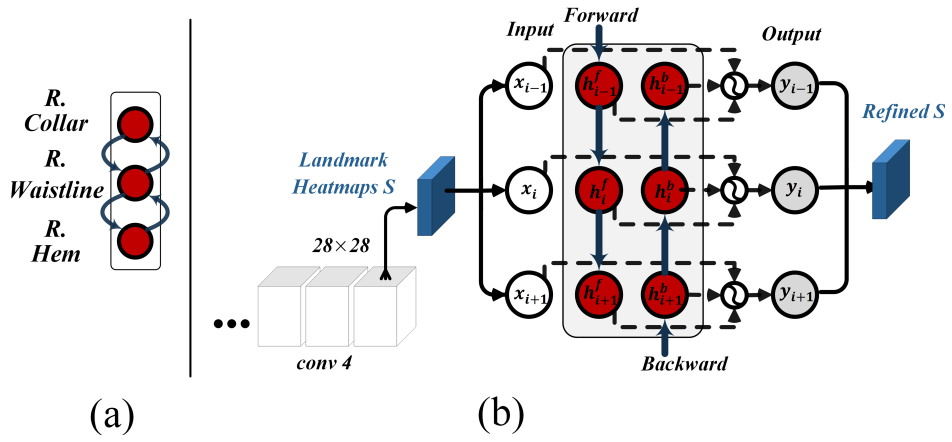


Figure 6.3: (a) **BCRNN with a fashion grammar**. (b) **Architecture of BCRNN**. With the input landmark predictions, the corresponding BCRNN is used for approaching massage passing over the grammar topology (a), resulting more reasonable landmark estimations. See §6.3.1 for more details.

to produce K landmark heatmaps with *sigmoid* activation. Due to the max-pooling operation, we achieve $\times 8$ down-scaled heatmaps. We employ eight BCRNNs to simulate message passing procedures among cloth landmarks in each chain. A grammar BCRNN takes initial heatmaps and features from *conv4-3* as inputs and the forward process correspond to a passing process (on two directions). Generally, message passing takes several iterations to converge, while in practice three iterations are sufficient to generate satisfactory results (see more detailed discussions in §6.4.4). In implementation, we stack three BCRNNs (*i.e.*, T=3) for each grammar (totally 3×8 BCRNNs for all the grammars) and updated estimation via a max-pooling of predicted heatmaps from corresponding BCRNNs at the end of each stack.

6.3.2 Attention Modules for Clothing Category Classification

Previous studies in VQA [SSH16, YHG16] and object detection [CLY15, XXY15] indicate that top-down attention is good at selecting task-related locations and enhancing important features. As demonstrated in Fig. 6.1(b), we incorporate our fashion model with two kinds of attentions, namely fashion landmark-aware attention and category-driven attention, to improve the classification accuracy.

Fashion Landmark-Aware Attention. Clothing landmarks are keypoints centered in functional parts of clothes [LLQ16, LYL16]. Such representation actually provides useful information about fashion styles. Based on this observation, we introduce a landmark-aware attention mechanism that constrains our fashion model to concentrate on functional clothing regions.

For the predicted heatmaps $\{S_i\}_{i=1}^{K}$, we apply cross-channel average-pooling operation to generate a 28×28 *weight map*, A^L:

$$A^L = \frac{1}{K} \sum\nolimits_{i=1}^{K} S_i, \tag{6.11}$$

where $A^L \in [0,1]^{28\times 28}$. We call A^L as the landmark-aware attention. Let $F \in \mathbb{R}^{28\times 28\times 512}$ denote features obtained from *conv4-3* layer in VGG-Net, F is further updated by the landmark-aware attention A^L with same spatial dimensions, that is,

$$G_c^L = A^L \circ F_c, \quad c \in \{1, \ldots, 512\}, \tag{6.12}$$

where $\circ$ denotes the Hadamard product, F_c denotes the 2D tensor from the c-th channel of F, and G^L denotes the refined feature map. The feature is re-weighted by the landmark-aware attention and has the same size as F. Here the attention A^L works as a feature selector which produces fashion landmark aligned features. In contrast to spatial attention [JSZ15], our attention is learned in a supervised manner and encodes semantic and contextual constraints.

Clothing Category-Driven Attention. Our landmark-aware attention enhances the features from functional regions of clothes. However, such mechanism may be insufficient to discover all the informative locations to accurately classify diverse fashion categories and attributes. Inspired by recent advances in attention models [CYW16, WJQ17], we further propose a cloth category-driven attention A^C, which is goal directed and learned in top-down manner.

Given features F from the *conv4-3* layer, we apply a *bottom-up top-down* network [LSD15, NYD16] (*e.g.*, $\times 2$ down-pooling$\rightarrow$3$\times$3 conv$\rightarrow\times 2$ down-pooling$\rightarrow$3$\times$3conv$\rightarrow\times 4$ up-pooling) to learn a global attention map $A^C \in [0,1]^{28\times 28\times 512}$. The attention features are first pooled down to a very low resolution 7$\times$7, then are $\times 4$ up-sampled. Thus the attention module

gains a large receptive field covers all the fashion image, but is the same size as the feature map. For each position in A^C, *sigmoid* function is applied to shrink the attention values, ranging from $[0, 1]$. Afterwards, we use the attention A^C to softly weight output features F:

$$G^C = A^C \circ F. \tag{6.13}$$

With the bottom-up top-down network, the attention obtains a large receptive field and directly enhances the task-related features from a global view. Such attention facilitates our model to learn more discriminative representations for fashion style recognition. Different from our landmark-aware attention, the category-driven attention A^C is goal-directed and learned without explicit supervision. Visualization of our attention mechanisms can be found in Fig. 6.4.

Network Architecture. With the feature F from the *conv4-3* layer, we consider landmark-aware attention $A^L \in [0, 1]^{28\times28}$ and clothing category-driven attention $A^C \in [0, 1]^{28\times28\times512}$ simultaneously:

$$G_c = (1 + A^L + A^C_c) \circ F_c, \quad c \in \{1, \ldots, 512\}. \tag{6.14}$$

Such design is inspired by works in residual learning [HZR16, WJQ17]. If the attention models can be constructed as *identical mapping*, the performance should be no worse than its counterpart without attention. We offer more detailed analyses for our attention modules in §6.4.4.

As seen, the updated feature G has the same size of the feature F from *conv4-3* layer. Thus the rest layers (*pooling-4*, *conv5*s, *pooling-5*, and *fc*s) of VGG-Net can be stacked for final cloth image classification. Our attention mechanisms incorporate semantic information and global information into network and help constrain the network to focus on important clothing regions. Refined features are further used to learn classifiers on foreground clothing regions (please see Fig. 6.1(b)). Our whole fashion network is fully differentiable and can be trained end-to-end.

Methods	Category		Texture		Fabric		Shape		Part		Style		All	
	top-3	top-5	top-3	top-5	top-3	top-5	top-3	top-5	top-3	top-5	top-3	top-5	top-3	top-5
WTBI [CGG12]	43.73	66.26	24.21	32.65	25.38	36.06	23.39	31.26	26.31	33.24	49.85	58.68	27.46	35.37
DARN [HFC15]	59.48	79.58	36.15	48.15	36.64	48.52	35.89	46.93	39.17	50.14	66.11	71.36	42.35	51.95
FashionNet [LLQ16]	82.58	90.17	37.46	49.52	39.30	**49.84**	39.47	48.59	**44.13**	54.02	66.43	73.16	45.52	54.61
Lu *et al.* [LKZ17]	86.72	92.51	-	-	-	-	-	-	-	-	-	-	-	-
Corbiere *et al.* [CBR17]	86.30	92.80	**53.60**	63.20	39.10	48.80	50.10	59.50	38.80	48.90	30.50	38.30	23.10	30.40
Ours	**90.99**	**95.78**	50.31	**65.48**	**40.31**	48.23	**53.32**	**61.05**	40.65	**56.32**	**68.70**	**74.25**	**51.53**	**60.95**

- Detailed results are not available.

Table 6.1: **Quantitative results for category classification and attribute prediction on the DeepFashion-C dataset** [LLQ16]. Higher values are better. The best scores are marked in **bold**.

Methods	L.Collar	R.Collar	L.Sleeve	R.Sleeve	L.Waistline	R.Waistline	L.Hem	R.Hem	Avg.
FashionNet [LLQ16]	.0854	.0902	.0973	.0935	.0854	.0845	.0812	.0823	.0872
DFA [LYL16]	.0628	.0637	.0658	.0621	.0726	.0702	.0658	.0663	.0660
DLAN [YLL17]	.0570	.0611	.0672	.0647	.0703	.0694	.0624	.0627	.0643
Ours	**.0415**	**.0404**	**.0496**	**.0449**	**.0502**	**.0523**	**.0537**	**.0551**	**.0484**

Table 6.2: **Quantitative results for clothing landmark detection on the DeepFashion-C dataset** [LLQ16] with normalized error (NE). Lower values are better. The best scores are marked in **bold**.

6.4 Experiments

In this section, we evaluate the performance of the proposed fashion model on two large-scale fashion datasets, DeepFashion: Category and Attribute Prediction Benchmark (DeepFashion-C) [LLQ16] and Fashion Landmark Dataset (FLD) [LYL16]. Then ablation study is performed for offering more detailed exploration for the proposed approach.

6.4.1 Datasets

DeepFashion-C [LLQ16][1] is a large collection of 289, 222 fashion images with comprehensive annotations. Those images are collected from shopping websites and Google image search engine. Each image in this dataset is extensively labeled with 46 clothing categories, 1, 000 attributes, 8 landmarks and bounding box. The attributes are further categorized into five groups, characterizing texture, fabric, shape, part, and style, respectively. Based on this dataset, we extensively examine the performance of our deep fashion model in fashion landmark detection, clothing category and attribute classification.

FLD [LYL16][2] is collected for fashion landmark detection. It contains 123, 016 clothing images, with diverse and large pose/zoom-in variations. For each image, the annotations for 8 fashion landmarks are offered. In our experiments, we use this dataset to only evaluate fashion landmark detection, as no garment category annotations are provided.

6.4.2 Experiments on DeepFashion-C Dataset

Experimental Setup. We follow the settings in DeepFashion-C [LLQ16] for training and testing. More specifically, 209, 222 fashion images are used for training and 40, 000 images are used for validation. The evaluation is performed on the remaining 40, 000 images. For training and testing, following [LLQ16, LKZ17], we crop each image using ground truth bounding box. For category classification, we employ the standard top-k classification accuracy as evaluation metric. For attribute prediction, our measuring criteria is the top-k recall rate following [LLQ16], which is obtained by ranking the 1, 000 classification scores and determine how many attributes have been matched in the top-k list. For clothing fashion detection, we adopt normalized error (NE) metric [LYL16] for evaluation. NE refers to the ℓ_2 distance between predicted landmarks and ground-truth in the normalized coordinate space (*i.e.*, normalized with respect to the width/height of the image).

[1]Available at http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion/AttributePrediction.html

[2]Available at http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion/LandmarkDetection.html

Methods	L.Collar	R.Collar	L.Sleeve	R.Sleeve	L.Waistline	R.Waistline	L.Hem	R.Hem	Avg.
FashionNet [LLQ16]	.0784	.0803	.0975	.0923	.0874	.0821	.0802	.0893	.0859
DFA [LYL16]	.048	.048	.091	.089	-	-	.071	.072	.068
DLAN [YLL17]	.0531	.0547	.0705	.0735	.0752	.0748	.0693	.0675	.0672
Ours	**.0463**	**.0471**	**.0627**	**.0614**	**.0635**	**.0692**	**.0635**	**.0527**	**.0583**

- Detailed results are not released.

Table 6.3: **Quantitative results for clothing landmark detection on the FLD dataset** [LYL16] with normalized error (NE). Lower values are better. The best scores are marked in **bold**.

Implementation Details. Our network is built upon VGG-16 with fashion grammar BCRNNs (§6.3.1) and attention modules (§6.3.2). We resize all the cropped images into 224×224. Thus our network would generate eight 28×28 heatmaps for clothing landmarks. We replace the last fully connected layer by two branched fully connected layers for fashion category classification and attribute estimation. In DeepFashion-C dataset, each image receives one category label and multiple attribute labels (average 3.3 attributes per image). For category classification, we apply 1-of-K softmax loss for training the branch of fashion category. For training the other branch of attribute prediction, we apply asymmetric weighted cross-entropy loss [MYS15], due to the data unbalance between positive and negative samples.

Our model is implemented in Python with the help of TensorFlow back-end, and trained with Adam optimizer. For the BCRNNs and category-related attention module, we use 3×3 kernel for all the convolution operations. In each training iteration, we use a mini-batch of 10 images, which are randomly sampled from DeepFashion-C dataset. We first pre-train the former four convolution blocks of our network with cloth landmark detection with two epochs. Then our whole model is trained with ten epochs. The learning rate is set as 0.0001 and is decreased by a factor of 10 every two epochs. We perform early-stopping without improvements on the validation set. The entire training procedure takes about 40 hours with a single NVIDIA TITAN X GPU and a 4.0 GHz Intel processor with 32GB memory.

Figure 6.4: **Clothing category classification results and visualization of attention mechanisms** on DeepFashion-C dataset [LLQ16]. The correct predictions are marked in green and the wrong predications are marked in red. Best viewed in color. For category-aware attention, we randomly select attentions from 2 channels for visualization.

Performance Evaluation. For category classification and attribute prediction, we compare our method with five recent deep learning models [CGG12, HFC15, LLQ16, LKZ17, CBR17] that showed compelling performance in clothes recognition and human attribute classification. For cloth landmark detection, we compare our model with three top-performing deep learning models [LLQ16, LYL16, YLL17]. Note that the results are biased towards [LLQ16], as it is pre-trained with $300,000$ images from DeepFashion and fine-tuned on the DeepFashion-C. For the model [LYL16], the training settings follow the standard protocol in DeepFashion-C. For unconstrained landmark detection model [YLL17], which is reimplemented according to the authors' descriptions, we use the cropped fashion images as inputs for the sake of fair comparison.

Table 6.1 summarizes the performance of different methods on clothing category classification and attribute prediction. As seen, the proposed fashion model achieves the best score on clothing category classification (top-3: 90.99, top-5: 95.78) and the best average score over all attributes (top-3: 51.53, top-5: 60.95). In Table 6.2 we present comparison results with other models [LLQ16, LYL16, YLL17] for clothing landmark detection. Our total NE score achieves state-of-the-art at 0.0484, which is much lower than the closest competitor (0.0643), and it is noteworthy that our method consistently improves the accuracy in all landmarks.

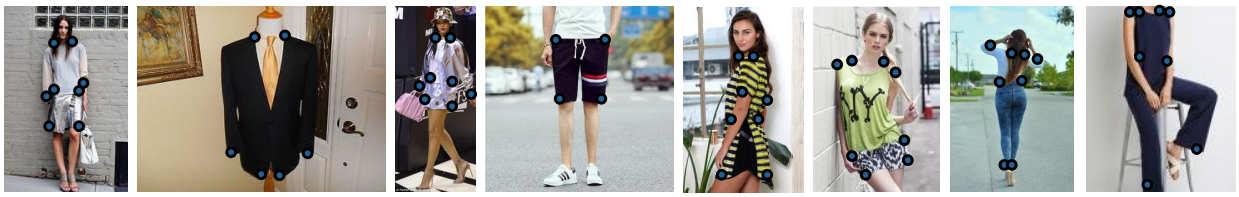


Figure 6.5: **Visual results for clothing landmark detection** on DeepFashion-C [LLQ16] (first row) and FLD [LYL16] (bottom row). The detected landmarks are marked in blue circles. Best viewed in color.

6.4.3 Experiments on FLD Dataset

Experimental Setup. FLD dataset [LYL16] is specially designed for fashion landmark detection. Each image in this dataset is labeled with eight landmarks. With dataset, we study the performance of deep fashion model on fashion landmark detection. Following the protocol in FLD, $83,033$ images and $19,992$ fashion images are used for training and validating, $19,991$ images are used for testing. NE metric suggested by FLD is used for evaluation. The images are also cropped according to the available bounding boxes.

Implementation Details. Since we only concentrate on fashion landmark detection. We preserve the former four convolution blocks (without *pooling4*) and our fashion grammar BCRNNs, which are used for estimating heatmaps for landmarks. 3×3 convolution kernels are also used in BCRNNs. Other settings are similar to the ones used for DeepFashion-C dataset in § 6.4.2.

Performance Evaluation. We compare our model with FashionNet [LLQ16], DFA [LYL16] and DLAN [YLL17]. For sake of fair comparison, we train FashionNet [LLQ16] and DLAN [YLL17] following standard train/val/test settings in FLD. For DFA, we preserve their original results reported in [LYL16]. But the results are biased for DFA, since it's trained with extra clothing labels (upper-/lower-/whole-body clothes).

In Table 6.3, we report the comparison results on the FLD dataset with NE score. Our model again achieves state-of-the-art at 0.0583 and consistently outperforms other competitors on all of the fashion landmarks. Note that our method achieves such high accuracy without any pre-processing (*e.g.*, [LYL16] groups cloth images into different clusters and

Variants	DeepFashion-C		FLD	
	NE ↓	ΔNE ↓	NE ↓	ΔNE ↓
Ours (iteration 3)	**.0484**	-	**.0583**	-
Ours w/o $\mathcal{R}^K$	.0525	.0041	.0659	.0076
Ours w/o $\mathcal{R}^S$	.0538	.0054	.0641	.0058
Ours w/o $\mathcal{R}^K$ & $\mathcal{R}^S$	.0615	.0131	.0681	.0098
Ours-iteration 1	.0579	.0095	.0657	.0074
Ours-iteration 2	.0512	.0028	.0632	.0049

Table 6.4: **Ablation study for the effect of fashion grammars and message passing** on DeepFashion-C [LLQ16] and FLD [LYL16] datasets.

considers extra clothing labels). Sampled landmark detection results are presented in Fig. 6.5.

6.4.4 Ablation Study

In this section, we perform an in-depth study of each component in our deep fashion network.

Effectiveness of Fashion Grammars and Message Passing. We first examine the effectiveness of our fashion grammars, which are models via BCRNNs. In §6.3.1, we consider two types of grammars that account for kinematic dependencies $\mathcal{R}^K$ and symmetric relations $\mathcal{R}^S$, respectively. Three baselines are considered:

- *Ours w/o $\mathcal{R}^K$:* training our model without considering kinematics grammar $\mathcal{R}^K$.
- *Ours w/o $\mathcal{R}^S$:* training our model without considering symmetry grammar $\mathcal{R}^K$.
- *Ours w/o $\mathcal{R}^K$&$\mathcal{R}^S$:* training our model without considering kinematics grammar $\mathcal{R}^K$ and symmetry grammar $\mathcal{R}^K$.

For accessing the effect of iterative message passing over grammars, we report two baselines: *Ours-iteration 1*, *Ours-iteration 2*, which correspond to the results from different passing iterations. The final results (baseline *Ours*) can be viewed as the results in the third passing iteration.

Variants	Category		Attribute	
	top-3 ↑	top-5 ↑	top-3 ↑	top-5 ↑
Ours (w/ A^L & A^C)	**90.99**	**95.78**	**51.53**	**60.95**
Ours w/o A^L	85.27	91.32	48.29	56.65
Ours w/o A^C	87.75	93.67	49.93	58.78
Ours w/o A^L & A^C	83.23	89.51	43.28	53.54

Table 6.5: **Ablation study for the effectiveness of attention mechanisms** on DeepFashion-C [LLQ16] dataset.

We carry out experiments on the DeepFashion-C [LLQ16] and FLD [LYL16] datasets with landmark detection task, and measure the performance using normalized error (NE). Table 6.4 shows the performance of each of the baselines described above. We can observe that fashion grammars provides domain-specific knowledge for regularizing the landmark outputs, boosting further the results (0.0615→0.0484 on DeepFashion-C, 0.0681→0.0583 on FLD). In addition, both kinematics and symmetry grammars contribute the improvement. We also observe the massage passing is able to gradually improve the performance.

Effectiveness of Attention Mechanisms. Next we study the influence of our attention modules. In §6.3.2, we consider two kinds of attentions, namely landmark-aware attention A^L and cloth category-driven attention A^C, for enhancing landmark-aligned and category-related features. Three variants derived from our method are considered:

- *Ours w/o* A^L: training our model without considering landmark-aware attention A^L.
- *Ours w/o* A^C: training our model without considering cloth category-driven attention A^C.
- *Ours w/o* A^L & A^C: training our model without considering landmark-aware attention A^L and cloth category-driven attention A^C.

We experiment on the DeepFashion-C dataset with tasks of cloth category classification and fashion attribute estimation, and measure the performance using the top-k accuracy and top-k recall. As evident in Table 6.5, by disabling attentions A^C and A^L, we observe

significant drop of performance, on both tasks. This suggests that our attention models indeed improve the discriminability of deep learning features. When enabling A^C or A^L attention module, we can achieve better performance. The best performance is achieved via combining A^C and A^L.

6.5 Summary

In this chapter, we proposed a knowledge-driven and attention-involved fashion model. Our model extended neural network with domain-specific grammars, learning to a powerful fashion network that inherits the advantages of both. In our fashion grammar representations, kinetic dependencies and symmetric relations are encoded. We introduce Bidirectional Convolutional Recurrent Neural Networks (BCRNNs) for modeling the message passing over our grammar topologies, leading to a fully differentiable network that can be end-to-end training. We further introduced two types of attentions for improving the performance of clothing image classification. We demonstrate our model on two benchmarks, and achieve the state-of-the-art fashion image classification and landmark detection performance against recent methods.

CHAPTER 7

Human 3D Pose Estimation using Pose Grammar

7.1 Introduction

Estimating 3D human poses from a single-view RGB image has attracted growing interest in the past few years for its wide applications in robotics, autonomous vehicles, intelligent drones etc. This is a challenging inverse task since it aims to reconstruct 3D spaces from 2D data and the inherent ambiguity is further amplified by other factors, *e.g.*, clothes, occlusions, background clutters. With the availability of large-scale pose datasets, *e.g.*, Human3.6M [IPO14], deep learning based methods have obtained encouraging success. These methods can be roughly divided into two categories: i) learning end-to-end networks that recover 2D input images to 3D poses directly, ii) extracting 2D human poses from input images and then lifting 2D poses to 3D spaces.

There are some advantages to decouple 3D human pose estimation into two stages. i) For 2D pose estimation, existing large-scale pose estimation datasets [APG14, CPM16] have provided sufficient annotations; whereas pre-trained 2D pose estimators [NYD16] are also generalized and mature enough to be deployed elsewhere. ii) For 2D to 3D reconstruction, infinite 2D-3D pose pairs can be generated by projecting each 3D pose into 2D poses under different camera views. Recent works [YIK16, MHR17] have shown that well-designed deep networks can achieve state-of-the-art performance on Human3.6M dataset using only 2D pose detections as system inputs.

However, despite their promising results, few previous methods explored the problem of encoding domain-specific knowledge into current deep learning based detectors.

In this chapter, we develop a deep grammar network to explicitly encode a set of knowl-

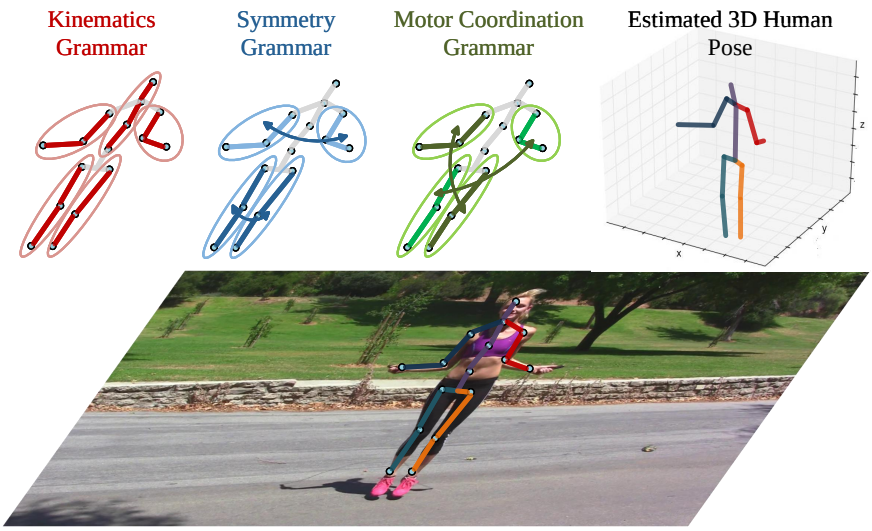


Figure 7.1: Illustration of human pose grammar, which express the knowledge of human body configuration. We consider three kinds of human body dependencies and relations in this chapter, *i.e.*, kinematics (red), symmetry (blue) and motor coordination (green).

edge over human body dependencies and relations, as illustrated in Figure 7.1. These knowledges explicitly express the composition process of joint-part-pose, including kinematics, symmetry and motor coordination, and serve as knowledge bases for reconstructing 3D poses. We ground these knowledges in a multi-level RNN network which can be end-to-end trained with back-propagation. The composed hierarchical structure describes composition, context and high-order relations among human body parts.

Additionally, we empirically find that previous methods are restricted to their poor generalization capabilities while performing cross-view pose estimation, *i.e.*, being tested on human images from unseen camera views. Notably, on the Human3.6M dataset, the largest publicly available human pose benchmark, we find that the performance of state-of-the-art methods heavily relies on the camera viewpoints. As shown in Table 1, once we change the split of training and testing set, using 3 cameras for training and testing on the forth camera (new *protocol #3*), performance of state-of-the-art methods drops dramatically and is much worse than image-based deep learning methods. These empirical studies suggested that existing methods might over-fit to sparse camera settings and bear poor generalization capabilities.

To handle the issue, we propose to augment the learning process with more camera views, which explore a generalized mapping from 2D spaces to 3D spaces. More specifically, we develop a pose simulator to augment training samples with virtual camera views, which can further improve system robustness. Our method is motivated by the previous works on learning by synthesis. Differently, we focus on the sampling of 2D pose instance from a given 3D space, following the basic geometry principles. In particular, we develop a pose simulator to effectively generate training samples from unseen camera views. These samples can greatly reduce the risk of over-fitting and thus improve generalization capabilities of the developed pose estimation system.

We conduct exhaustive experiments on public human pose benchmarks, *e.g.*, Human3.6M, HumanEva, MPII, to verify the generalization issues of existing methods, and evaluate the proposed method for cross-view human pose estimation. Results show that our method can significantly reduce pose estimation errors and outperform the alternative methods to a large extend.

Contributions. There are two major contributions of the proposed framework: i) a deep grammar network that incorporates both powerful encoding capabilities of deep neural networks and high-level dependencies and relations of human body; ii) a data augmentation technique that improves generalization ability of current 2-step methods, allowing it to catch up with or even outperforms end-to-end image-based competitors.

7.2 Related Work

The proposed method is closely related to the following two tracks in computer vision and artificial intelligence.

3D pose estimation. In literature, methods solving this task can be roughly classified into two frameworks: i) directly learning 3D pose structures from 2D images, ii) a cascaded framework of first performing 2D pose estimation and then reconstructing 3D pose from the estimated 2D joints. Specifically, for the first framework, [LC14] proposed a multi-task convolutional network that simultaneously learns pose regression and part detection. [TKS16]

first learned an auto-encoder that describes 3D pose in high dimensional space then mapped the input image to that space using CNN. [PZD17] represented 3D joints as points in a discretized 3D space and proposed a coarse-to-fine approach for iterative refinement. [ZHS17] mixed 2D and 3D data and trained an unified network with two-stage cascaded structure. These methods heavily relies on well-labeled image and 3D ground-truth pairs, since they need to learn depth information from images.

To avoid this limitation, some work [PVD03, Jia10, YIK16] tried to address this problem in a two step manner. For example, in [YIK16], the authors proposed an exemplar-based method to retrieve the nearest 3D pose in the 3D pose library using the estimated 2D pose. Recently, [MHR17] proposed a network that directly regresses 3D keypoints from 2D joint detections and achieves state-of-the-art performance. Our work takes a further step towards a unified 2D-to-3D reconstruction network that integrates the learning power of deep learning and the domain-specific knowledge represented by hierarchy grammar model. The proposed method would offer a deep insight into the rationale behind this problem.

Grammar model. This track receives long-lasting endorsement due to its interpretability and effectiveness in modeling diverse tasks [LCK14, XLL16, XLQ17]. In [HZ09], the authors approached the problem of image parsing using a stochastic grammar model. After that, grammar models have been used in [XLZ13, XMH14] for 2D human body parsing. [PNZ15] proposed a phrase structure, dependency and attribute grammar for 2D human body, representing decomposition and articulation of body parts. Notably, [NWZ17] represented human body as a set of simplified kinematic grammar and learn their relations with LSTM. In this chapter, our representation can be analogized as a hierarchical attributed grammar model, with similar hierarchical structures, BRNNS as probabilistic grammar. The difference lies in that our model is fully recursive and without semantics in middle levels.

7.3 Representation

We represent the 2D human pose $\mathbf{U}$ as a set of N_U joint locations

$$\mathbf{U} = \{u_i : i = 1, \ldots, N_U,\ u_i \in \mathbb{R}^2\}. \tag{7.1}$$

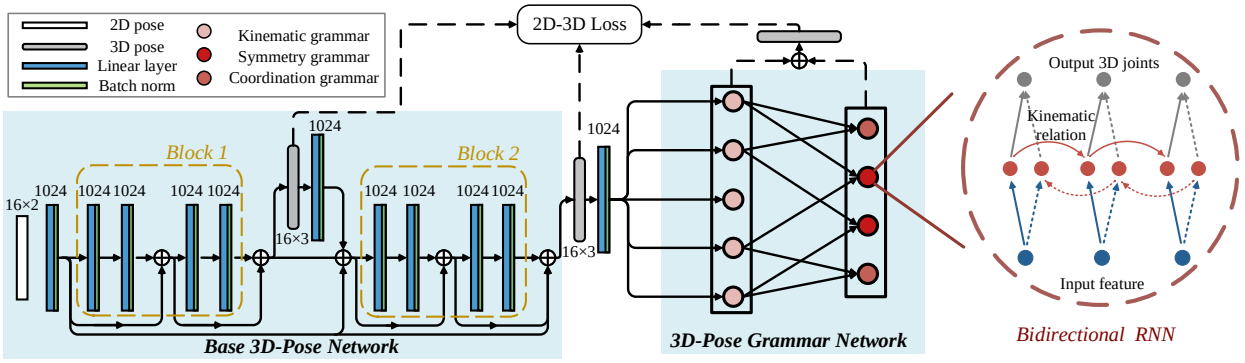


Figure 7.2: The proposed deep grammar network. Our model consists of two major components: a base network constituted by two basic blocks and a pose grammar network encoding human body dependencies and relations w.r.t. kinematics, symmetry and motor coordination. Each grammar is represented as a Bi-directional RNN among certain joints. See text for detailed explanations.

Our task is to estimate the corresponding 3D human pose $\mathbf{V}$ in the world reference frame. Suppose the 2D coordinate of a joint u_i is $[x_i, y_i]$ and the 3D coordinate v_i is $[X_i, Y_i, Z_i]$, we can describe the relation between 2D and 3D as a pinhole image projection

$$\begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix} = K\,[R|RT] \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}, K = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}, T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}, \tag{7.2}$$

where w_i is the depth w.r.t. the camera reference frame, K is the camera intrinsic parameter (*e.g.*, focal length α_x and α_y, principal point x_0 and y_0), R and T are camera extrinsic parameters of rotation and translation, respectively. Note we omit camera distortion for simplicity.

It involves two sub-problems in estimating 3D pose from 2D pose: i) calibrating camera parameters, and ii) estimating 3D human joint positions. Noticing that these two sub-problems are entangled and cannot be solved without ambiguity, we propose a deep neural network to learn the generalized 2D→3D mapping $\mathbf{V} = f(\mathbf{U}; \theta)$, where $f(\cdot)$ is a multi-to-multi mapping function, parameterized by θ.

7.3.1 Model Overview

Our model follows the line that directly estimating 3D human keypoints from 2D joint detections, which renders our model high applicability. More specifically, we extend various human pose grammar into deep neural network, where a basic 3D pose detection network is first used for extracting pose-aligned features, and a hierarchy of RNNs is built for encoding high-level 3D pose grammar for generating final reasonable 3D pose estimations. Above two networks work in a cascaded way, resulting in a strong 3D pose estimator that inherits the representation power of neural network and high-level knowledge of human body configuration.

7.3.2 Base 3D-Pose Network

For building a solid foundation for high-level grammar model, we first use a base network for capturing well both 2D and 3D pose-aligned features. The base network is inspired by [MHR17], which has been demonstrated effective in encoding the information of 2D and 3D poses. As illustrated in Figure 7.2, our base network consists of two cascaded blocks. For each block, several linear (fully connected) layers, interleaved with Batch Normalization, Dropout layers, and $ReLU$ activation, are stacked for efficiently mapping the 2D-pose features to higher-dimensions. The input 2D pose detections $\mathbf{U}$ (obtained as ground truth 2D joint locations under known camera parameters, or from other 2D pose detectors) are first projected into a 1024-d features, with a fully connected layer. Then the first block takes this high-dimensional features as input and an extra linear layer is applied at the end of it to obtain an explicit 3D pose representation. In order to have a coherent understanding of the full body in 3D space, we re-project the 3D estimation into a 1024-dimension space and further feed it into the second block. With the initial 3D pose estimation from the first block, the second block is able to reconstruct a more reasonable 3D pose. To take a full use of the information of initial 2D pose detections, we introduce *residual connections* [HZR16] between the two blocks. Such technique is able to encourage the information flow and facilitate our training. Additionally, each block in our base network is able to directly access

to the gradients from the loss function (detailed in Sec.7.4), leading to an implicit deep supervision [LXG15]. With the refined 3D-pose, estimated from base network, we again re-projected it into a 1024-d features. We combine the 1024-d features from the 3D-pose and the original 1024-d feature of 2D-pose together, which leads to a powerful representation that has well-aligned 3D-pose information and preserves the original 2D-pose information. Then we feed this feature into our 3D-pose grammar network.

7.3.3 3D-Pose Grammar Network

So far, our base network directly estimated the depth of each joint from the 2D pose detections. However, the natural of human body that rich inherent structures are involved in this task, motivates us to reason the 3D structure of the whole person in a global manner. Here we extend Bi-directional RNNs (BRNN) to model high-level knowledge of 3D human pose grammar, which towards a more reasonable and powerful 3D pose estimator that is capable of satisfying human anatomical and anthropomorphic constraints. Before going deep into our grammar network, we first detail our grammar formulations that reflect interpretable and high-level knowledge of human body configuration. Basically, given a human body, we consider the following three types of grammar in our network.

Kinematic grammar $\mathcal{G}^{kin}$ describes human body movements without considering forces (*i.e.*, the red skeleton in Figure 7.1)). We define 5 kinematic grammar to represent the constraints among kinematically connected joints:

$$\mathcal{G}^{kin}_{spine}: \ head \leftrightarrow thorax \leftrightarrow spine \leftrightarrow hip \ , \tag{7.3}$$

$$\mathcal{G}^{kin}_{l.arm}: \ l.shoulder \leftrightarrow l.elbow \leftrightarrow l.wrist \ , \tag{7.4}$$

$$\mathcal{G}^{kin}_{r.arm}: \ r.shoulder \leftrightarrow r.elbow \leftrightarrow r.wrist \ , \tag{7.5}$$

$$\mathcal{G}^{kin}_{l.leg}: \ l.hip \leftrightarrow l.knee \leftrightarrow l.foot \ , \tag{7.6}$$

$$\mathcal{G}^{kin}_{r.leg}: \ r.hip \leftrightarrow r.knee \leftrightarrow r.foot \ . \tag{7.7}$$

Kinematic grammar focuses on connected body parts and works both forward and backward. Forward kinematics takes the last joint in a kinematic chain into account while backward

kinematics reversely influences a joint in a kinematics chain from the next joint.

Symmetry grammar $\mathcal{G}^{sym}$ measure bilateral symmetry of human body (*i.e.*, blue skeleton in Figure 7.1), as human body can be divided into matching halves by drawing a line down the center; the left and right sides are mirror images of each other.

$$\mathcal{G}^{sym}_{arm} : \mathcal{G}^{kin}_{l.arm} \leftrightarrow \mathcal{G}^{kin}_{r.arm} , \tag{7.8}$$

$$\mathcal{G}^{sym}_{leg} : \mathcal{G}^{kin}_{l.leg} \leftrightarrow \mathcal{G}^{kin}_{r.leg} . \tag{7.9}$$

Motor coordination grammar $\mathcal{G}^{crd}$ represents movements of several limbs combined in a certain manner (*i.e.*, green skeleton in Figure 7.1). In this chapter, we consider simplified motor coordination between human arm and leg. We define 2 coordination grammar to represent constraints on people coordinated movements:

$$\mathcal{G}^{crd}_{l\rightarrow r} : \mathcal{G}^{kin}_{l.arm} \leftrightarrow \mathcal{G}^{kin}_{r.leg} , \tag{7.10}$$

$$\mathcal{G}^{crd}_{r\rightarrow l} : \mathcal{G}^{kin}_{r.arm} \leftrightarrow \mathcal{G}^{kin}_{l.leg} . \tag{7.11}$$

The RNN naturally supports chain-like structure, which provides a powerful tool for modeling our grammar formulations with deep learning. There are two states (forward/backward directions) encoded in BRNN. At each time step t, with the input feature a_t, the output y_t is determined by considering two-direction states h^f_t and h^b_t:

$$y_t = \phi(W^f_y h^f_t + W^b_y h^b_t + b_y), \tag{7.12}$$

where ϕ is the softmax function and the states h^f_t, h^b_t are computed as:

$$\begin{aligned} h^f_t &= \tanh(W^f_h h^f_{t-1} + W^f_a a_t + b^f_h) , \\ h^b_t &= \tanh(W^b_h h^b_{t+1} + W^b_a a_t + b^b_h) , \end{aligned} \tag{7.13}$$

As shown in Figure 7.2, we build a two-layer tree-like hierarchy of BRNNs for modeling our three grammar, where each of the BRNNs shares same equation in Equation (7.12) and the three grammar are represented by the edges between BRNNs nodes or implicitly encoded into BRNN architecture.

For the bottom layer, five BRNNs are built for modeling the five relations defined in kinematics grammar. More specifically, they accept the pose-aligned features from our base network as input, and generate estimation for a 3D joint at each time step. The information is forward/backward propagated efficiently over the two states with BRNN, thus the five Kinematics relations are implicitly modeled by the bi-directional chain structure of corresponding BRNN. Note that we take the advantages of recurrent natures of RNN for capturing our chain-like grammar, instead of using RNN for modeling the temporal dependency of sequential data.

For the top layer, totally four BRNN nodes are derived, two for symmetry relations and two for motor coordination dependencies. For the symmetry BRNN nodes, taking $\mathcal{G}^{sym}_{arm}$ node as an example, it takes the concatenated 3D-joints (totally 6 joints) from the $\mathcal{G}^{kin}_{l.arm}$ and $\mathcal{G}^{kin}_{r.arm}$ BRNNs in the bottom layer in all times as input, and produces estimations for the six 3D-joints taking their symmetry relations into account. Similarly, for the coordination nodes, such as $\mathcal{G}^{crd}_{l\rightarrow r}$, it leverages the estimations from $\mathcal{G}^{kin}_{l.arm}$ and $\mathcal{G}^{kin}_{r.leg}$ BRNNs and refines the 3D joints estimations according to coordination grammar.

In this way, we inject three kinds of human pose grammar into a tree-BRNN model and the final 3D human joints estimations are achieved by mean-pooling the results from all the nodes in the grammar hierarchy.

7.4 Learning

Given a training set Ω:

$$\Omega = \{(\hat{\mathbf{U}}^k, \hat{\mathbf{V}}^k) : \ k = 1, \ldots, N_\Omega\}, \tag{7.14}$$

where $\hat{\mathbf{U}}^k$ and $\hat{\mathbf{V}}^k$ denote ground-truth 2D and 3D pose pairs, we define the 2D-3D loss of learning the mapping function $f(\mathbf{U};\theta)$ as

$$\begin{aligned} \theta^* &= \arg\min_{\theta} \ell(\Omega|\theta) \\ &= \arg\min_{\theta} \sum_{k=1}^{N_\Omega} \|f(\hat{\mathbf{U}}^k;\theta) - \hat{\mathbf{V}}^k\|_2. \end{aligned} \tag{7.15}$$

The loss measures the Euclidean distance between predicted 3D pose and true 3D pose.

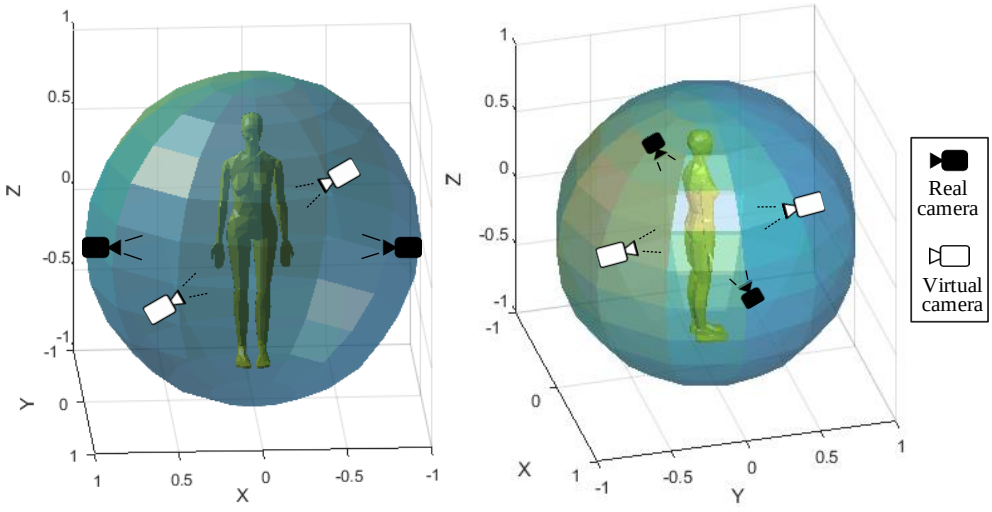


Figure 7.3: Illustration of virtual camera simulation. The black camera icons stand for real camera settings while the white camera icons simulated virtual camera settings.

The entire learning process consists of two steps: i) learning basic blocks in the base network with 2D-3D loss. ii) attaching pose grammar network on the top of the trained base network, and fine-tune the whole network in an end-to-end manner.

7.4.1 Pose Sample Simulator

We conduct an empirical study on popular 3D pose estimation datasets (*e.g.*, *Human3.6M*, *HumanEva*) and notice that there are usually limited number of cameras (4 on average) recording the human subject. This raises the doubt whether learning on such dataset can lead to a generalized 3D pose estimator applicable in other scenes with different camera positions. We believe that a data augmentation process will help improve the model performance and generalization ability. For this, we propose a novel Pose Sample Simulator (PSS) to generate additional training samples. The generation process consists of two steps: i) projecting ground-truth 3D pose $\hat{\mathbf{V}}$ onto virtual camera planes to obtain ground-truth 2D pose $\hat{\mathbf{U}}$, ii) simulating 2D pose detections $\mathbf{U}$ by sampling conditional probability distribution $p(\mathbf{U}|\hat{\mathbf{U}})$.

In the first step, we first specify a series of virtual camera calibrations. Namely, a virtual camera calibration is specified by quoting intrinsic parameters K' from other real cameras and

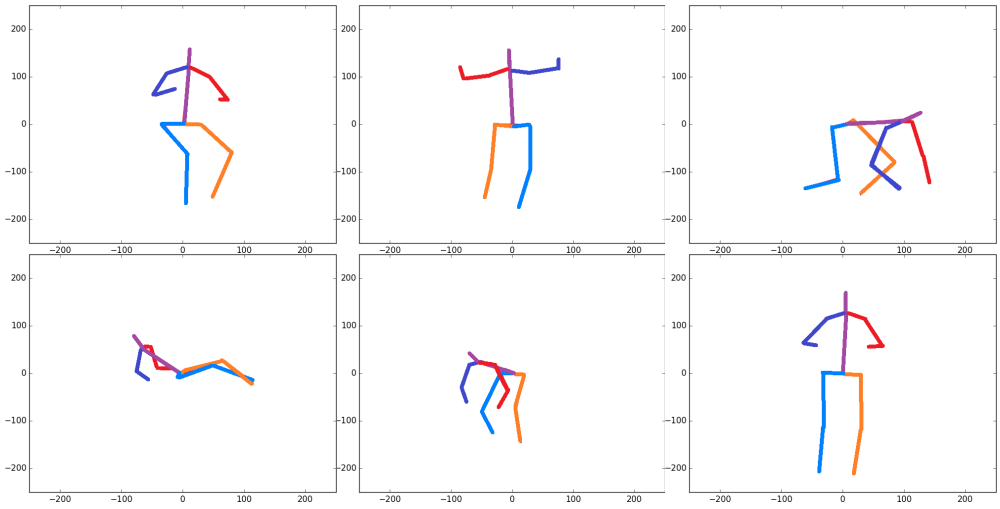


Figure 7.4: Examples of learned 2D atomic poses in probability distribution $p(\mathbf{U}|\hat{\mathbf{U}})$.

simulating reasonable extrinsic parameters (*i.e.*, camera locations T' and orientations R'). As illustrated in Figure 7.3, two white virtual camera calibrations are determined by the other two real cameras. Given a specified virtual camera, we can perform a perspective projection of a ground-truth 3D pose $\hat{\mathbf{V}}$ onto the virtual camera plane and obtain the corresponding ground-truth 2D pose $\hat{\mathbf{U}}$.

In the second step, we first model the conditional probability distribution $p(\mathbf{U}|\hat{\mathbf{U}})$ to mitigate the discrepancy between 2D pose detections $\mathbf{U}$ and 2D pose ground-truth $\hat{\mathbf{U}}$. Assuming $p(\mathbf{U}|\hat{\mathbf{U}})$ follows a mixture of Gaussian distribution, that is,

$$p(\mathbf{U}|\hat{\mathbf{U}}) = p(\epsilon) = \sum_{j=1}^{N_G} \omega_j \, \mathbb{N}(\epsilon; \mu_j, \Sigma_j), \tag{7.16}$$

where $\epsilon = \mathbf{U} - \hat{\mathbf{U}}$, N_G denotes the number of Gaussian distributions, ω_j denotes a combination weight for the j-th component, $\mathbb{N}(\epsilon; \mu_j, \Sigma_j)$ denotes the j-th multivariate Gaussian distribution with mean μ_j and covariance Σ_j. As suggested in [APG14], we set $N_G = 42$. For efficiency issues, the covariance matrix Σ_j is assumed to be in the form:

$$\Sigma_j = \begin{bmatrix} \sigma_{j,1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{j,i} \end{bmatrix}, \quad \sigma_{j,i} \in \mathbb{R}^{2\times 2} \tag{7.17}$$

where $\sigma_{j,i}$ is the covariance matrix for joint u_i at j-th multivariate Gaussian distribution. This constraint enforces independence among each joint u_i in 2D pose $\mathbf{U}$.

The probability distribution $p(\mathbf{U}|\hat{\mathbf{U}})$ can be efficiently learned using an EM algorithm, with E-step estimating combination weights ω and M-step updating Gaussian parameters μ and Σ. We utilizes K-means clustering to initialize parameters as a warm start. The learned mean μ_j of each Gaussian can be considered as an atomic pose representing a group of similar 2D poses. We visualize some atomic poses in Figure 7.4.

Given a 2D pose ground-truth $\hat{\mathbf{U}}$, we sample $p(\mathbf{U}|\hat{\mathbf{U}})$ to generate simulated detections $\mathbf{U}$ and thus use it augment the training set Ω. By doing so we mitigate the discrepancy between the training data and the testing data. The effectiveness of our proposed PSS is validated in Section 7.5.5.

7.5 Experiments

In this section, we first introduce datasets and settings for evaluation, and then report our results and comparisons with state-of-the-art methods, and finally conduct an ablation study on components in our method.

7.5.1 Datasets

We evaluate our method quantitatively and qualitatively on three popular 3D pose estimation datasets.

Human3.6M [IPO14] is the current largest dataset for human 3D pose estimation, which consists of 3.6 million 3D human poses and corresponding video frames recorded from 4 different cameras. Cameras are located at the front, back, left and right of the recorded subject, with around 5 meters away and 1.5 meter height. In this dataset, there are 11 actors in total and 15 different actions performed (*e.g.*, greeting, eating and walking). The 3D pose ground-truth is captured by a motion capture (Mocap) system and all camera parameters (intrinsic and extrinsic parameters) are provided.

Protocol #1	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
LinKDE (PAMI'16)	132.7	183.6	132.3	164.4	162.1	205.9	150.6	171.3	151.6	243.0	162.1	170.7	177.1	96.6	127.9	162.1
Tekin et al. (ICCV'16)	102.4	147.2	88.8	125.3	118.0	182.7	112.4	129.2	138.9	224.9	118.4	138.8	126.3	55.1	65.8	125.0
Du et al. (ECCV'16)	85.1	112.7	104.9	122.1	139.1	135.9	105.9	166.2	117.5	226.9	120.0	117.7	137.4	99.3	106.5	126.5
Chen & Ramanan (Arxiv'16)	89.9	97.6	89.9	107.9	107.3	139.2	93.6	136.0	133.1	240.1	106.6	106.2	87.0	114.0	90.5	114.1
Pavlakos et al. (CVPR'17)	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3	83.7	96.5	71.7	65.8	74.9	59.1	63.2	71.9
Bruce et al. (ICCV'17)	90.1	88.2	85.7	95.6	103.9	92.4	90.4	117.9	136.4	98.5	103.0	94.4	86.0	90.6	89.5	97.5
Zhou et al. (ICCV'17)	54.8	60.7	58.2	71.4	**62.0**	**65.5**	53.8	**55.6**	75.2	111.6	64.1	66.0	**51.4**	63.2	55.3	64.9
Martinez et al. (ICCV'17)	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Ours	**50.1**	**54.3**	**57.0**	**57.1**	66.6	73.3	**53.4**	55.7	**72.8**	**88.6**	**60.3**	**57.7**	62.7	**47.5**	**50.6**	**60.4**
Protocol #2	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Ramakrishna et al.(ECCV'12)	137.4	149.3	141.6	154.3	157.7	158.9	141.8	158.1	168.6	175.6	160.4	161.7	150.0	174.8	150.2	157.3
Bogo et al. (ECCV'16)	62.0	60.2	67.8	76.5	92.1	77.0	73.0	75.3	100.3	137.3	83.4	77.3	86.8	79.7	87.7	82.3
Moreno-Noguer (CVPR'17)	66.1	61.7	84.5	73.7	65.2	67.2	60.9	67.3	103.5	74.6	92.6	69.6	71.5	78.0	73.2	74.0
Pavlakos et al. (CVPR'17)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	51.9
Bruce et al. (ICCV'17)	62.8	69.2	79.6	78.8	80.8	72.5	73.9	96.1	106.9	88.0	86.9	70.7	71.9	76.5	73.2	79.5
Martinez et al. (ICCV'17)	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
Ours	**38.2**	**41.7**	**43.7**	**44.9**	**48.5**	**55.3**	**40.2**	**38.2**	**54.5**	**64.4**	**47.2**	**44.3**	**47.3**	**36.7**	**41.7**	**45.7**
Protocol #3	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SitingD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Pavlakos et al. (CVPR'17)	79.2	85.2	78.3	89.9	86.3	87.9	75.8	81.8	106.4	137.6	86.2	92.3	72.9	82.3	77.5	88.6
Bruce et al. (ICCV'17)	103.9	103.6	101.1	111.0	118.6	105.2	105.1	133.5	150.9	113.5	117.7	108.1	100.3	103.8	104.4	112.1
Zhou et al. (ICCV'17)	61.4	70.7	**62.2**	76.9	**71.0**	**81.2**	67.3	71.6	96.7	126.1	**68.1**	76.7	**63.3**	72.1	68.9	75.6
Martinez et al. (ICCV'17)	65.7	68.8	92.6	79.9	84.5	100.4	72.3	88.2	109.5	130.8	76.9	81.4	85.5	69.1	68.2	84.9
Ours	**57.5**	**57.8**	81.6	**68.8**	75.1	85.8	**61.6**	**70.4**	**95.8**	**106.9**	68.5	**70.4**	73.8	**58.5**	**59.6**	**72.8**

Table 7.1: Quantitative comparisons of Average Euclidean Distance (mm) between the estimated pose and the ground-truth on *Human3.6M* under *Protocol #1*, *Protocol #2* and *Protocol #3*. The best score is marked in **bold**.

HumanEva-I [SBB10] is another widely used dataset for human 3D pose estimation, which is also collected in a controlled indoor environment using a Mocap system. *HumanEva-I* dataset has fewer subjects and actions, compared with *Human3.6M* dataset.

MPII [APG14] is a challenging benchmark for 2D human pose estimation in the wild, containing a large amount of human images in the wild. We only validate our method on this dataset qualitatively since no 3D pose ground-truth is provided.

7.5.2 Evaluation Protocols

For **Human3.6M**, the standard protocol is using all 4 camera views in subjects S1, S5, S6, S7 and S8 for training and the same 4 camera views in subjects S9 and S11 for testing. This standard protocol is called *protocol #1*. In some works, the predictions are post-processed via a rigid transformation before comparing to the ground-truth, which is referred as *protocol #2*.

In above two protocols, the same 4 camera views are both used for training and testing. This raise the question whether or not the learned estimator over-fits to training camera parameters. To validate the generalization ability of different models, we propose a new protocol based on different camera view partitions for training and testing. In our setting, subjects S1, S5, S6, S7, and S8 in 3 camera views are used for training while subjects S9 and S11 in the other camera view are selected for testing (down-sampled to 10 FPS). The suggested protocol guarantees that not only subjects but also camera views are different for training and testing, eliminating interferences of subject appearance and camera parameters, respectively. We refer our new protocol as *protocol #3*.

For **HumanEva-I**, we follow the previous protocol, evaluating on each action separately with all subjects. A rigid transformation is performed before computing the mean reconstruction error.

7.5.3 Implementation Details

We implement our method using Keras with Tensorflow as back-end. We first train our base network for 200 epoch. The learning rate is set as 0.001 with exponential decay and the batch size is set to 64 in the first step. Then we add the 3D-Pose Grammar Network on top of the base network and fine-tune the whole network together. The learning rate is set as 10^{-5} during the second step to guarantee model stability in the training phase. We adopt Adam optimizer for both steps.

We perform 2D pose detections using a state-of-the-art 2D pose estimator [NYD16]. We fine-tuned the model on *Human3.6M* and use the pre-trained model on *HumanEva-I* and *MPII*. Our deep grammar network is trained with 2D pose detections as inputs and 3D pose ground-truth as outputs. For *protocol #1* and *protocol #2*, the data augmentation is omitted due to little improvement and tripled training time. For *protocol #3*, in addition to the original 3 camera views, we further augment the training set with 6 virtual camera views on the same horizontal plane. Consider the circle which is centered at the human subject and locates all cameras is evenly segmented into 12 sectors with 30 degree angles each, and 4 cameras occupy 4 sectors. We generate training samples on 6 out of 8 unoccupied sectors and leave 2 closest to the testing camera unused to avoid over-fitting. The 2D poses generated from virtual camera views are augmented by our PCSS. During each epoch, we will sample our learned distribution once and generate a new batch of synthesized data.

Empirically, one forward and backward pass takes 25 ms on a Titan X GPU and a forward pass takes 10 ms only, allowing us to train and test our network efficiently.

7.5.4 Results and Comparisons

Human3.6M. We evaluate our method under all three protocols. We compare our method with 10 state-of-the-art methods [IPO14, TRL16, DWL16, CR17, SNP16, RS16, BKL16, PZD17, NWZ17, ZHS17, MHR17] and report quantitative comparisons in Table 7.1. From the results, our method obtains superior performance over the competing methods under all protocols.

Figure 7.5: Quantitative results of our method on *Human3.6M* and *MPII*. We show the estimated 2D pose on the original image and the estimated 3D pose from a novel view. Results on *Human3.6M* are drawn in the first row and results on *MPII* are drawn in the second to fourth row. Best viewed in color.

To verify our claims, we re-train three previous methods, which obtain top performance under *protocol #1*, with *protocol #3*. The quantitative results are reported in Table. 7.1. The large drop of performance (17% – 41%) of previous 2D-3D reconstruction models [PZD17, NWZ17, ZHS17, MHR17], which demonstrates the blind spot of previous evaluation protocols and the over-fitting problem of those models.

Notably, our method greatly surpasses previous methods (12mm improvement over the second best under cross-view evaluation (*i.e.*, *protocol #3*). Additionally, the large performance gap of [MHR17] under *protocol #1* and *protocol #3* (62.9mm *vs* 84.9mm) demonstrates that previous 2D-to-3D reconstruction networks easily over-fit to camera views. Our

Methods	Walking			Jogging			Avg.
	S1	S2	S3	S1	S2	S3	
Simo-Serra *et al.* (CVPR'13)	65.1	48.6	73.5	74.2	46.6	32.2	56.7
Kostrikov *et al.* (BMVC'14)	44.0	30.9	41.7	57.2	35.0	33.3	40.3
Yasin *et al.* (CVPR'16)	35.8	32.4	41.6	46.6	41.4	35.4	38.9
Moreno-Noguer (CVPR'17)	19.7	**13.0**	**24.9**	39.7	20.0	21.0	26.9
Pavlakos *et al.* (CVPR'17)	22.3	19.5	29.7	28.9	21.9	23.8	24.3
Martinez *et al.* (ICCV'17)	19.7	17.4	46.8	**26.9**	18.2	18.6	24.6
Ours	**19.4**	16.8	37.4	30.4	**17.6**	**16.3**	**22.9**

Table 7.2: Quantitative comparisons of the mean reconstruction error (mm) on *HumanEva-I*. The best score is marked in **bold**.

general improvements over different settings demonstrate our superior performance and good generalization.

HumanEva-I. We compare our method with 6 state-of-the-art methods [SQT13, KG14, YIK16, Mor17, PZD17, MHR17]. The quantitative comparisons on *HumanEva-I* are reported in Table 7.2. As seen, our results outperforms previous methods across the vast majority of subjects and on average.

MPII. We visualize sampled results generated by our method on *MPII* as well as *Human3.6M* in Figure 7.5. As seen, our method is able to accurately predict 3D pose for both indoor and in-the-wild images.

7.5.5 Ablation studies

We study different components of our model on *Human 3.6M* dataset under *protocol #3*, as reported in Table 7.3.

Pose grammar. We first study the effectiveness of our grammar model, which encodes high-level grammar constraints into our network. First, we exam the performance of our baseline by removing all three grammar from our model, the error is 75.1mm. Adding the

Component	Variants	Error (mm)	Δ
	Ours, full	72.8	–
Pose grammar	w/o. grammar	75.1	2.3
	w. kinematics	73.9	1.1
	w. kinematics+symmetry	73.2	0.4
PSS	w/o. extra 2D-3D pairs	82.6	9.8
	w. extra 2D-3D pairs, GT	76.7	3.9
	w. extra 2D-3D pairs, simple	78.0	5.2
PSS Generalization	Bruce et al. (ICCV'17) w/o.	112.1	–
	Bruce et al. (ICCV'17) w.	96.3	15.8
	Martinez *et al.* (ICCV'17) w/o.	84.9	–
	Martinez *et al.* (ICCV'17) w.	76.0	8.9

Table 7.3: Ablation studies on different components in our method. The evaluation is performed on *Human3.6M* under *Protocol #3*. See text for detailed explanations.

kinematics grammar provides parent-child relations to body joints, reducing the error by 1.6% ($75.1mm \rightarrow 73.9mm$). Adding on top the symmetry grammar can obtain an extra error drops ($73.9mm \rightarrow 73.2mm$). After combing all three grammar together, we can reach an final error of $72.8mm$.

Pose Sample Simulator (PSS). Next we evaluate the influence of our 2D-pose samples simulator. Comparing the results of only using the data from original 3 camera views in *Human 3.6M* and the results of adding samples by generating ground-truth 2D-3D pairs from 6 extra camera views, we see an 7% errors drop ($82.6mm \rightarrow 76.7mm$), showing that extra training data indeed expand the generalization ability. Next, we compare our Pose Sample Simulator to a simple baseline, *i.e.*, generating samples by adding random noises to each joint, say an arbitrary Gaussian distribution or a white noise. Unsurprisingly, we observe a drop of performance, which is even worse than using the ground-truth 2D pose. This suggests that the conditional distribution $p(E|\hat{E})$ helps bridge the gap between detection results and ground-truth. Furthermore, we re-train models proposed in [NWZ17, MHR17] to validate the generalization of our PSS. Results also show a performance boost for their methods,

which confirms the proposed PSS is a generalized technique. Therefore, this ablative study validates the generalization as well as effectiveness of our PSS.

7.6 Summary

In this chapter, we propose a pose grammar model to encode the mapping function of human pose from 2D to 3D. Our method obtains superior performance over other state-of-the-art methods by explicitly encoding human body configuration with pose grammar and a generalized data argumentation technique. We will explore more interpretable and effective network architectures in the future.

CHAPTER 8

Event Understanding using Causal And-Or Graph

8.1 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 subject. 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, driver-less 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

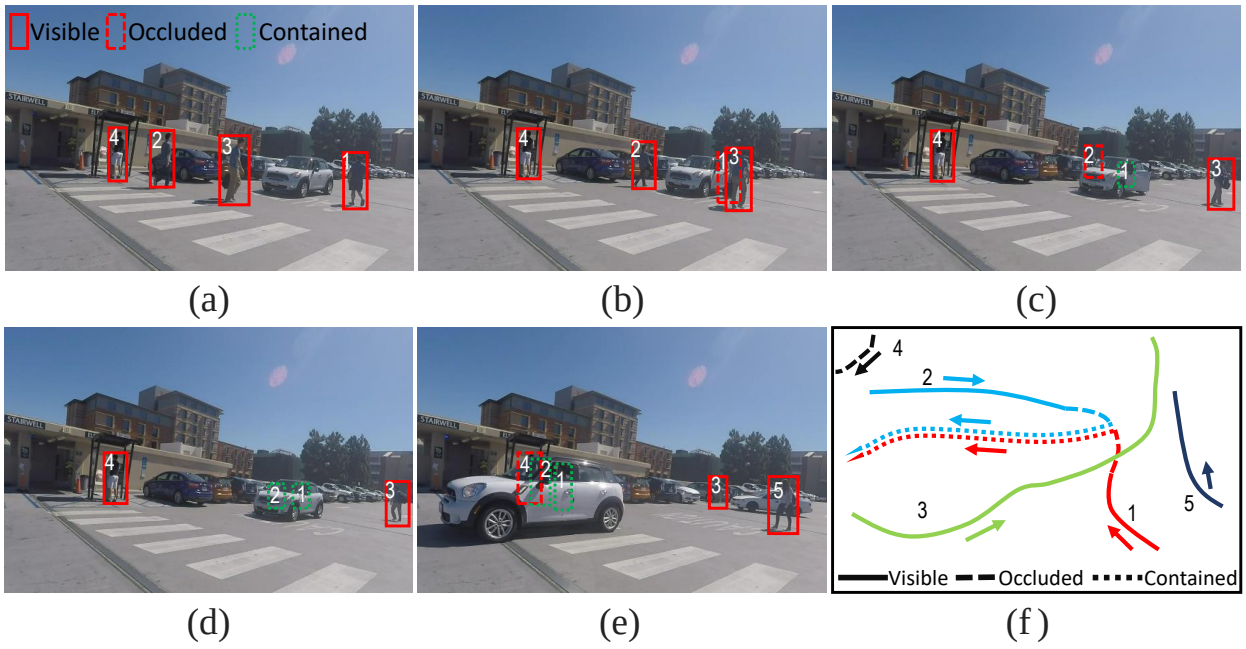


Figure 8.1: **Illustration of visibility fluent changes.** There are three states: visible, occluded, contained. When a person approaches a vehicle, its state changes from "visible" to "occluded" to "contained", such as the person_1 and person_2 (a-e). When a vehicle passes, the person_4 is occluded. 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.

denote the time varying status of an object. It is also used to represent the varying object status in commonsense reasoning [Mue14]. In this chapter, the visibility status of objects can be described as fluents varying over time. As illustrated in Fig. 8.1, 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. 8.1 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. 8.3(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.

8.2 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 [WLY14, DTT15, XLL16, XLQ17, DSY17, DSW18]. Specifically, a general detector [FGM10, RHG15] is first applied to generate detection proposals, and then data association techniques [BFT11, DAS15, YMZ16] 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 [WTF14, WTF16, MWF16] 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 [CS12, STZ17], team sports [LCC13], and people carrying luggage [BML13]. In this chapter, 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.*, images or videos. There are two popular causality models: Bayesian Network [GT05, Pea09] and grammar models [GT07, LZZ16]. Grammar models [XLZ13, FXW18, WXS18] are powerful tool for modeling high-level human knowledge in specific domains. Notably, Fire and Zhu [FZ16] 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

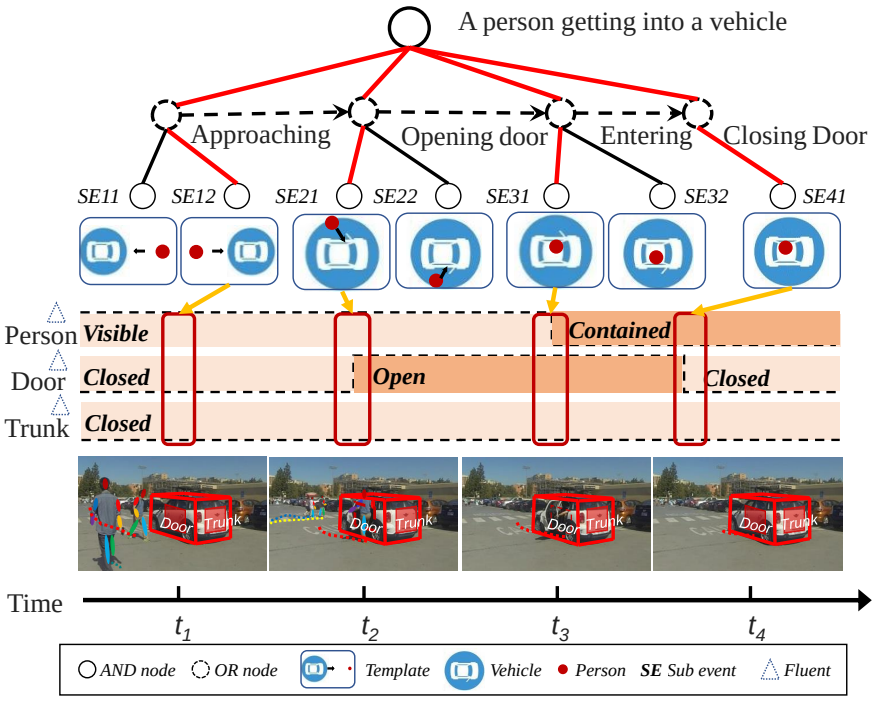


Figure 8.2: **Illustration of a person's actions and her visibility fluent changes** when she enters a vehicle.

ground the task on challenging videos in surveillance systems.

8.3 Representation

In this chapter, 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,

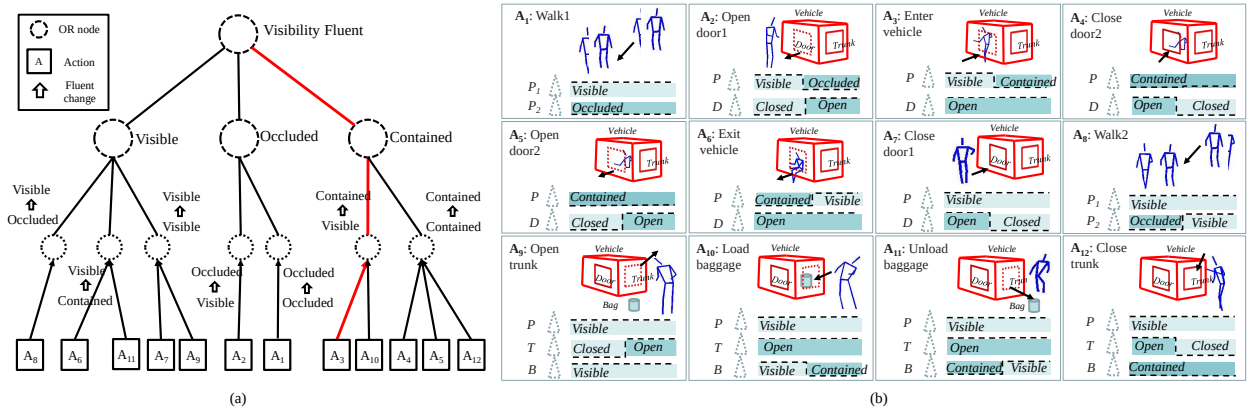


Figure 8.3: **(a) The proposed Causal And-Or Graph (C-AOG) model for the fluent of visibility.** We use a C-AOG to represent the visibility status of an subject. Each OR node indicates a possible choice and an arrow shows how visibility fluent transits among states. **(b) A series of atomic actions that 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.

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. 8.2, 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.

8.3.1 Causal And-Or Graph

In this chapter, we propose a Causal And-Or Graph (C-AOG) to represent the action-fluent relationship, as illustrated in Fig. 8.3(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

$$\begin{aligned} \mathcal{R} &= \{O_t : t = 1, 2, ..., T\}, \\ O_t &= \left\{o_t^i : i = 1, 2, ..., N_t\right\}, \end{aligned} \tag{8.1}$$

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 and

an action label a_t^i, that is,

$$o_t^i = \left(l_t^i,\ \phi_t^i,\ s_t^i, a_t^i\right). \tag{8.2}$$

Thus, the state of a subject is defined as

$$s_t^i \in S = \{\,\text{visible, occluded, contained}\,\}. \tag{8.3}$$

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. 8.3(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

$$\begin{aligned} M &= \{pg_t : t = 1, 2, \ldots, T\}, \\ pg_t &= \{o_t^i = (l_t^i, \phi_t^i, s_t^i, a_t^i) \,|\, i = 1, 2, ..., N_t\}, \end{aligned} \tag{8.4}$$

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

8.4 Problem Formulation

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

$$\begin{aligned} M^* &= \arg\max_M p(M|I;\theta) \\ &\propto \arg\max_M p(I|M;\theta) \cdot p(M;\theta) \\ &= \arg\max_M \frac{1}{Z} \exp\{-\mathcal{E}(M;\theta) - \mathcal{E}(I|M;\theta)\}. \end{aligned} \tag{8.5}$$

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

$$\begin{aligned} \mathcal{E}(M;\theta) &= \sum_{t=1}^{T-1} \mathcal{E}(pg_{t+1}|pg_t) \\ &= \sum_{t=1}^{T-1} \sum_{i=1}^{N_t} \Phi(l_{t+1}^i, l_t^i, s_t^i) + \Psi(s_{t+1}^i, s_t^i, a_t^i). \end{aligned} \tag{8.6}$$

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

$$\Phi(l_{t+1}^i, l_t^i, s_t^i) = \begin{cases} \delta(\mathcal{D}_s(l_{t+1}^i, l_t^i) > \tau_s), & s_t^i = \text{Visible}, \\ 1, & s_t^i = \text{Occ, Con}, \end{cases} \tag{8.7}$$

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:

$$\Psi(s_{t+1}^i, s_t^i, a_t^i) = -\log p(s_{t+1}^i | s_t^i, a_t^i), \tag{8.8}$$

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

$$\begin{aligned} \mathcal{E}(I|M;\theta) &= \sum_{t=1}^{T} \mathcal{E}(I_t|pg_t) \\ &= \sum_{t=1}^{T} \sum_{i=1}^{N_t} \Upsilon(l_t^i, \phi_t^i, s_t^i) + \Gamma(l_t^i, \phi_t^i, a_t^i), \end{aligned} \tag{8.9}$$

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:

$$\Upsilon(l_t^i, \phi_t^i, s_t^i) = \begin{cases} 1 - h_o(l_t^i, \phi_t^i), & s_t^i = \text{Visible}, \\ \sigma(\mathcal{D}_\varsigma(\varsigma_1^i, \varsigma_2^i)), & s_t^i = \text{Occluded}, \\ 1 - h_c(l_t^i, \phi_t^i), & s_t^i = \text{Contained}, \end{cases} \tag{8.10}$$

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,

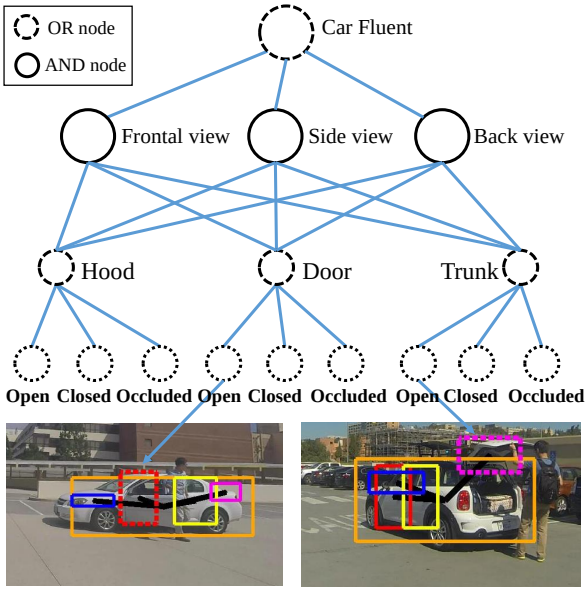


Figure 8.4: **Illustration of Hierarchical And-Or Graph.** The vehicle is decomposed into different views, semantic parts and fluents. Some detection results are drawn below, with different colored bounding boxes denoting different vehicle parts, solid/dashed boxes denoting state "closed"/"open".

there is no visual evidence to determine its state. Therefore, we utilize temporal information to generate candidate locations. We employ the SSP algorithm [PRF11] 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,

$$\Gamma(l_t^i, \phi_t^i, a_t^i) = \sigma(\mathcal{D}_h(l_t^i, \phi_t^i | a_t^i)) + \sigma(\mathcal{D}_v(l_t^i, \phi_t^i | a_t^i)), \tag{8.11}$$

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 [WRK16]. 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:

$$\mathcal{D}_h(l_t^i, \phi_t^i | a_t^i) = -\log\ N(\omega_t^h; \mu_{a_t^i}, \Sigma_{a_t^i}), \tag{8.12}$$

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. 8.4. 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

$$\mathcal{D}_v(l_t^i, \phi_t^i | a_t^i) = \| \omega_t^v - \varpi_{a_t^i} \|_2. \tag{8.13}$$

8.5 Inference

We cast the intractable optimization of Equation (8.5) 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. 8.5. Then the energy function Equation (8.5) can be re-written as:

$$\begin{aligned} f^* &= \arg\max_f \sum_{mn\in E_o} c_{mn} f_{mn}, \\ c_{mn} &= -\Phi(l_n, l_m, s_m) - \Psi(s_n, s_m, a_m) - \Upsilon(l_m, \phi_m, s_m) \\ &\quad - \Gamma(l_m, \phi_m, a_m), \\ s.t. &\quad f_{mn} \in \{0,1\},\ \sum_m f_{mn} \le 1,\ \sum_m f_{mn} = \sum_k f_{nk}, \end{aligned} \tag{8.14}$$

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

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 Equation (8.14), the energy function of container is as follows:

$$\begin{aligned} g^* &= \arg\max_g \sum_{mn\in E_c} d_{mn}\, g_{mn}, \\ d_{mn} &= h_c(l_m, \phi_m) - 1, \\ s.t. &\quad g_{mn} \in \{0,1\},\ \sum_m g_{mn} \le 1,\ \sum_m g_{mn} = \sum_k g_{nk}, \end{aligned} \tag{8.15}$$

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

$$\begin{aligned} &\sum_{mn\in E_c} g_{mn} \ge \sum_{ij\in E_o} f_{ij}, \\ &s.t.\ \ t_n = t_j,\ \|l_n - l_j\|_2 < \tau_c, \end{aligned} \tag{8.16}$$

where τ_c is the distance threshold. Finally, we combine Equation (8.14)-(8.16) to obtain objective function for our model:

$$\begin{aligned} f^*, g^* &= \max_{f,g} \sum_{mn\in E_o} c_{mn} f_{mn} + \sum_{ij\in E_c} d_{mn}\, g_{mn}, \\ s.t.\ \ & f_{mn} \in \{0,1\},\ \sum_m f_{mn} \le 1,\ \sum_m f_{mn} = \sum_k f_{nk}, \\ & g_{mn} \in \{0,1\},\ \sum_i g_{mn} \le 1,\ \sum_i g_{mn} = \sum_k g_{nk}, \\ & t_n = t_j,\ \|l_n - l_j\|_2 < \tau_c. \end{aligned} \tag{8.17}$$

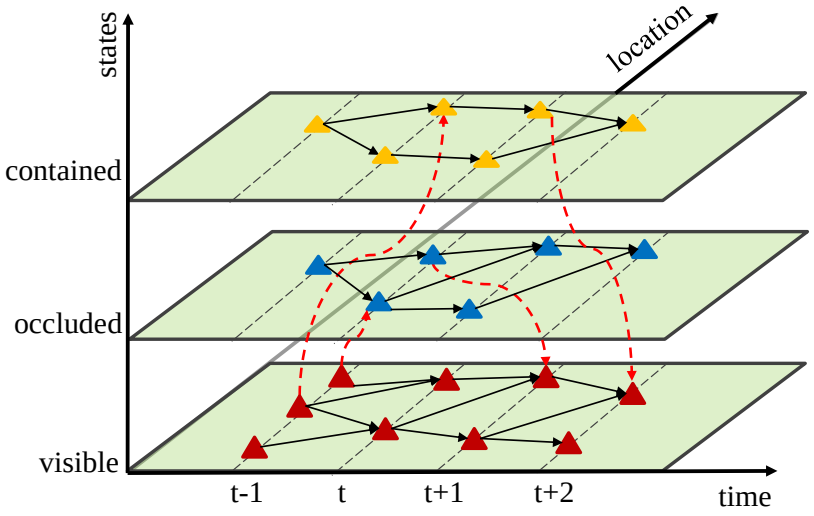


Figure 8.5: **Transition graph utilized to formulate the integer linear programming.** Each node m has its location l_m, state s_m, and time instant t_m. Black solid arrows indicate the possible transitions in the same state. Red dashed arrows indicate the possible transitions between different states.

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. 8.5.

8.6 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.

8.6.1 Implementation Details

We first utilize the Faster R-CNN models [RHG15] 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_s 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 [ZST15], which applies square root operator [AZ12] and Bag-of-word encoding to the original ColorNames descriptors. For human skeleton estimation, we use the public implementation of [WRK16]. For vehicle detection and semantic part status estimation, we use the implementation provided by [LWX16] with default parameters mentioned in their paper.

We adopt the widely used CLEAR metrics [KGS09] 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 ground-truth is above 0.5, we accept the tracking result as a correct hit.

8.6.2 Datasets

People-Car dataset [WTF14][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 [FS09], KITTI dataset [GLU12]), 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.

[1]Available at https://cvlab.epfl.ch/research/research-surv/trackinteractobj/

People-Car	Metric	Our-full	Our-1	Our-2	POM	SSP	LP2D	LP3D	KSP-fixed	KSP-free	KSP-seq	TIF-LP	TIF-MIP
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 8.1: **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**.

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 trunk/suitcase.

All video sequences are captured by a GoPro camera, with frame rate 30 FPS 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 [VPR13] to annotate the videos.

8.6.3 Results and Comparisons

For People-Car dataset, we compare our proposed method with 5 baseline methods and their variants: successive shortest path algorithm (SSP) [PRF11], K-Shortest Paths Algorithm (KSP-fixed, KSP-free, KSP-seq) [BFT11], Probability Occupancy Map (POM) [FBL08], Linear Programming (LP2D, LP3D) [LFK14], and Tracklet-Based Intertwined Flows (TIF-IP, TIF-MIP) [WTF16]. We refer the reader to [WTF16] for more details about the method variants. The quantitative results are reported in Table 8.1. 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) [PRF11], multiple hypothesis tracking with distinctive appearance model (MHT_D) [KLC15], Markov Decision Processes with Reinforcement

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 [KLC15]	34.3%	73.8%	56	661	15	18
MDP [XAS15]	32.9%	73.2%	24	656	9	7
DCEM [MSR16]	32.3%	**76.5**%	2	675	2	2
SSP [PRF11]	31.7%	72.1%	19	678	21	25
DCO [ASR12]	29.5%	76.4%	22	673	6	2
JPDA_m [HMZ15]	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 [XAS15]	30.1%	76.4%	397	2296	26	22
DCEM [MSR16]	29.4%	77.5%	383	2346	16	15
SSP [PRF11]	28.9%	75.0%	416	2337	12	14
MHT_D [KLC15]	25.6%	75.7%	720	2170	15	12
DCO [ASR12]	24.3%	78.1%	536	2367	38	10
JPDA_m [HMZ15]	12.3%	74.2%	1173	2263	28	17

Table 8.2: **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**.

Figure 8.6: **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.

Learning (MDP) [XAS15], Discrete-Continuous Energy Minimization (DCEM) [MSR16], Discrete-continuous optimization (DCO) [ASR12] and Joint Probabilistic Data Association (JPDA_m) [HMZ15]. We use the public implementations of these methods.

We report quantitative results and comparisons in Table 8.2 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.

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.

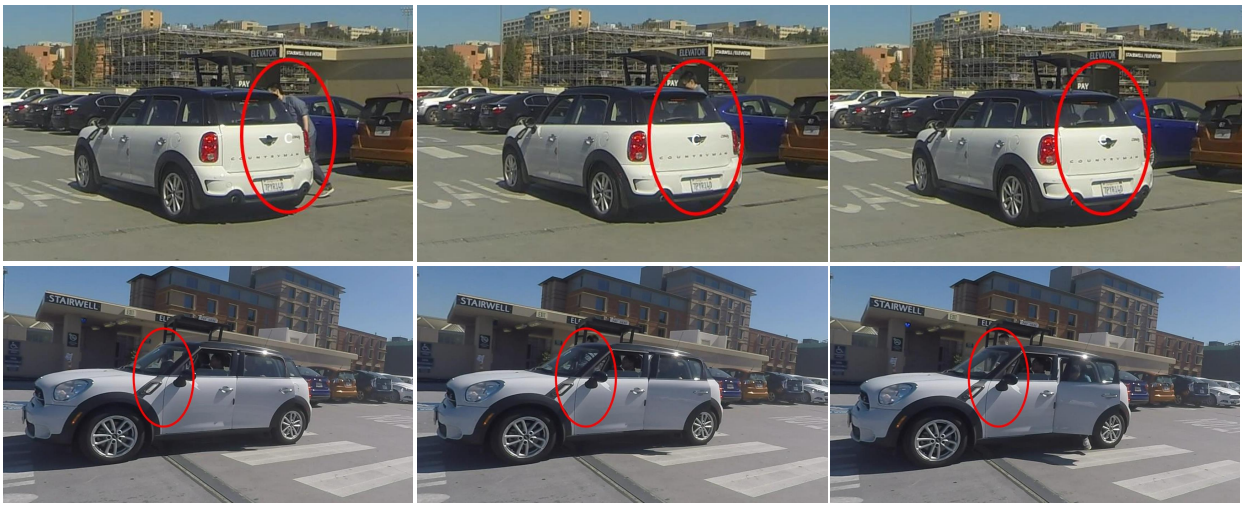


Figure 8.7: **Sampled failure cases**. When people stay behind vehicles, it is hard to determine whether or not they are interacting with the vehicle, *e.g.*, entering, exiting.

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. 8.6.

We further report fluent estimation results on TIO-Plaza sequences and TIO-ParkingLot sequences in Fig. 8.8. 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. 8.7. 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.

8.7 Summary

In this chapter, 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

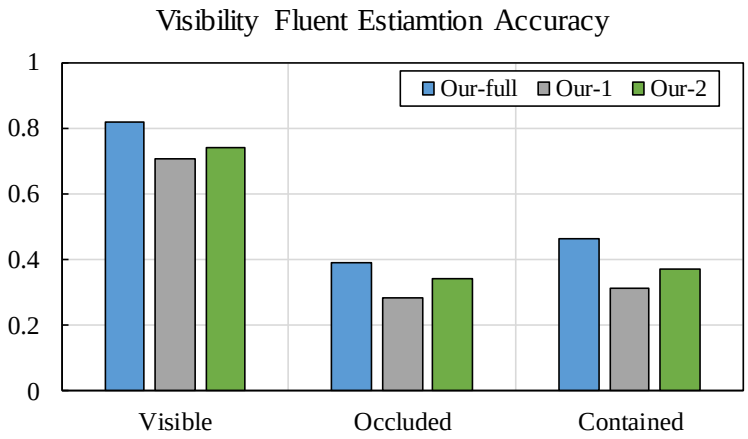


Figure 8.8: **Visibility fluent estimation** results on TIO dataset.

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.

CHAPTER 9

Conclusion

In computer vision, 3D scene and event understanding is an important yet under-explored task, as it involves many independent vision tasks (*e.g.*, object detection and tracking, human pose and attribute estimation, action and event recognition) and also heavily relies on joint inference and reasoning of those tasks. In this dissertation, we propose several novel approaches for multiple subtasks of 3D scene and event understanding and use extensive experiment results on public datasets to demonstrate the benefits of our proposed methods. In general, we solve those task by either introducing domain knowledge guided grammar models or formulating multiple tasks into a joint learning and inference framework in order to improve the performance of each other. Specifically, our contribution lies in six-fold.

First, we propose a Spatial and Temporal Attributed Parse Graph model (ST-APG) for multi-view people tracking in crowded scenes. Given videos from multiple cameras with overlapping and non-overlapping field of view (FOV), our algorithm parses all people trajectories in the scene into a scene-centric representation which explicitly encodes various fine-grained attributes of humans in both spatial and temporal domains. Our representation encodes two principles: (i) compositionality, i.e. decomposing a trajectory into sub-trajectories into boxes, using multi-model information in both 2D image and 3D scene, *e.g.*, appearance, ground occupancy, motion, which are mutually complementary while tracking people over time; (ii) attribution, i.e. augmenting each trajectory elements with a set of fine-grained semantic attributes (*e.g.*, activities), or geometric attributes (*e.g.*, facing directions, postures and actions), to enhance multi-view tracklet associations. The inference of the optimal representation is approached by iteratively grouping tracklets with cluster sampling and estimating people semantic attributes by dynamic programming. The two algorithms iterate

until convergence.

Second, we propose a joint parsing framework that integrates view-centric proposals into scene-centric parse graphs that represent a coherent scene-centric understanding of cross-view scenes. Our key observations are that overlapping fields of views embed rich appearance and geometry correlations and that knowledge fragments corresponding to individual vision tasks are governed by consistency constraints available in commonsense knowledge. The proposed joint parsing framework represents such correlations and constraints explicitly and generates semantic scene-centric parse graphs.

Third, we propose a structured neural network that combines the learning power of deep learning and the interpretable structured representation of graphical models. The proposed deep hierarchical model, *i.e.*, α-β-γ network, not only explores how hierarchical graphical structures are represented in neural network, but also focuses on how predictions are conducted. In particular, with the direct application of modern network architectures, three kinds of information flows, from image input to label output (*i.e.*, straight pass), low level to high level (*i.e.*, bottom-up process), high level to low level (*i.e.*, top-down process), are integrated and learned in end-to-end and back-propagation manner.

Fourth, we propose a knowledge-guided neural network for estimating human appearance and attributes, *e.g.*, fashion landmark localization and clothing category classification. The suggested fashion model is leveraged with high-level human knowledge in this domain. We propose two important fashion grammars: (i) *dependency grammar* capturing kinematics-like relation, and (ii) *symmetry grammar* accounting for the bilateral symmetry of clothes. We introduce Bidirectional Convolutional Recurrent Neural Networks (BCRNNs) for efficiently approaching message passing over grammar topologies, and producing regularized landmark layouts. For enhancing clothing category classification, our fashion network is encoded with two novel attention mechanisms, *i.e.*, landmark-aware attention and category-driven attention. The former enforces our network to focus on the functional parts of clothes, and learns domain-knowledge centered representations, leading to a supervised attention mechanism. The latter is goal-driven, which directly enhances task-related features and can be learned in an implicit, top-down manner. Experimental results on large-scale fashion datasets demon-

strate the superior performance of our fashion grammar network.

Fifth, we propose a pose grammar to tackle the problem of 3D human pose estimation. Our model directly takes 2D pose as input and learns a generalized 2D-3D mapping function. The proposed model consists of a base network which efficiently captures pose-aligned features and a hierarchy of Bi-directional RNNs (BRNN) on the top to explicitly incorporate a set of knowledge regarding human body configuration (*i.e.*, kinematics, symmetry, motor coordination). The proposed model thus enforces high-level constraints over human poses. In learning, we develop a pose sample simulator to augment training samples in virtual camera views, which further improves our model generalization ability.

Last, we focus on complex event understanding (*i.e.*, tracking interacting objects). 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.

9.1 Future Work

In the future, we would like to develop explainable AI systems upon the results from 3D scene and event understanding, which will effectively communicate with human users, *e.g.* analysts or users collaborating with the system, so that users gain insights and trust by understanding the inner functioning and inference trace of the system that derive its results and decisions. We will develop explanations at three levels in increasing depth.

i) Concept compositions that are represented by fragments of parse graph. The latter

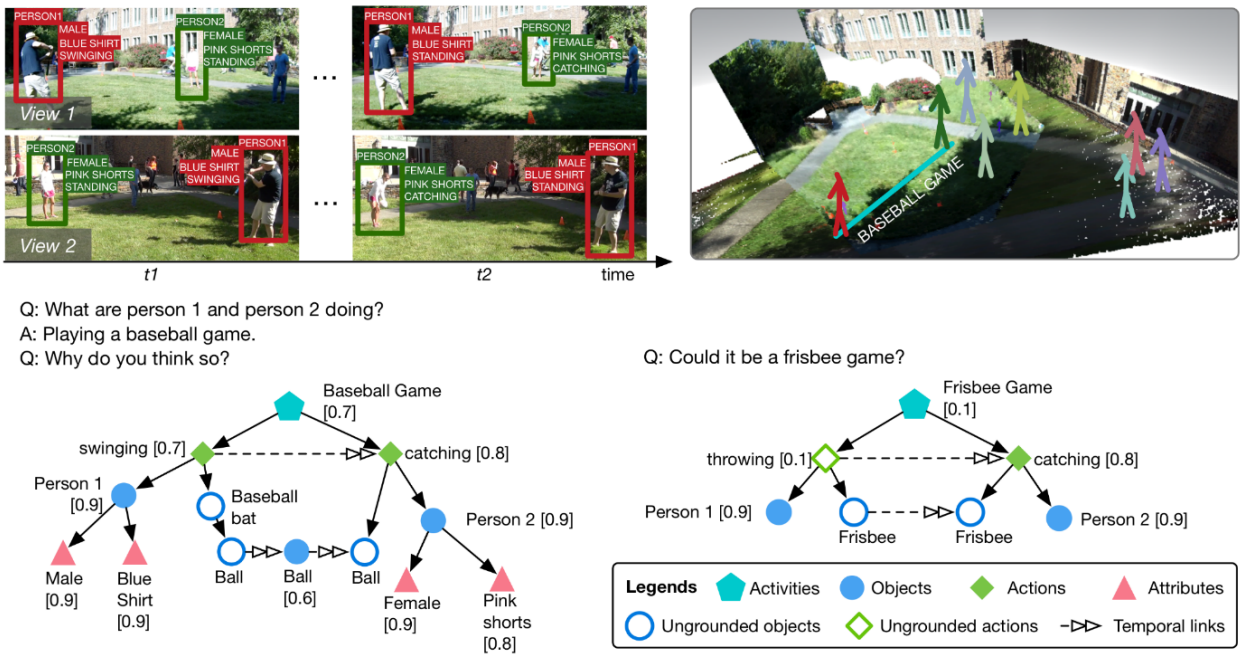


Figure 9.1: Example of complex events captured by a network of cameras in a large space and time range. The illustrative explanations are extracted from the interpretable representation – parse graphs computed by AOG model.

show how information is aggregated from its constituents and contexts, how decisions are made at various nodes under uncertainty, and confidence levels of these decisions.

ii) Causal and counter-factual reasoning which is realized by extracting causal diagrams from STC-AOG, predicts what will happen and what could have happened if certain alternative actions had been performed, and thus answers the how and what if questions.

iii) Cognition (state value, decision loss and action cost) that is the ultimate answer to why the system makes decisions in comparison with alternative actions and choices.

These explanations are mixed in dialogues with visualization and simulation in graphical interface. As Fig. 9.1 illustrates, the developed system ingests videos captured by a network of cameras (indoor, outdoor, mobile, infrared) and text input from human intelligence; reconstructs and composes 3D scenes; infers the objects, human pose, actions, attributes and group activities in the global context of the scene; and outputs spatial and temporal parse graphs with probabilities associated with nodes. There are three key components in the

proposed system:

i) Proposing models with state-of-the-art performance as well as interpretability for video analytics,

ii) Integrating different vision modules and improving the results by joint inference,

iii) Expanding the system with dialogue functions and explanation interface.

For example, when it is asked "Why do you think person 1 and person 2 are playing a baseball game?" and "Could it be a Frisbee game?", the system will output diagnostic parse graphs to support the baseball game event, in contrast to the Frisbee game. The former has a high probability for the key 'swing' action (0.9) and the latter has a low probability for the key 'swing' action (0.1).

For evaluation, we would like to work on two directions: data collection and QA system development. For data collection, we notice that most popular dataset utilize RGB cameras to record, which loses the information of 3D geometry and difficult to recover. Unlike these datasets, we plan to employ motion capture systems to capture human activities and RFID chips to record object key locations, which provides 3D ground-truth for scene and human actions. Another way could be utilizing synthetic data from 3D animations and video games. This guarantees a comprehensive recovery of the original scenario people staying at and sensor-level error, leaving human supervision outside of the loop. For example, given multi-modal sensor input, our QA system back-end first performs video analytics. Given questions raised by the agent, our system then analyzes question types and transforms into formal queries for the database storing parse graphs. Queried partial parse graphs are further returned and formatted into formal answers. The agent can make judgment about the returned answers whether the logic coincides with human cognition.

REFERENCES

[AAL15] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. "VQA: Visual Question Answering." In *IEEE International Conference on Computer Vision*, 2015.

[ABY16] S. Aditya, C. Baral, Y. Yang, Y. Aloimonos, and C. Fermuller. "DeepIU: An Architecture for Image Understanding." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[ALD11] M. Ayazoglu, B. Li, C. Dicle, M. Sznaier, and O. Camps. "Dynamic Subspace-Based Coordinated Multicamera Tracking." In *IEEE International Conference on Computer Vision*, 2011.

[APG14] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. "2D Human Pose Estimation: New Benchmark and State of the Art Analysis." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[ARS06] A. Adam, E. Rivlin, and I. Shimshoni. "Robust Fragments-based Tracking using the Integral Histogram." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[AS11] A. Andriyenko and K. Schindler. "Multi-target Tracking by Continuous Energy Minimization." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[AS12] A. Andriyenko and K. Schindler. "Discrete-continuous Optimization for Multi-Target Tracking." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[ASG17] Z. Al-Halah, R. Stiefelhagen, and K. Grauman. "Fashion Forward: Forecasting Visual Style in Fashion." In *IEEE International Conference on Computer Vision*, 2017.

[ASR12] A. Andriyenko, K. Schindler, and S. Roth. "Discrete-continuous Optimization for Multi-target Tracking." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[AT07] Y. Amit and A. Trouve. "Pop: Patchwork of Parts Models for Object Recognition." *International Journal of Computer Vision*, **75**(2):267–282, 2007.

[AZ12] R. Arandjelovic and A. Zisserman. "Three Things Everyone Should Know to Improve Object Retrieval." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[BFT11] 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.

[BKL16] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. "Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image." In *European Conference on Computer Vision*, 2016.

[BM14] O. Biran and K. McKeown. "Justification Narratives for Individual Classifications." In *IEEE International Conference on Machine Learning Workshops*, 2014.

[BML13] T. Baumgartner, D. Mitzel, and B. Leibe. "Tracking People and Their Objects." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[BZ05] A. Barbu and S. Zhu. "Generalizing Swendsen-Wang to Sampling Arbitrary Posterior Probabilities." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27(8)**:1239–1253, 2005.

[BZ07] A. Barbu and S. Zhu. "Generalizing Swendsen-Wang to Sampling Arbitrary Posterior Probabilities." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(8):1239–1253, 2007.

[CBR17] C. Corbiere, H. Ben-Younes, A. Rame, and C. Ollion. "Leveraging Weakly Annotated Data for Fashion Image Retrieval and Label Prediction." In *IEEE International Conference on Computer Vision Workshop*, 2017.

[CGG12] H. Chen, A. Gallagher, and B. Girod. "Describing clothing by semantic attributes." *European Conference on Computer Vision*, 2012.

[CHF15] Q. Chen, J. Huang, R. Feris, L. M. Brown, J. Dong, and S. Yan. "Deep Domain Adaptation for Describing People Based on Fine-grained Clothing Attributes." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[CLV06] M. G. Core, H. C. Lane, M. Van Lent, D. Gomboc, S. Solomon, and M. Rosenberg. "Building Explainable Artificial Intelligence Systems." In *AAAI Conference on Artificial Intelligence*, 2006.

[CLX16] S. Cao, W. Lu, and Q. Xu. "Deep Neural Networks for Learning Graph Representations." In *AAAI Conference on Artificial Intelligence*, 2016.

[CLY15] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, D. Ramanan, and T. S. Huang. "Look and Think Twice: Capturing Top-Down Visual Attention With Feedback Convolutional Neural Networks." In *IEEE International Conference on Computer Vision*, 2015.

[COL16] X. Chu, W. Ouyang, H. Li, and X. Wang. "CRF-CNN: Modeling Structured Information in Human Pose Estimation." In *Annual Conference on Neural Information Processing Systems*, 2016.

[CPK16] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. "Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[CPM16] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman. "Personalizing Human Video Pose Estimation." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[CR17] C.-H. Chen and D. Ramanan. "3D Human Pose Estimation = 2D Pose Estimation+Matching." In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5759–5767, 2017.

[CS12] W. Choi and S. Savarese. "A Unified Framework for Multi-Target Tracking and Collective Activity Recognition." In *European Conference on Computer Vision*, 2012.

[CSP16] H. Chen, D. Seita, X. Pan, and J. Canny. "An Efficient Minibatch Acceptance Test for Metropolis-Hastings." *arXiv preprint arXiv:1610.06848*, 2016.

[CSY15] L.-C. Chen, A. Schwing, A. Yuille, and R. Urtasun. "Learning Deep Structured Models." In *IEEE International Conference on Machine Learning*, 2015.

[CXL06] H. Chen, Z. J. Xu, Z. Q. Liu, and S.-C. Zhu. "Composite Templates for Cloth Modeling and Sketching." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[CY14] X. Chen and A. Yuille. "Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations." In *Annual Conference on Neural Information Processing Systems*, 2014.

[CYW16] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. "Attention to Scale: Scale-aware Semantic Image Segmentation." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[CZY09] Y. Chen, L. Zhu, A. Yuille, and H. Zhang. "Unsupervised Learning of Probabilistic Object Models (POMs) for Object Classification, Segmentation, and Recognition using Knowledge Propagation." **31**(10):1747–1761, 2009.

[DAS15] 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.

[DCS13] C. Dicle, O. Camps, and M. Sznaier. "The Way They Move: Tracking Multiple Targets with Similar Appearance." In *IEEE International Conference on Computer Vision*, 2013.

[DSW18] X. Dong, J. Shen, W. Wang, Y. Liu, and L. Shao. "Hyperparameter Optimization for Tracking with Continuous Deep Q-Learning." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[DSY17] 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.

[DTT15] 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.

[DWL16] Y. Du, Y. Wong, Y. Liu, F. Han, Y. Gui, Z. Wang, M. Kankanhalli, and W. Geng. "Marker-less 3D Human Motion Capture with Monocular Image Sequence and Height-maps." In *European Conference on Computer Vision*, 2016.

[EM96] P. Eilers and B. Marx. "Flexible Smoothing with B-splines and Penalties." *Statistical Science*, **11**(2):89–121, 1996.

[ESF09] A. Ellis, A. Shahrokni, and J. Ferryman. "PETS2009 and Winter-PETS 2009 Results: A Combined Evaluation." In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2009.

[FBL08] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. "Multi-camera People Tracking with a Probabilistic Occupancy Map." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(2):267–282, 2008.

[FGM10] 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.

[FS09] J. Ferryman and A. Shahrokni. "PETS2009: Dataset and Challenge." In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2009.

[FXW18] H.-S. Fang, Y. Xu, W. Wang, and S.-C. Zhu. "Learning Pose Grammar to Encode Human Body Configuration for 3D Pose Estimation." In *AAAI Conference on Artificial Intelligence*, 2018.

[FZ16] A. Fire and S.-C. Zhu. "Learning Perceptual Causality from Video." *ACM Transactions on Intelligent Systems and Technology*, **7**(2), 2016.

[FZL15] X. Fan, K. Zheng, Y. Lin, and S. Wang. "Combining Local Appearance and Holistic View: Dual-Source Deep Neural Networks for Human Pose Estimation." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[GFM11] R. B. Girshick, P. F. Felzenszwalb, and D. A. McAllester. "Object Detection with Grammar Models." In *Annual Conference on Neural Information Processing Systems*, 2011.

[GGH15] D. Geman, S. Geman, N. Hallonquist, and L. Younes. "Visual Turing Test for Computer Vision Systems." *Proceedings of the National Academy of Sciences*, **112**(12):3618–3623, 2015.

[Gir15] R. Girshick. "Fast R-CNN." In *IEEE International Conference on Computer Vision*, 2015.

[GLU12] 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.

[GLZ17] K. Gong, X. Liang, D. Zhang, X. Shen, and L. Lin. "Look Into Person: Self-Supervised Structure-Sensitive Learning and a New Benchmark for Human Parsing." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[GSR17] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. "Neural Message Passing for Quantum Chemistry." In *IEEE International Conference on Machine Learning*, 2017.

[GSS12] R. B. Grosse, C. Sci, R. Salakhutdinov, W. T. Freeman, C. Sci, and J. B. Tenenbaum. "Exploiting Compositionality to Explore a Large Space of Model Structures." In *The Conference on Uncertainty in Artificial Intelligence*, 2012.

[GT05] T. Griffiths and J. Tenenbaum. "Structure and Strength in Causal Induction." *Cognitive Psychology*, **51**(4):334–384, 2005.

[GT07] T. Griffiths and J. Tenenbaum. "Two Proposals for Causal Grammars." *Causal learning: Psychology, philosophy, and computation*, pp. 323–345, 2007.

[HAR16] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. "Generating Visual Explanations." In *European Conference on Computer Vision*, 2016.

[HFC15] J. Huang, R. S. Feris, Q. Chen, and S. Yan. "Cross-domain Image Retrieval with a Dual Attribute-aware Ranking Network." In *IEEE International Conference on Computer Vision*, 2015.

[HG17] W.-L. Hsiao and K. Grauman. "Learning the Latent "Look": Unsupervised Discovery of a Style-Coherent Embedding from Fashion Images." In *IEEE International Conference on Computer Vision*, 2017.

[HHL15] M. Hadi Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg. "Where to Buy It: Matching Street Clothing Photos in Online Shops." In *IEEE International Conference on Computer Vision*, 2015.

[HHR13] M. Hofmann, M. Haag, and G. Rigoll. "Unified Hierachical Multi-Ojbect Tracking using Global Data Assoication." In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2013.

[HLN13] C. Huang, Y. Li, and R. Nevatia. "Multiple Target tracking by learning-based hierarchical association of detection responses." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(4):898–910, 2013.

[HMZ15] 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.

[HR16] P. Hu and D. Ramanan. "Bottom-up and top-down reasoning with hierarchical rectified gaussians." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[HWH17] X. Han, Z. Wu, P. X. Huang, X. Zhang, M. Zhu, Y. Li, Y. Zhao, and L. S. Davis. "Automatic Spatially-Aware Fashion Concept Discovery." In *IEEE International Conference on Computer Vision*, 2017.

[HWJ17] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis. "Learning Fashion Compatibility with Bidirectional LSTMs." In *ACM International Conference on Multimedia*, 2017.

[HWR13] M. Hofmann, D. Wolf, and G. Rigoll. "Hypergraphs for Joint Multi-view Reconstruction and Multi-object Tracking." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[HZ09] F. Han and S. Zhu. "Bottom-up/top-down Image Parsing with Attribute Grammar." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**(1):59–73, 2009.

[HZR16] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[IPO14] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. "Human3.6m: Large Scale Datasets and Predictive Methods for 3d Human Sensing in Natural Environments." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **36**(7):1325–1339, 2014.

[JE10] S. Johnson and M. Everingham. "Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation." In *British Machine Vision Conference*, 2010.

[JFL07] H. Jiang, S. Fels, and J. Little. "A Linear Programming Approach for Multiple Object Tracking." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[JFY09] T. Joachims, T. Finley, and C.-N. J. Yu. "Cutting-plane Training of Structural SVMs." *Machine Learning*, 2009.

[Jia10] H. Jiang. "3D Human Pose Reconstruction using Millions of Exemplars." In *IEEE International Conference on Pattern Recognition*, 2010.

[JSD14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. "Caffe: Convolutional Architecture for Fast Feature Embedding Authors." In *ACM International Conference on Multimedia*, 2014.

[JSZ15] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. "Spatial Transformer Networks." In *Annual Conference on Neural Information Processing Systems*, 2015.

[JZS16] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. "Structural-RNN: Deep Learning on Spatio-temporal Graphs." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[KG14] I. Kostrikov and J. Gall. "Depth Sweep Regression Forests for Estimating 3D Human Pose from Images." In *British Machine Vision Conference*, 2014.

[KGS09] 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.

[KGS13] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang. "Sparse Representation Based Image Interpolation With Nonlocal Autoregressive Modeling." *IEEE Transactions on Image Processing*, **22**(4):1382–1394, 2013.

[KLC15] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg. "Multiple Hypothesis Tracking Revisited." In *IEEE International Conference on Computer Vision*, 2015.

[KMY06] I. Kokkinos, P. Maragos, and A. Yuille. "Bottom-up & Top-down Object Detection using Primal Sketch Features and Graphical Models." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[KN10] C.-H. Kuo and R. Nevatia. "How Does Person Identity Recognition Help Multi-person Tracking." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[KS06] S. Khan and M. Shah. "A Multiview Approach to Tracking People in Crowded Scenes using a Planar Homography Constraint." In *European Conference on Computer Vision*, 2006.

[KS16] H. S. Koppula and A. Saxena. "Anticipating Human Activities using Object Affordances for Reactive Robotic Response." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**(1):14–29, 2016.

[KYB14] M. H. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg. "Hipster Wars: Discovering Elements of Fashion Styles." In *European Conference on Computer Vision*, 2014.

[KZG17] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalanditis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. "Visual Genome: Connecting Language and Vision using Crowdsourced Dense Image Annotations." *International Journal of Computer Vision*, **123**(1):32–73, 2017.

[LC14] S. Li and A. B. Chan. "3D Human Pose Estimation from Monocular Images with Deep Convolutional Neural Network." In *Asian Conference on Computer Vision*, 2014.

[LCC12] M. Lomas, R. Chevalier, E. V. Cross II, R. C. Garrett, J. Hoare, and M. Kopack. "Explaining Robot Actions." In *ACM/IEEE International Conference on Human-Robot Interaction*, 2012.

[LCC13] 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.

[LCK14] T. Liu, S. Chaudhuri, V. Kim, Q. Huang, N. Mitra, and T. Funkhouser. "Creating Consistent Scene Graphs using a Probabilistic Grammar." *ACM Transactions on Graphics*, **33**(6):1–12, 2014.

[LCV05] H. C. Lane, M. G. Core, M. Van Lent, S. Solomon, and D. Gomboc. "Explainable Artificial Intelligence for Training and Tutoring." Technical report, Defense Technical Information Center, 2005.

[LFK14] 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.

[LHN09] Y. Li, C. Huang, and R. Nevatia. "Learning to Associate: HybridBoosted Multi-Target Tracker for Crowded Scene." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[LKZ17] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. Feris. "Fully-adaptive Feature Sharing in Multi-Task Networks with Applications in Person Attribute Classification." *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[LLA16] J. Liu, Y. Li, P. Allen, and P. N. Belhumeur. "Articulated Pose Estimation using Hierarchical Exemplar-Based Models." In *AAAI Conference on Artificial Intelligence*, 2016.

[LLJ13] X. Liu, L. Lin, and H. Jin. "Contextualized Trajectory Parsing via Spatio-temporal Graph." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(12):3010–3024, 2013.

[LLQ16] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. "DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[LLS17] X. Liang, L. Lin, X. Shen, J. Feng, S. Yan, and E. P. Xing. "Interpretable Structure-Evolving LSTM." In *IEEE International Conference on Computer Vision*, 2017.

[LLY11] X. Liu, L. Lin, S. Yan, and H. Jin. "Adaptive Tracking via Learning Hybrid Template Online." *IEEE Transactions on Circuits and Systems for Video Technology*, **21**(11):1588–1599, 2011.

[LPR12] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn. "Branch-and-price Global Optimization for Multi-view Multi-object Tracking." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[LS04] H. Liu and P. Singh. "ConceptNet– A Practical Commonsense Reasoning Toolkit." *BT technology journal*, **22**(4):211–226, 2004.

[LS10] J. Liebelt and C. Schmid. "Multi-view Object Class Detection with a 3D Geometric Model." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[LSD15] J. Long, E. Shelhamer, and T. Darrell. "Fully Convolutional Networks for Semantic Segmentation." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[LSH16] G. Lin, C. Shen, A. van den Hengel, and I. Reid. "Efficient Piecewise Training of Deep Structured Models for Semantic Segmentation." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[LSL12] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan. "Street-to-shop: Cross-scenario Clothing Retrieval via Parts Alignment and Auxiliary Set." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[LWX16] 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.

[LXG15] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. "Deeply-supervised Nets." In *Artificial Intelligence and Statistics*, 2015.

[LXS15] X. Liang, C. Xu, X. Shen, J. Yang, S. Liu, J. Tang, L. Lin, and S. Yan. "Human Parsing with Contextualized Convolutional Neural Network." In *IEEE International Conference on Computer Vision*, 2015.

[LYL16] Z. Liu, S. Yan, P. Luo, X. Wang, and X. Tang. "Fashion Landmark Detection in the Wild." In *European Conference on Computer Vision*, 2016.

[LZZ14] X. Liu, Y. Zhao, and S.-C. Zhu. "Single-View 3D Scene Parsing by Attributed Grammar." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[LZZ16] 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.

[MHR17] J. Martinez, R. Hossain, J. Romero, and J. J. Little. "A Simple yet Effective Baseline for 3D Human Pose Estimation." In *IEEE International Conference on Computer Vision*, 2017.

[MJ99] L. Medsker and L. C. Jain. *Recurrent Neural Networks: Design and Applications*. CRC press, 1999.

[ML11] X. Mei and H. Ling. "Robust Visual Tracking and Vehicle Classification via Sparse Representation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**(11):2259–2272, 2011.

[Mor17] F. Moreno-Noguer. "3D Human Pose Estimation from a Single Image via Distance Matrix Regression." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[MRR53] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. "Equation of State Calculations by Fast Computing Machines." *Journal of Chemical Physics*, **21(6)**:1087–1092, 1953.

[MSR16] 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.

[Mue14] E. T. Mueller. *Commonsense Reasoning: An Event Calculus Based Approach*. Morgan Kaufmann, 2014.

[MWF16] 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.

[MYS15] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. "Feedforward Semantic Segmentation with Zoom-out Features." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[NHH15] H. Noh, S. Hong, and B. Han. "Learning Deconvolution Network for Semantic Segmentation." In *IEEE International Conference on Computer Vision*, 2015.

[NWZ17] B. X. Nie, P. Wei, and S.-C. Zhu. "Monocular 3D Human Pose Estimation by Predicting Depth on Joints." In *IEEE International Conference on Computer Vision*, 2017.

[NXZ15] B. X. Nie, C. Xiong, and S. Zhu. "Joint Action Recognition and Pose Estimation From Video." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[NYD16] A. Newell, K. Yang, and J. Deng. "Stacked Hourglass Networks for Human Pose Estimation." In *European Conference on Computer Vision*, 2016.

[PBH13] L. Pero, J. Bowdish, E. Hartley, B. Kermgard, and K. Barnard. "Understanding bayesian rooms using composite 3d object models." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[PCZ15] T. Pfister, J. Charles, and A. Zisserman. "Flowing Convnets for Human Pose Estimation in Videos." In *IEEE International Conference on Computer Vision*, 2015.

[Pea09] J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2009.

[PMR14] H. Possegger, T. Mauthner, P. Roth, and H. Bischof. "Occlusion Geodesics for Online Multi-Object Tracking." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[PNZ15] S. Park, X. Nie, and S.-C. Zhu. "Attributed And-Or Grammar for Joint Parsing of Human Pose, Parts and Attributes." In *IEEE International Conference on Computer Vision*, 2015.

[PNZ18] S. Park, X. Nie, and S.-C. Zhu. "Attribute And-Or Grammar for Joint Parsing of Human Pose, Parts and Attributes." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40**(7):1555–1569, 2018.

[PRF11] H. Pirsiavash, D. Ramanan, and C. Fowlkes. "Globally-optimal Greedy Algorithms for Tracking a Variable Number of Objects." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[PVD03] G. S. Paul, P. Viola, and T. Darrell. "Fast Pose Estimation with Parameter-sensitive Hashing." In *IEEE International Conference on Computer Vision*, 2003.

[PZ11] J. Porway and S. Zhu. "C4 : Computing Multiple Solutions in Graphical Models by Cluster Sampling." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**(9):1713–1727, 2011.

[PZD17] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. "Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[QHW17] S. Qi, S. Huang, P. Wei, and S.-C. Zhu. "Predicting Human Activities using Stochastic Grammar." In *IEEE International Conference on Computer Vision*, 2017.

[QWL15] H. Qi, T. Wu, M.-W. Lee, and S.-C. Zhu. "A Restricted Visual Turing Test for Deep Scene and Event Understanding." *arXiv preprint arXiv:1512.01715*, 2015.

[QZH18] S. Qi, Y. Zhu, S. Huang, C. Jiang, and S.-C. Zhu. "Human-centric Indoor Scene Synthesis using Stochastic Grammar." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[RHG15] S. Ren, K. He, R. Girshick, and J. Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." In *Annual Conference on Neural Information Processing Systems*, 2015.

[RMH14] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. "Pose Nachines: Articulated Pose Estimation via Inference Machines." In *European Conference on Computer Vision*, 2014.

[RS16] G. Rogez and C. Schmid. "MoCap-guided Data Augmentation for 3D Pose Estimation in the Wild." In *Annual Conference on Neural Information Processing Systems*, 2016.

[RSG16] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why Should I Trust You?: Explaining the Predictions of Any Classifier." In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.

[SBB10] L. Sigal, A. O. Balan, and M. J. Black. "HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion." *International Journal of Computer Vision*, **87**(1):4–27, 2010.

[SBF13] H. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. "Multi-Commodity Network Flow for Tracking Multiple People." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **36**(8):1614–1627, 2013.

[SCW15] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting." In *Annual Conference on Neural Information Processing Systems*, 2015.

[SFM15] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun. "Neuroaesthetics in Fashion: Modeling the Perception of Beauty." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[SK04] H. Schneiderman and T. Kanade. "Object Detection using the Statistics of Parts." *International Journal of Computer Vision*, **56**(3):151–177, 2004.

[SLM11] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. "Parsing Natural Scenes and Natural Language with Recursive Neural Networks." In *IEEE International Conference on Machine Learning*, 2011.

[SLX13] X. Shi, H. Ling, J. Xing, and W. Hu. "Multi-target Tracking by Rank-1 Tensor Approximation." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[SNP16] M. Sanzari, V. Ntouskos, and F. Pirri. "Bayesian Image based 3D Pose Estimation." In *European Conference on Computer Vision*, 2016.

[SP97] M. Schuster and K. K. Paliwal. "Bidirectional Recurrent Neural Networks." *IEEE Transactions on Signal Processing*, **45**(11):2673–2681, 1997.

[SQT13] E. Simo-Serra, A. Quattoni, C. Torras, and F. Moreno-Noguer. "A Joint Model for 2D and 3D Pose Estimation from a Single Image." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[SSH16] K. J. Shih, S. Singh, and D. Hoiem. "Where to Look: Focus Regions for Visual Question Answering." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[STZ17] 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.

[SWH11] Z. Song, M. Wang, X.-S. Hua, and S. Yan. "Predicting occupation via human clothing and contexts." In *IEEE International Conference on Computer Vision*, 2011.

[SWJ13] X. Song, T. Wu, Y. Jia, and S.-C. Zhu. "Discriminatively Trained And-Or Tree Models for Object Detection." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[SXR15] T. Shu, D. Xie, B. Rothrock, S. Todorovic, and S. Zhu. "Joint Inference of Groups, Events and Human Roles in Aerial Videos." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[SZ15] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-scale Image Recognition." In *International Conference on Learning Representations*, 2015.

[SZS03] J. Sun, N. Zheng, and H. Shum. "Stereo Matching using Belief Propagation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(7):787–800, 2003.

[TBB09] M. Tenorth, J. Bandouch, and M. Beetz. "The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition." In *IEEE International Conference on Computer Vision Workshop*, 2009.

[TJL14] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation." In *Annual Conference on Neural Information Processing Systems*, 2014.

[TKS16] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. "Structured Prediction of 3D Human Pose with Deep Neural Networks." In *British Machine Vision Conference*, 2016.

[TML14] K. Tu, M. Meng, M. W. Lee, T. E. Choe, and S.-C. Zhu. "Joint Video and Text Parsing for Understanding Events and Answering Queries." *IEEE MultiMedia*, **21**(2):42–70, 2014.

[TRL16] B. Tekin, A. Rozantsev, V. Lepetit, and P. Fua. "Direct Prediction of 3D Body Poses from Motion Compensated Sequences." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[TSM15] K. S. Tai, R. Socher, and C. D. Manning. "Improved Semantic Representations from Tree-structured Long Short-term Memory Networks." In *Annual Meeting of the Association for Computational Linguistics*, 2015.

[UB11] A. Utasi and C. Benedek. "A 3-D Marked Point Process Model for Multi-view People Detection." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[VFM04] M. Van Lent, W. Fisher, and M. Mancuso. "An Explainable Artificial Intelligence System for Small-unit Tactical Behavior." In *National Conference on Artificial Intelligence*, 2004.

[VPR13] C. Vondrick, D. Patterson, and D. Ramanan. "Efficiently Scaling Up Crowdsourced Video Annotation." *International Journal of Computer Vision*, **32**(9):184–204, 2013.

[WBK11] Z. Wu, M. Betke, and T. Kunz. "Efficient Track Linking Methods for Track Graphs using Network-flow and Set-cover Techniques." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[WF11] E. Weng and L. Fu. "On-line Human Action Recognition by Combining Joint Tracking and Key Pose Recognition." In *IEEE/RSJ Conference on Intelligent Robots and Systems*, 2011.

[WHH09] Z. Wu, N. Hristov, T. Hedrick, T. Kunz, and M. Betke. "Tracking a Large Number of Objects from Multiple Views." In *IEEE International Conference on Computer Vision*, 2009.

[WJQ17] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. "Residual Attention Network for Image Classification." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[WL13] F. Wang and Y. Li. "Beyond Physical Connections: Tree Models in Human Pose Estimation." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[WLY14] 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.

[WNX14] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu. "Cross-view Action Modeling, Learning and Recognition." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[WRK16] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. "Convolutional Pose Machines." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[WS18] W. Wang and J. Shen. "Deep Visual Attention Prediction." *IEEE Transactions on Image Processing*, **27**(5):2368–2378, 2018.

[WTF14] X. Wang, E. Turetken, F. Fleuret, and P. Fua. "Tracking Interacting Objects Optimally using Integer Programming." In *European Conference on Computer Vision*, 2014.

[WTF16] X. Wang, E. Turetken, F. Fleuret, and P. Fua. "Tracking Interacting Objects using Intertwined Flows." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**(11):2312–2326, 2016.

[WWC14] B. Wang, G. Wang., K. L. Chan, and L. Wang. "Tracklet Association with Online Target-Specific Metric Learning." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[WXS18] W. Wang, Y. Xu, J. Shen, and S.-C. Zhu. "Attentive Fashion Grammar Network for Fashion Landmark Detection and Clothing Category Classification." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[WZ11a] X. Wang and T. Zhang. "Clothes Search in Consumer Photos via Color Matching and Attribute Learning." In *ACM International Conference on Multimedia*, 2011.

[WZ11b] T. Wu and S.-C. Zhu. "A Numerical Study of the Bottom-up and Top-down Inference Processes in And-Or Graphs." *International Journal of Computer Vision*, **93**(2):226–252, 2011.

[WZZ17] P. Wei, Y. Zhao, N. Zheng, and S.-C. Zhu. "Modeling 4D Human-object Interactions for Joint Event Segmentation, Recognition, and Object Localization." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**(6):1165–1179, 2017.

[XAS15] 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.

[XBK15] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." In *IEEE International Conference on Machine Learning*, 2015.

[XLL16] Y. Xu, X. Liu, Y. Liu, and S. Zhu. "Multi-view People Tracking via Hierarchical Trajectory Composition." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[XLQ17] Y. Xu, X. Liu, L. Qin, and S.-C. Zhu. "Cross-view People Tracking by Scene-centered Spatio-temporal Parsing." In *AAAI Conference on Artificial Intelligence*, 2017.

[XLZ13] Y. Xu, L. Lin, W.-S. Zheng, and X. Liu. "Human Re-identification by Matching Compositional Template with Cluster Sampling." In *IEEE International Conference on Computer Vision*, 2013.

[XMH14] Y. Xu, B. Ma, R. Huang, and L. Lin. "Person Search in a Scene by Jointly Modeling People Commonness and Person Uniqueness." In *ACM International Conference on Multimedia*, 2014.

[XXY15] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang. "The Application of Two-level Attention Models in Deep Convolutional Neural Network for Fine-grained Image Classification." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[XZW12] Y. Xu, H. Zhou, Q. Wang, and L. Lin. "Realtime Object-of-Interest Tracking by Learning Composite Patch-Based Templates." In *IEEE International Conference on Image Processing*, 2012.

[YGF11] A. Yao, J. Gall, G. Fanelli, and L. Gool. "Does Human Action Recognition Benefit from Pose Estimation?" In *British Machine Vision Conference*, 2011.

[YHB13] K. Yamaguchi, M. Hadi Kiapour, and T. L. Berg. "Paper Doll Parsing: Retrieving Similar Styles to Parse Clothing Items." In *IEEE International Conference on Computer Vision*, 2013.

[YHG16] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. "Stacked Attention Networks for Image Question Answering." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[YIK16] H. Yasin, U. Iqbal, B. Kruger, A. Weber, and J. Gall. "A Dual-source Approach for 3D Pose Estimation from a Single Image." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[YKO12] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. "Parsing Clothing in Fashion Photographs." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[YLL14] W. Yang, P. Luo, and L. Lin. "Clothing Co-parsing by Joint Image Segmentation and Labeling." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[YLL17] S. Yan, Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. "Unconstrained Fashion Landmark Detection via Hierarchical Recurrent Transformer Networks." In *AAAI Conference on Artificial Intelligence*, 2017.

[YMC07] Q. Yu, G. Medioni, and I. Cohen. "Multiple Target Tracking using Spatio-Temporal Markov Chain Monte Carlo Data Association." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[YMZ16] 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.

[YOL16] W. Yang, W. Ouyang, H. Li, and X. Wang. "End-to-end Learning of Deformable Mixture of Parts and Deep Convolutional Neural Networks for Human Pose Estimation." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[YYL10] B. Z. Yao, X. Yang, L. Lin, M. W. Lee, and S.-C. Zhu. "I2t: Image Parsing to Text Description." *Proceedings of the IEEE*, **98**(8):1485–1508, 2010.

[ZDS12] A. Zamir, A. Dehghan, and M. Shah. "GMCP-Tracker: Global Multi-object Tracking using Generalized Minimum Clique Graphs." In *European Conference on Computer Vision*, 2012.

[ZGB16] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. "Visual7W: Grounded Question Answering in Images." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[ZHS17] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei. "Towards 3D Human Pose Estimation in the Wild: a Weakly-supervised Approach." In *IEEE International Conference on Computer Vision*, 2017.

[ZJR15] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. "Conditional Random Fields as Recurrent Neural Networks." In *IEEE International Conference on Computer Vision*, 2015.

[ZLA15] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem. "Robust Visual Tracking via Consistent Low-rank Sparse Learning." *International Journal of Computer Vision*, **111**(2):171–190, 2015.

[ZLN08] L. Zhang, Y. Li, and R. Nevatia. "Global Data Association for Multiobject Tracking using Network Flows." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[ZSG15] X. Zhu, P. Sobihani, and H. Guo. "Long Short-term Memory over Recursive Structures." In *IEEE International Conference on Machine Learning*, 2015.

[ZST15] 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.

[ZYS15] S. Zhang, X. Yu, Y. Sui, S. Zhao, and L. Zhang. "Object Tracking with Multi-view Support Vector Machines." *IEEE Transactions on Multimedia*, **17**(3):265–278, 2015.

[ZZ11] Y. Zhao and S. Zhu. "Image Parsing via Stochastic Scene Grammar." In *Annual Conference on Neural Information Processing Systems*, 2011.