UNIVERSITY OF CALIFORNIA

Los Angeles

A Joint Parsing System for Visual Scene Understanding

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

by

HANG QI

2018

ABSTRACT OF THE DISSERTATION

A Joint Parsing System for Visual Scene Understanding

by

HANG QI
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2018
Professor Song-Chun Zhu, Chair

The computer vision community has been long focusing on classic tasks such as object detection, human attributes classification, action recognition. While the state-of-the-art performance is getting improved every year for a wide range of tasks, it remains a challenge to organize individual pieces into an integral system that parses visual scenes and events jointly. In this dissertation, we explore the problem of joint visual scene parsing in a restricted visual Turing test scenario that encourages explicit concept grounding. The goal is to build a scalable computer vision system that leverages the advancement of individual modules in various tasks and exploits the inherent correlation and constraints between them for a comprehensive understanding of visual scenes.

This dissertation contains three main parts.

Firstly, we describe a restricted visual Turing test scenario that evaluates computer vision systems across various tasks with a domain ontology and explicitly tests the grounding of concepts with formal queries. We present a benchmark for evaluating long-range recognition and event reasoning in videos captured from a network of cameras. The data and queries distinguish us from visual question answering in images and video captioning in that we emphasize explicit groundings of concepts in a restricted ontology via formal language queries.

Secondly, we propose a scalable system which leverages off-the-shelf computer vision modules to parse cross-view videos jointly. The system defines a unified knowledge representation for information sharing and is extendable to new tasks and domains. To fuse information from multiple modules and camera views, we proposed a joint parsing method 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 overlapped 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 method captures such correlations and constraints explicitly and generates semantic scene-centric parse graphs. Quantitative experiments show that scene-centric predictions outperform view-centric proposals.

Thirdly, we discuss a principled method to construct parse graph knowledge bases that retains rich structures and grounding details. By casting questions into graph fragments, we present a graph-matching based question-answering system that retrieves answers for questions via graph pattern matching.

The dissertation of HANG QI is approved.

Ying Nian Wu

Wei Wang

Demetri Terzopoulos

Song-Chun Zhu, Committee Chair

University of California, Los Angeles

2018

To my wife,

who supports me unconditionally,

and my parents,

who gave me every opportunity.

TABLE OF CONTENTS

1 Introduction **1**

1.1 Motivation 1

1.2 Our Approach 3

1.3 Outline 5

1.4 Related Work 5

1.4.1 Visual Scene Understanding 5

1.4.2 Knowledge Representation 7

2 Restricted Visual Turing Test For Visual Scene Understanding **9**

2.1 Introduction 9

2.2 Dataset 14

2.3 Queries 16

2.3.1 Formal Language Queries 18

2.3.2 Queries and Groundings Collection 19

2.4 Summary 22

3 Joint Parsing System **23**

3.1 Introduction 23

3.2 Parsing Pipeline 23

3.3 Knowledge Base and Query Answering 25

3.4 Online Computation 26

3.5 Implementation Details 27

3.6 Evaluation 28

3.7 Summary 30

4 Scene-centric Joint Parsing of Cross-view Videos . . . 31

4.1 Introduction . . . 31

4.2 Representation . . . 33

4.3 Probabilistic Formulation . . . 35

4.3.1 Cross-view Compatibility . . . 37

4.4 Inference . . . 38

4.4.1 Inferring Parse Graph Hierarchy . . . 39

4.4.2 Inferring Parse Graph Variables . . . 40

4.5 Experiments . . . 40

4.5.1 Setup and Datasets . . . 40

4.5.2 Evaluation . . . 41

4.5.3 Runtime . . . 45

4.6 Summary . . . 46

5 Parse Graph Knowledge Base and Question Answering . . . 47

5.1 Introduction . . . 47

5.2 Preliminaries . . . 48

5.2.1 Ontology and Parse Graphs . . . 48

5.2.2 RDF Triples . . . 49

5.2.3 Property Graph Model . . . 50

5.3 Parse Graph Knowledge Base . . . 50

5.3.1 Graph Structure . . . 50

5.3.2 Property List . . . 54

5.3.3 Scene-centric and View-centric Hierarchy . . . 55

5.4 Question Answering by Graph Matching . . . 56

5.4.1 Formal Language Questions . . . 57
5.4.2 Natural Language Questions . . . 59
5.4.3 Implementation Details . . . 59
5.5 Reasoning Potentials . . . 63
5.6 Summary . . . 66
6 Conclusion . . . 67
References . . . 71

LIST OF FIGURES

1.1 A comprehensive scene and event understanding demands a wide range of perception and reasoning capabilities. 2

2.1 Illustration of depth and complexity of the proposed benchmark. 10

2.2 The ontology used in our QA benchmark 11

2.3 Overview of the restricted visual Turing test. 12

2.4 An example of composing queries using our query collection toolkit 20

2.5 Distribution of predicates in each category. 21

3.1 Illustration of our prototype vision system. 24

3.2 The architecture of query engine. 26

3.3 Experiment results breakdown. 29

4.1 An example of the spatio-temporal semantic parse graph hierarchy. 32

4.2 An illustration of the proposed ontology graph describing objects, parts, actions and attributes. 34

4.3 The proposed spatio-temporal parse graph hierarchy. 34

4.4 Confusion matrices of action recognition on view-centric proposals and scene-centric predictions. 43

4.5 The breakdown of action recognition accuracy according to the number of camera views in which each entity is observed. 45

4.6 Success and failure examples of view-centric and scene-centric predictions. . . . 46

5.1 Graph matching in the parse graph knowledge base. 48

5.2 Two equivalent ways to model binary relationships. 51

5.3 Primary structures in property graph knowledge bases. 53

5.4 Examples of class taxonomy hierarchy and part hierarchy. 54

5.5 Example properties associated with different types of nodes. 55

5.6 Scene-centric and view-centric representations. 56

5.7 Graph fragments correspond to formal language queries. 58

5.8 Example dependency trees and query graph fragment for natural language questions. 60

5.9 Example parse graph fragment of a baseball game that has two participants. . . 65

LIST OF TABLES

2.1 Summary of our dataset 15

2.2 Summary of the selected subset of data. 16

2.3 Results from detection and tracking. 17

3.1 Performance by data collection. 29

4.1 Quantitative comparisons of multi-object tracking on CAMPUS and TUM Kitchen datasets. 42

4.2 Quantitative comparisons of human action recognition on CAMPUS and TUM Kitchen datasets. 43

4.3 Quantitative comparisons of human attribute recognition on CAMPUS and TUM Kitchen datasets. 44

5.1 Correspondence between ontology concepts and first-order logic syntax. 52

ACKNOWLEDGMENTS

First, I would like to express my deepest gratitude and highest respect to my advisor, Prof. Song-Chun Zhu, for his guidance and supervision during my study at UCLA. He is a true scholar who devotes all of his career to the marvelous field of computer vision, statistics, and artificial intelligence. He truly cares for his student and is very passionate about his research. From him, I learned how to do research with a high standard including defining problems, developing elegant formulations, and keeping in mind of practical applications.

I would also like to thank my doctorate committee members Prof. Yingnian Wu, Prof. Demetri Terzopoulos, and Prof. Wei Wang for their guidance throughout my study and constructive suggestions for improving this dissertation.

I feel very fortunate to have worked at the VCLA lab (Center for Vision, Cognition, Learning, and Autonomy). It is a great pleasure to work with the group members and external collaborators. In particular, I would like to express my gratitude to Tianfu Wu, Tao Yuan, Yuanlu Xu, Arjun Akula, Mun Wai Lee, Alexander Grushin, who collaborated with me in various projects related to this dissertation. I would also like to thank many of my friends in the VCLA lab for their support and insights: Jungseock Joo, Weixin Li, Seyoung Park, Dan Xie, Brandon Rothrock, Yibiao Zhao, Yang Lu, Xiaohan Nie, Chengcheng Yu, Xiaobai Liu, Bo Li, Yang Liu, Feng Shi, Wenguan Wang, Siyuan Qi, Tianmin Shu, Yixin Zhu, Nishant Shukla, Ping Wei.

Last but not the least, I would like to thank my parents for their unwavering support, and my beloved wife for sharing every moment with me.

Portions of this work were supported by DARPA MSEE project FA 8650-11-1-7149, DARPA SIMPLEX N66001-15-C-4035, ONR MURI project N00014-16-1-2007, DARPA XAI Award N66001-17-2-4029, and NSF IIS 1423305.

VITA

2012 B.S. (Software Engineering), Tongji University, Shanghai, China.

2013 M.S. (Computer Science), UCLA, Los Angeles, CA.

2012–2018 Graduate Student Researcher, Center for Vision, Cognition, Learning, and Autonomy, UCLA.

PUBLICATIONS

Hang Qi, Matthew Brown, David G. Lowe. “Low-shot Learning with Imprinted Weights”, in *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2018.

Hang Qi*, Yuanlu Xu*, Tao Yuan*, Tianfu Wu, and Song-Chun Zhu. “Scene-centric Joint Parsing of Cross-view Videos.” In *AAAI Conference on Artificial Intelligence* (AAAI), 2018.

Hang Qi, Evan R. Sparks, and Ameet Talwalkar. “Paleo: A Performance Model for Deep Neural Networks”. In *International Conference on Learning Representations* (ICLR), 2017.

Weixin Li, Jungseock Joo, Hang Qi, and Song-Chun Zhu. “Joint Image-Text News Topic Detection and Tracking by Multimodal Topic And-Or Graph.” In *IEEE Transactions on Multimedia* (TMM), vol. 19, no. 2, pp. 367-381, 2017

Hang Qi*, Tianfu Wu*, Mun Wai Lee, and Song-Chun Zhu. “A Restricted Visual Turing Test for Deep Scene and Event Understanding”. *arXiv preprint, arXiv:1512.01715.* 2015. (* Equal contributors).

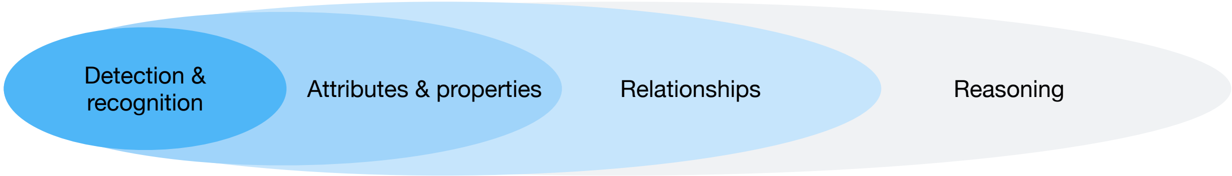
CHAPTER 1

Introduction

1.1 Motivation

Many computer vision tasks, such as object detection [FFP06, EVW], human attributes classification [ZPR14], action recognition [SZS12] segmentation [AGB07], have gained great performance improvements over the past decades. However, being able to solve the individual tasks solely does not necessarily lead to solving a complex problem as a whole. Many real-world applications, e.g. scene and events understanding, large-scale surveillance, and autonomous driving, demand a complex set of perception and reasoning capabilities on different data modalities that is not covered by any individual well-defined computer vision task in isolation. Moreover, many of these tasks are inherently correlated. Exploring the implicit connections and constraints can provide us additional information for pursuing a solution jointly. In the scope of this work, we are interested in visual scene and event understanding.

A comprehensive scene and event understanding demands a wide range of perception tasks and reasoning capabilities. As shown in Figure 1.1, detection and recognition tasks focus on locating entities and assigning labels to the data; attribute inference tasks characterize more nuanced aspects for the detected entities; relationship inference tasks explore connections between entities from various aspects (spatial, social, functional, etc); whereas reasoning tasks focus on making predictions or drawing additional conclusions from the observed information based on external knowledge. These tasks across the spectrum are fundamental to higher level applications such robotics, human-machine collaboration. In this dissertation, we primarily focus on detecting entities and objects, describing objects with a rich set of attributes, understanding the actions and activities in visual scenes.



A person	Female	On a bike. Together with a friend.	Her feet are above the ground.
Dropping (action)	A couple of seconds	He dropped a glass.	The glass is likely to be broken.
A room	About 80 sq. feet	There is a refrigerator in the room.	It's likely to be a kitchen.

Figure 1.1: A comprehensive scene and event understanding demands a wide range of perception and reasoning capabilities.

Learning a single model for a number of multimodal tasks jointly requires monolithic datasets containing annotations with multiple types of labels, such as bounding box ground-truth for localization, semantic labels ground-truth for classification, natural language sentences for image captioning and question answering. Such approach does not scale well when new tasks are introduced, since this often demands (i) a larger dataset that contains labels for new tasks in addition to existing tasks, (ii) a more complex model with more parameters to be trained on the new dataset. The size of the dataset can grow exponentially for an unbiased distribution of labels. However, the bias in existing multimodal datasets is salient and many monolithic approaches are data hungry and prone to dataset bias [ZWY17, GKS17]. Moreover, the lack of explicit grounding of concepts [JJM16] render it hard to reason the strengths and weaknesses of such models.

Alternatively, it has been our focus to pursuit a scalable approach. We consider a desired system shall have the following properties.

- **Modular.** In contrast to "one-model-solves-them-all" approaches, we pursuit a system that is able to leverage the out-of-the-box modules. With a modular framework, the system can be reconfigured with various different modules to achieve tasks that cover a range of application requirements. For example, in delay-sensitive scenarios, such as real-time surveillance and robotic applications, modules with slightly degraded performance but lower computation workload may be preferred over computation intensive alternatives.

- **Decoupled.** The system shall be able to be expanded and improved without requiring all existing modules to be retrained. Being able to decouple tasks and for the system to take in pre-trained models allows contributing components to be developed independently.
- **Extendable.** The system shall be able to be extended to additional tasks or domains while having the high-level interface unchanged such as knowledge base management, retrieval, and question answering.
- **Explicit.** We view the ability of representing system's internal state explicitly an very important characteristics. When the system complexity increases, an implicit representation allows users to identify strengths and weaknesses of the system and therefore to establish trusts and reliance.

1.2 Our Approach

Our approach can be summarized as three main parts.

Restricted Visual Turing Test. We propose an evaluation framework called restricted Turing test that evaluates a computer vision system's capabilities from various perspectives *explicitly* via formal queries. In contrast to viewing question answering as a multimodal and language prediction problem [FHS10, AAL15], we treat question answering as an interface for systematically exploring the capacity of an intelligent system. Specifically, we propose to use a restricted domain-specific ontology to covering important spatial, temporal, and causal aspects in videos with the quality of queries and answers controlled. The proposed benchmark gives informed performance measure across the task spectrum regarding a system's strengths and weaknesses. Our contributions to scene and event understanding include:

(i) a new scene and event understanding benchmark consisting of a long-term and multi-camera captured video dataset;

(ii) a set of formal and storyline-based queries that evaluates the capability of computer vision systems with explicit grounding of concepts.

Joint Parsing System. Rather than building a monolithic model, we propose a joint parsing system that leverages the advancement of various computer vision tasks while, more importantly, exploits interconnections between them for a comprehensive understanding of visual scenes and events. Concretely, our system uncovers the semantic structure of scenes in a cross-view camera network by integrating pre-trained modules . The cross-view setting implies rich physical and geometry constraints due to the overlap between fields of views. Our joint parsing system optimizes for a scene-centric parse graph that summarizes all spatial and temporal concepts from multiple view-centric local understanding of the scene obtained from originally isolated components. The contributions of our method 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.

Unified Knowledge Representation. For sharing information among multiple modules, we explore a extendable unified knowledge representation. We propose a principled approach originated from first-order logic to build knowledge bases in the form of labeled property graphs from parse graphs and answer queries with this knowledge base.

Our joint approach for visual scene understanding is designed with desired characteristics in mind to be scalable. In particular, our modular and decoupled architecture leverages off-the-shelf computer vision models. The underlying ontology can be easily extended to other domains and applications. The parse graph knowledge base is an explicit representation of the system's interpretation of input data.

1.3 Outline

We present our approach in the rest of the dissertation as follows.

In Chapter 2, we propose a restricted visual Turing test with a dataset and benchmark that embodies an extendable query-answering framework for systematically experimenting and evaluating computer vision systems across the task spectrum explicitly with a domain ontology.

In Chapter 3, we describe a modular software system architecture for leveraging the advancements from individual computer vision tasks. The system provides an unified knowledge representation for information sharing. We present evaluation results on our prototype system. In Chapter 4, we formulate cross-view joint parsing as an MAP problem that infers scene-centric parse graphs from view-centric proposals by exploring appearance and geometry constraints embedded in multi-camera cross-view videos. We compare our method with multiple baselines on publicly available datasets.

Finally, in Chapter 5, we discuss principles and implementation details for building parse graph knowledge bases and performing question answering with graph pattern matching. In addition, the close correspondence to first-order logic gives reasoning potentials.

1.4 Related Work

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

1.4.1 Visual Scene Understanding

Visual Turing Test. Inspired by the generic Turing test principle in AI [Tur50], Geman et al. proposed a visual Turing test [GGH15] for object detection tasks in images which organizes queries into storylines, within which queries are connected and the complexities are increased gradually – similar to conversations between human beings. In a similar spirit, Malinowski and Fritz [MF14a, MF14b] proposed a multi-word method to address factual queries of scene images. In the dataset and evaluation framework proposed in this work, we adopt similar evaluation structure to [GGH15], but focus on a more complex scenario which features videos and overlapping cameras to facilitate a broader scope of vision tasks.

Image Description and Visual Question Answering. To go beyond labels and bounding boxes, image tagging [DBL11], image captioning [FHS10, KPD11, MXY15], and video captioning [RQT13] have been proposed recently. The state-of-the-art methods have shown, however, a coarse level understanding of an image (i.e., labels and bounding boxes of appeared objects) together with natural language *n*-gram statistics suffices to generate reasonable captions. Microsoft COCO [LMB14] provides descriptions or captions for images. Question answering focuses on specific contents on the image and evaluate the system's abilities using human generated question. Unlike the image description task where a generated sentence is consider correct as long as it describes the dominant objects and activities in the image, human generated questions can ask all details and even hidden knowledge that require deduction. In such scenario, a pre-trained end-to-end system may not necessarily perform well as the question space is too large to be covered by training data. IQA [RKZ15] converts image descriptions into QA pairs. VQA [AAL15] evaluates in a free-formed and open-ended questions about images, where the question-answer pairs are given by human annotators. Although it encourages participants to pursuit a deep and specific understanding about the image, it only focuses on the content of the image and does not address many other fundamental aspects of computer vision like 3D scene parsing, camera registration, etc. Moreover, actions are not static concepts, temporal information are largely missing in images. Visual Genome [KZG17] dataset collects fine-grained grounding annotations of objects, attributes, actions, and relations which targets similar goal as ours but on image data. Existing video description datasets [RRW13, RRT15] and activity videos datasets [PR12, CSS09, RA11, SXR15] addresses high-level actions and activities in videos but do not incorporate multiple simultaneous videos from different views.

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.

Multi-modal Embedding. Many tasks and applications rely on bridge the semantic of visual and linguistic signals. One stream of research, such as image captioning [FHS10] and visual question answering [AAL15], focuses on predicting a sequence of words as a translated version of the input or answers to natural language inputs. Learning a hidden multi-modal embedding, as a common approach, requires training datasets and prones to dataset bias [ZWY17, JJM16].Alternatively, a modular approach [ARD16a, ARD16b] defines a set of neural network modules dedicated to different sub-tasks and solves the overall problem by composing multiple modules. These task-specific modules are low-level operations such as localization, composition, regression. In our work, however, we assume the modules are computer vision models pre-trained for high-level tasks such as detection, tracking, action recognition, and human attributes classification. A centralized parse graph knowledge base serve as the semantic bridge between vision and language. Our approach utilizes parse graph knowledge base to decouple the vision and language tasks. Similarly, [HLJ09] uses a RDF/OWL ontology to encode entities and relationships in the video and utilizes SPARQL translated from natural language to retrieve video segments. Rather than using triples, in our work, we adopt an property graph model to formulate the semantic knowledge for visual scenes, which allows a richer representation for internal structures due to the property list associated at every node and edge.

1.4.2 Knowledge Representation

Semantic representations. Semantic and expressive representations have been developed for various vision tasks, e.g., image parsing [HZ09], 3D scene reconstruction [LZZ14, PBH13], human-object interaction [KS16], pose and attribute estimation [WZZ16]. In this work, our parse graph representation also falls into this category. The difference is that our parse graph hierarchy is defined upon cross-view spatio-temporal domain and is able to incorporate a variety of tasks.

Graph-based representation. Graph structures has been widely used in database and knowledge base [Sow76, BEP08, SH13, CM08]. We adopt similar ideas but aims to build an parse graph knowledge base in the domain of visual understanding. A number of work in computer vision have utilized graph structures to represent objects and relations exists in single images. For exmaple, scene graph [JKS15] represents objects, attributes, and their relationships in the scenes. It collects first-order tuples directly from crowdsourcing human workers, whereas we extract the relationships directly from a large number of image captions. Visual Genome [KZG17] collects and constructs graphs for individual images from annotated bounding boxes and natural languages. In contrast, our parse graph knowledge graph captures spatio-temporal information in cross-view videos.

Visual ontology. In our scope of visual scene understanding, the ontology is predefined as a restricted domain [QWL15]. Although automatic ontology discovery is not the goal our work, our system can be generalized to ontologies that are automatically discovered. For example, NEIL [CSG13] discovers object-object, object-attribute, scene-attribute, and scene-object relationships by mining the web. Visual sentiment ontology [BJC13] builds a ontology of sentiment concepts in the form of adjective-noun phrases with links to 24 selected emotions based on a psychological model. [ZFF14] constructs a knowledge base by mining images from action dataset and text from the web with a set of manually designed labels and classes. [VC10] starts with a manually created concept list which then gets refined by users.

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.

CHAPTER 2

Restricted Visual Turing Test For Visual Scene Understanding

2.1 Introduction

During the past decades, we have seen tremendous progress in individual vision modules such as image classification [FP05, GD05, LSP06, ZWZ15] and object detection [FGM10, SWJ13, ZM06, Gir15, RHG15], especially after competitions like PASCAL VOC [EEV14] and ImageNet ILSVRC [RDS15] and the convolutional neural networks [LBD89, KSH12, HZR15] trained on the ImageNet dataset [DDS09] were proposed. Those tasks are evaluated based on either classification or detection accuracy, focusing on a coarse level understanding of data. In the area of natural language and text processing, there have been well-studied text-based question answering (QA). For example, a chatterbot named Eugene Goostman[1] was reported as the first computer program which has passed the famed Turing test [Tur50] in an event organized at the University of Reading. The success of text-based QA and the recent achievements of individual vision modules have inspired visual Turing tests (VTT) [GGH15, MF14b] where image-based questions (so-called visual question answering, VQA) or storyline-based queries are used to test a computer vision system. VTT has been suggested as a more suitable evaluation framework in going beyond measuring the accuracy of labels and bounding boxes. Most existing work on VTT focus on images and emphasize free-form and open-ended QA's [BJJ10, AAL15].

In this work, we are interested in a restricted visual Turing test setting with storyline-

[1] https://en.wikipedia.org/wiki/Eugene_Goostman

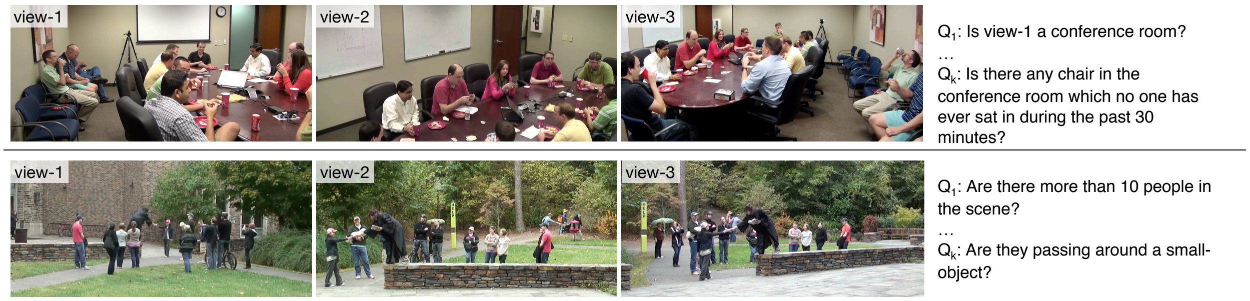

view-1
view-2
view-3
Q_1: Is view-1 a conference room?
…
Q_k: Is there any chair in the conference room which no one has ever sat in during the past 30 minutes?
view-1
view-2
view-3
Q_1: Are there more than 10 people in the scene?
…
Q_k: Are they passing around a small-object?

Figure 2.1: Illustration of depth and complexity of the proposed benchmark in scene and event understanding, which focuses on a largely unexplored task in computer vision – joint spatial, temporal, and causal understanding of scene and event in multi-camera videos over relatively long time durations. See text for details.

based visual query answering in long-term videos. Our scene and event understanding benchmark emphasizes a joint spatial, temporal, and causal understanding of scenes and events, which are largely unexplored in computer vision. By "restricted", we mean the queries are designed based on a selected ontology. Figure 2.1 shows two examples in our dataset. Consider the question how we shall test whether a computer vision system understands, for example, a conference room. In our benchmark, to understand a conference room, the input consists of multi-camera captured videos and storyline-based queries covering basic questions (e.g., Q_1, for a coarse level understanding) and difficult ones (e.g., Q_k) involving spatial, temporal, and causal inference for a deeper understanding. More specifically, to answer Q_k correctly, a computer vision system would need to build a scene-centered representation for the conference room, to detect, track, re-identify, and parse people coming into the room across cameras, and to understand the concept of sitting in a chair (i.e., the pose of a person and scene-centered spatial relation between a person and a chair), etc. Our motivation is in two folds as follows.

Web-scale images vs. long-term videos. Web-scale images emphasize the breadth that a computer vision system can learn and handle in different applications. These images are often of album photo styles collected from different image search engines such as Flickr, Google, Bing, and Facebook. This work focuses on long-term, especially multi-camera cap-

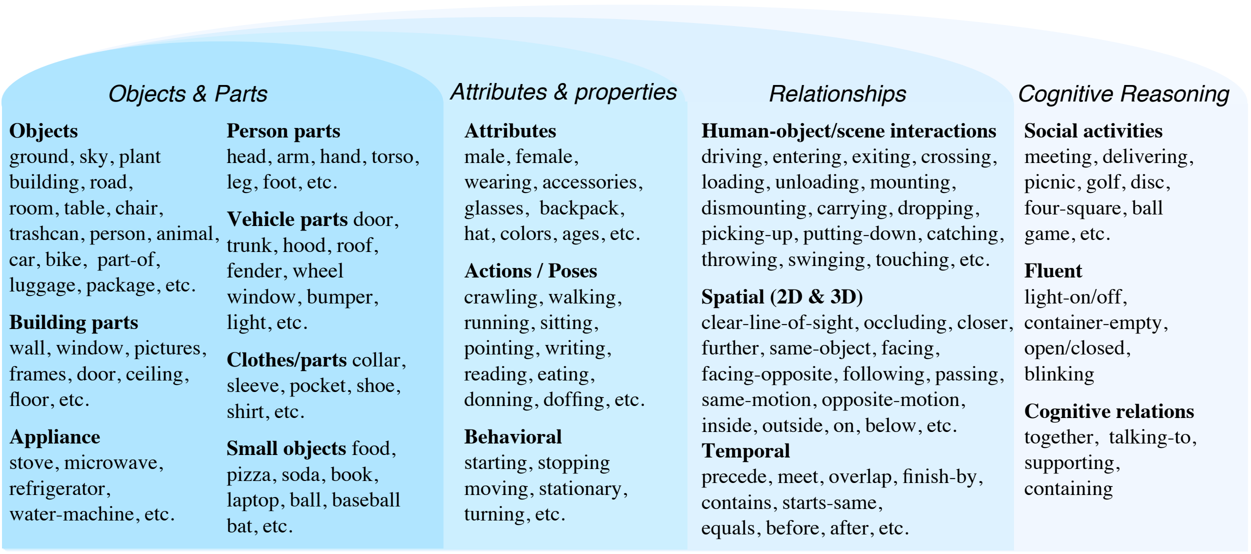


Figure 2.2: The ontology used in our QA benchmark

tured, videos usually produced by video surveillance, which are also important data sources in the visual big data epic and have important security or law enforcement applications. Furthermore, as the examples in Figure 2.1 show, mutli-camera videos can facilitate a much deeper understanding of scenes and events. The two types of datasets are complementary, but the latter has not been explored in a QA setting.

Free-form and open-ended questions vs. restricted storyline-based queries. In VQA [AAL15], the input is an image and a "bag-of-questions" (e.g., is this a conference room?) and the task is to provide a natural language answer (either in a multiple-choice manner or with free-form responses). Free-form and open-ended questions are usually collected through crowd-sourcing platforms like Amazon Mechanical Turk (MTurk) to achieve diversity. However, it is hard to obtain *well-posed* pairs from a massive amount of untrained workers on the Internet. This is challenging even for simple tasks like image labeling as investigated in the ImageNet dataset [DDS09] and the Label-Me dataset [KZM12]. For the queries in this work, we adopt a selected yet sufficiently expressive ontology (shown in Figure 2.2) in generating queries. Following the statistical principles stated in Geman el al's Turing test framework [GGH15], we design a easy-to-use toolkit by which several people with certain expertise can create a large number of storylines covering different interesting and important spatial, temporal, and causal aspects in videos with the quality of queries and answers controlled. We are working on a more sophisticated toolkit and inspection methods to exploit MTurk to scale up collecting storyline-based queries covering long-term temporal ranges and across multi-cameras.

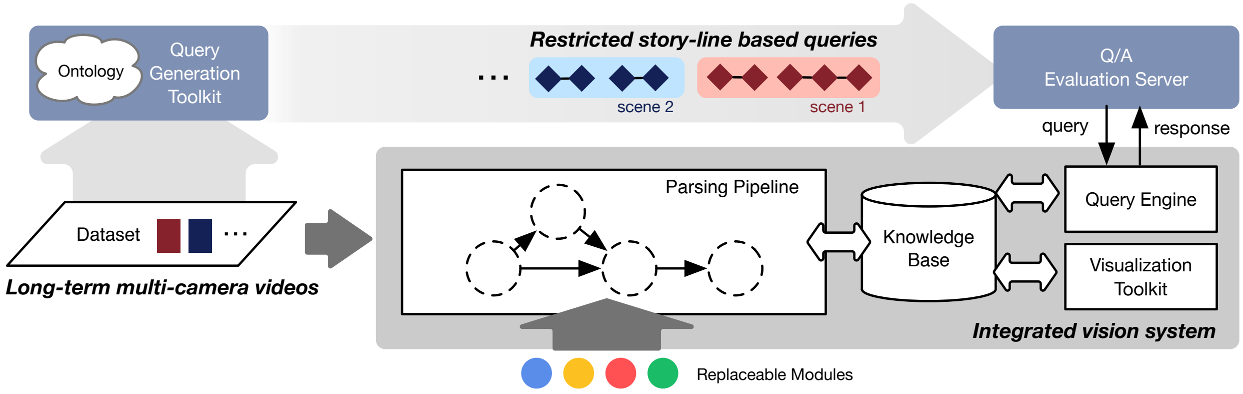


Figure 2.3: Overview of the restricted visual Turing test.

Figure 2.3 illustrates an overview of the proposed benchmark and system that consists of four components:

(i) Multi-camera video dataset collection. Existing datasets are either focusing on single individual images or short video sequences with clear action or event boundaries. Our multiple-camera video dataset includes a rich set of activities in both indoor and outdoor scenes. Videos are collected by multiple cameras with overlapping field-of-views during the same time window. A variety types of sensors are used: stationary HD video cameras located on the ground and rooftop, moving cameras mounted on bicycles and automobiles, and infrared cameras. The camera parameters are provided as meta data. The videos capture daily activities of a group of people and different events in a scene which include routine ones (e.g., an ordinary group lunch, playing four square soccer game) and abnormal ones (e.g., evacuating from a building during a fire alarm) with large appearance and structural variations exhibited.

(ii) Ontology guided storyline-based QA collection. We are interested in a selected ontology as listed in Figure 2.2. The ontology is sufficiently expressive to represent different aspects of spatial, temporal, and causal understanding in videos from basic level (e.g., identifying objects and parts) to fine-grained level (e.g., does person A have a clear-line-of-sight to person B?). Based on the ontology, we build a toolkit for collecting storyline-based queries and grounding annotations for each predicates. Queries organized in multiple storylines are designed to evaluate a computer vision system from basic object detection queries to more complex relationship queries, and further probe the system's ability in reasoning from the physical and social perspectives, which entails human-like commonsense reasoning. Cross-camera referencing queries requires the ability to integrate visual signals from multiple overlapping sensors.

(iii) Integrated vision system. We build a computer vision system that can be used to study the organization of modules designed for different tasks and interactions between them to improve the overall performance. It is designed with two principles in mind: first, well-established computer vision tasks shall be incorporated so that we can built upon the existing achievements; second, the modules shall be loosely coupled so that it allows user to replace one or more modules with alternatives to study the performance in an integrated environment. We define a set of APIs for individual tasks and connect all modules into a pipeline. After the system has processed the input videos and saved the results in a knowledge base, it fetches queries from the evaluation server one after another during the evaluation.

(iv) Evaluation server. We provide a web service API through which a computer vision system can interact with the evaluation server over HTTP connections. The evaluation server iterates through a stream of queries grouped by scenes. In each scene, queries are further grouped into storylines. A query is not available to the system until the previous storylines and all previous queries in the same storyline have finished. The correct answer is provided to the system after each query. This information can be used by the system to be adaptive with the ability to learn from the provided answers. The answer can be used to update the previous understanding such that any conflict has to be resolved and wrong interpretations can be discarded.

In this rest of this chapter, we will focus on the dataset and query characteristics of the proposed restricted visual Turing Test benchmark. The system architectural designs will be discussed in the next chapter.

2.2 Dataset

In this section, we introduce the video dataset we collected for our benchmark. Due to the space limit, we show the summarized characteristics of our dataset only and more examples of the videos will be presented in the supplementary material.

In our dataset, we organize data by multiple independent scenes. Each scene consists of video footage from eight to twelve cameras with overlapping fields of view during the same time period. By now, we have a total number of 14 collections covering both indoor and outdoor scenarios. Table 2.1 gives a summary of the data collections.

Our dataset reflects real-world video surveillance data and poses unique challenges to modern computer vision algorithms:

Varied number of entities. In our dataset, activities in the scene could involve individuals as well as multiple interacting entities.

Rich events and activities. The activities captured in the dataset involve different degrees of complexities: from the simplest single-person actions to the group sport activities which involve as many as dozens of people.

Unknown action boundary. Unlike existing action or activity dataset where each action data point is well segmented and each segment only contains one single action, our dataset consists of multiple video streams. Actions and activities are not pre-segmented and multiple actions may happen at the same time. Such characteristic preserves more information about the spatial context of one action and correlation between multiple actions.

Multiple overlapping cameras. This requires the system to perform multi-object tracking across multiple cameras with re-identification and 3D geometry reasoning.

Varied scales and view points. Most of our data are collected in 1920x1080 resolution, however, because of the difference in cameras' mounting points, a person who only occupies a couple of hundred pixels in bird's-eye views may occlude the entire view frame when he or

	Type	Cameras	Length	Major events and activities
		(Moving)	hh:mm:ss	
1	Indoor	9	8:27:23	Meetings, package exchange
2	Indoor	12	17:35:36	Meetings, card game, group lunch, coffee break
3	Indoor	10 (1)	2:29:50	Classroom routines, lectures
4	Indoor	11 (1)	8:53:24	Registration, classroom routines, lectures, evacuation
5	Outdoor	9 (1)	2:41:24	Parking lot routines
6	Outdoor	11 (2)	8:15:44	Parking lot routines
7	Outdoor	9	2:22:00	Four square game
8	Outdoor	11 (2)	8:14:42	Various group ball games, bicycle races
9	Outdoor	11 (1)	13:15:06	Various group ball games, auto repair
10	Outdoor	11 (1)	4:27:44	Parking lot routines, auto repair
11	Outdoor	7 (1)	1:57:01	Picnic, gardening, walking dogs
12	Outdoor	10 (2)	6:54:38	Picnic, gardening, preaching
13	Outdoor	8 (1)	3:27:00	Single-person exercises, ball and Frisbee games
14	Outdoor	8 (2)	4:15:56	Group exercises, fashion contest, ball and Frisbee games
		Total	93.5 hours	

Table 2.1: Summary of our dataset

she stands very close to a ground camera.

Illumination variation. Areas covered by different cameras have different illumination conditions: some areas are covered by dark shadows whereas some other areas have heavy reflection.

Infrared cameras and moving cameras. Apart from regular RGB cameras, our dataset include infrared cameras in some scenes as a supplementary. Moving cameras (i.e., cameras mounted on moving objects) also provide additional challenges to the dataset and reveal more spatial structure of the scene.

The complexity of our dataset. To demonstrate the difficulties of our dataset, we conduct a set of experiments on a typical subset of data using the state-of-the-art object detection models [RHG15] and multiple-object tracking methods [PRF11]. A summary of the data and results are shown in Tables 2.2 and 2.3, respectively.

Dataset	Fashion	Sport	Evacuation	Jeep
Cameras	4	4	4	4
Length (mm:ss)	4:30	1:35	3:00	3:35
Frames	32,962	11,798	21,830	25,907

Table 2.2: Summary of the selected subset of data.

2.3 Queries

In our framework, we support both formal language and natural langue queries. They are collected at the same time using a unified query collection tool. In this section, we first introduce the format of formal language queries and then describe our tool for collecting queries and groundings.

Fashion	Detection AP	MOTP	MOTA
View-CT2	0.475	0.692	0.341
View-HC2	0.413	0.674	0.304
View-HC3	0.635	0.692	0.494
View-IP1	0.485	0.694	0.339
Sport	Detection AP	MOTP	MOTA
View-CT2	0.554	0.728	0.413
View-HC2	0.596	0.727	0.483
View-HC3	0.534	0.716	0.430
View-IP1	0.694	0.739	0.573
Evacuation	Detection AP	MOTP	MOTA
View-HC3-6	0.518	0.698	0.389
View-HC4-6	0.556	0.692	-0.241
View-IP2	0.534	0.720	0.346
View-IP5	0.533	0.651	0.399
Jeep	Detection AP	MOTP	MOTA
View-GL1-2	0.252	0.680	0.172
View-GL2-2	0.250	0.651	0.170
View-GL5	0.280	0.689	0.203
View-GL6	0.389	0.696	0.270

Table 2.3: Results from detection and tracking. *For Detection:* AP of all object occurrence is calculated as in PASCAL VOC 2012 [EEV14] based on results by Faster R-CNN [RHG15]. *For Tracking:* Accuracy (MOTA) and Precison (MOTP) are calculated as in Multiple Object Tracking Benchmark [LMR15] based on results by [PRF11]

2.3.1 Formal Language Queries

A formal language query is a first-order logic sentence (with modification) composed using variables, predicates (as shown in Figure 2.2), logical operators ($\wedge, \vee, \neg$), arithmetic operators, and quantifiers ($\exists$ and $\forall$). The answer to a query is either true or false meaning whether the fact stated by the sentence holds given the data and the system's state of belief. The formal language representation eliminates the need of natural language processing and allows us to focus computer vision problems on a constrained set of predicates.

We evaluate computer vision systems by asking a sequence of queries organized into multiple storylines. Each storyline explores a natural event across a period of time in a way similar to conversations between humans. At the beginning of a storyline, major objects of interest are defined first. The vision system under evaluation shall indicate whether it detects these objects. A correct detection establishes a mutual conversation context for consecutive queries, which ensures the vision system and queries are referring to the same objects in later interactions. When the system fails to detect an object, consecutive queries regarding that object will be skipped.

Object predicates. To define an object, specifications of object type, time, and location are three components. Object type is specified by object predicates in the ontology. A time t is either a view-centric frame number in a particular video or a scene-centric wall clock time. A location is either a point (x, y) or a bounding box (x_1, y_1, x_2, y_2) represented by its two diagonal points, where a point can be specified either in view-centric coordinates (i.e. pixels) or in scene-centric coordinates (i.e. latitude-longitude, or coordinates in a customized reference coordinate system, if defined). For example, an object definition query regarding a person in the form of first-order logic sentence would look like:

$$\exists p \quad \text{person}(p; \text{time} = t; \text{location} = (x_1, y_1, x_2, y_2))$$

when the designated location is a bounding box.

Attribute and relationship predicates. Attribute and relationship predicates are used to explores a system's spatial, temporal, and causal understanding of events in a scene regarding the detected objects. The query space consists of all possible combinations of predicates in the ontology with the detected objects (and/or objects interacting with the detected ones) being the arguments. When expressing complex activities or relationships, multiple predicates are typically conjuncted to form a query. For example, suppose M_1 and F_1 are two detected people, the following query states "M_1 is a male, F_1 is a female, and there is a clear line of sight between them at time t_1":

$$\text{male}(M_1) \wedge \text{female}(F_1) \wedge \text{clear-line-of-sight}(M_1, F_1; \text{time} = t_1).$$

Note that the location is not specified, because once M_1 and F_1 is identified and detected, we expect the vision system can track them over space and time.

Moreover, storylines unfold fine-grained knowledge about the event in the scene as it goes. In particular, given the detected objects and established context, querying about objects interacting with the detected ones becomes unambiguous. As in the example shown in Figure 3.1, even the ball is not specified by any object definition queries (and actually it is hard to detect the ball even if the position is given), once the two people interacting with the ball are identified, it becomes legitimate to ask if "the female catches a ball at time t_2":

$$\exists b \quad \text{ball}(b) \wedge \text{catching}(F_1, b; \text{time} = t_2),$$

and if "the male and female are playing a ball game together over the period of t_1 to t_2":

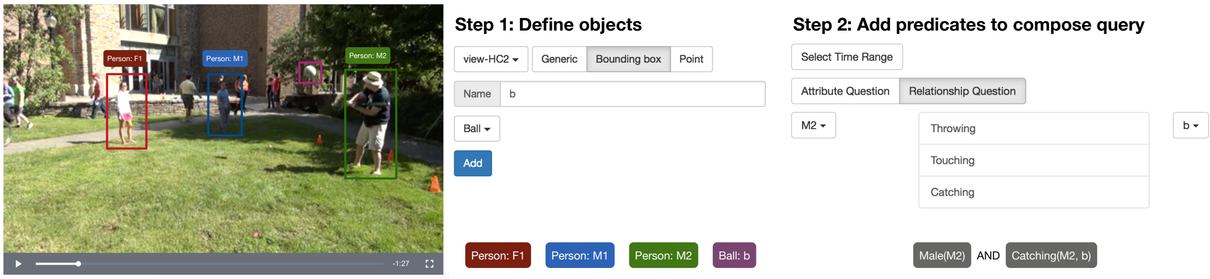
$$\text{game}(M_1, F_1; \text{time} = (t_1, t_2)).$$

Times and locations are specified the same way as in object definition queries with an extension that a time period (t_1, t_2) can be specified by a starting time and a ending time.

Correctly answering such queries is non-trivial as it requires joint cognitive reasoning based on spatial, temporal, and casual information across multiple cameras over a time period.

2.3.2 Queries and Groundings Collection

To collect queries and grounding annotations for objects and relationships, we design and implement a query collection toolkit. When composing a query, we first define and annotate



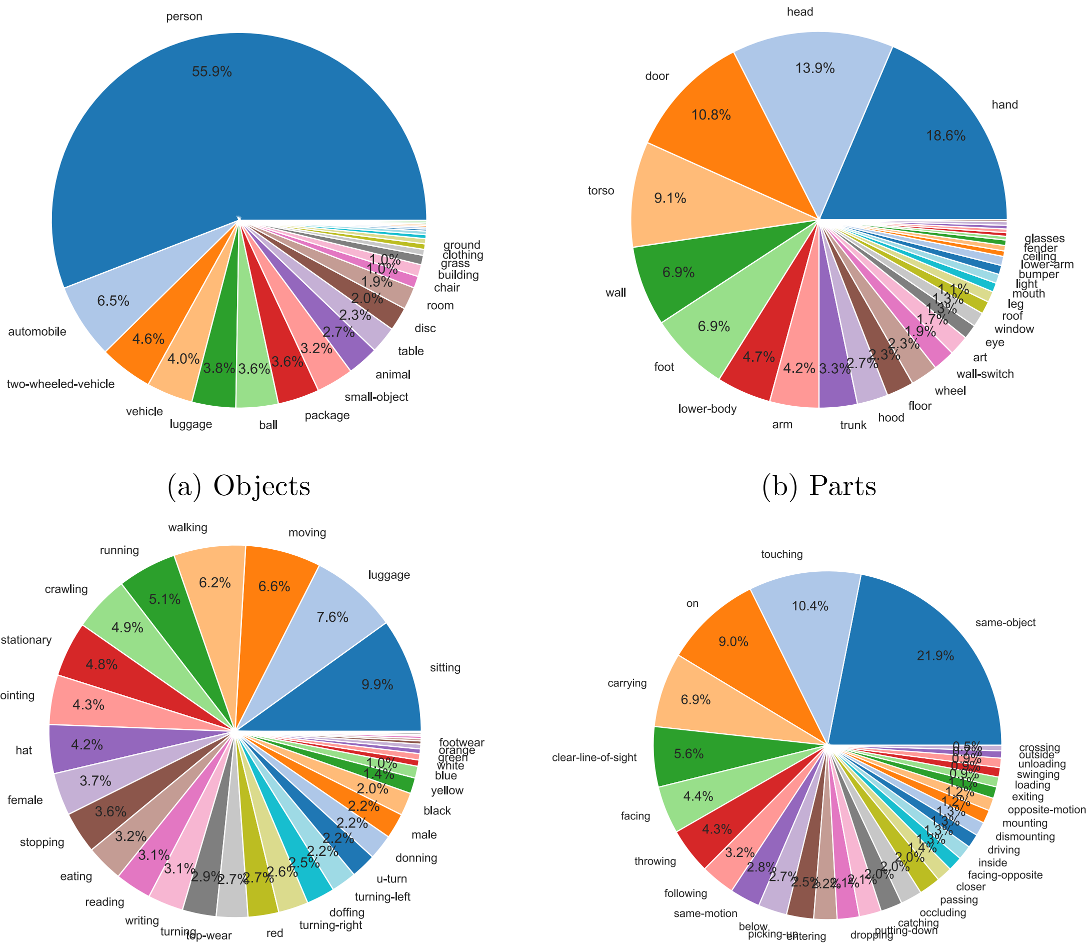
Name b

Male(M2) AND Catching(M2, b)

Figure 2.4: An example of composing queries using our query collection toolkit

the objects of interests. Our tool allows annotators to draw bounding boxes and points to refer to specific objects and move the annotated boxes along the video timeline to generate a ground-truth track. Tracks from different views can also be associated with same identity for collection cross-view tracking ground-truth. After the objects are annotated, we obtain a list of object predicates with groundings. Next step is to compose queries by concatenating an arbitrary number of attribute or relationship predicates. Each predicate is annotated with a binary label “true” or “false” indicating whether the objects involved in the predicate satisfy the relationship, this serves as the grounding of attributes and relationships of objects. To ensure the collected queries are meaningful, we constrain the possible choices for each argument of a predicate so that the allowed combinations will always represent conceptually correct relationships that align with commonsense. This lowers the bar for educating annotators and make it possible to adopt this tool to crowdsourcing platforms like Amazon Mechanical Turk. Figure 2.4 illustrates an example of this process. For each query, we also collect a ground-truth answer and a sentence that is the natural language equivalent to the first-order form.

Currently, we have created 3,426 queries in the dataset. Figure 2.5 shows the distribution of predicates in selected categories. Though we try to be unbiased in general, we do consider some predicates are more common in and important than others and thus make the distribution non-uniform. For example, among all occurrence of object predicates, “person” takes 55.9%, which is reasonable because human activities are our major point of interest.



(a) Objects

(b) Parts

(c) Attributes

(d) Relationships

Figure 2.5: Distribution of predicates in each category.

2.4 Summary

We described a restricted visual Turing test scenario that evaluates computer vision systems across a wide task spectrum with a domain ontology and explicitly tests the grounding of concepts with formal queries. Given a set of videos of a scene and a sequence of queries organized into storylines, the task of the restricted Turing test is to provide answers either simply in binary form “true/false” or in natural language. View-centered queries focus on evaluating visual parsing from particular camera views, whereas scene-centered queries involve data fusion and joint inference across different cameras.

The data and queries distinguish us from existing scenarios such as visual question answering in images and video captioning. In contrast to multi-way classification and sequence prediction problems, we emphasize explicit groundings of concepts in a restricted ontology via formal language queries. Our framework evaluates computer vision systems across the task spectrum and emphasizes a joint spatial, temporal, and causal understanding of visual scenes in multi-camera cross-view videos.

In the next chapter, we will present a modular system architecture that integrates various vision modules and evaluate its performance in the restricted visual Turing test.

CHAPTER 3

Joint Parsing System

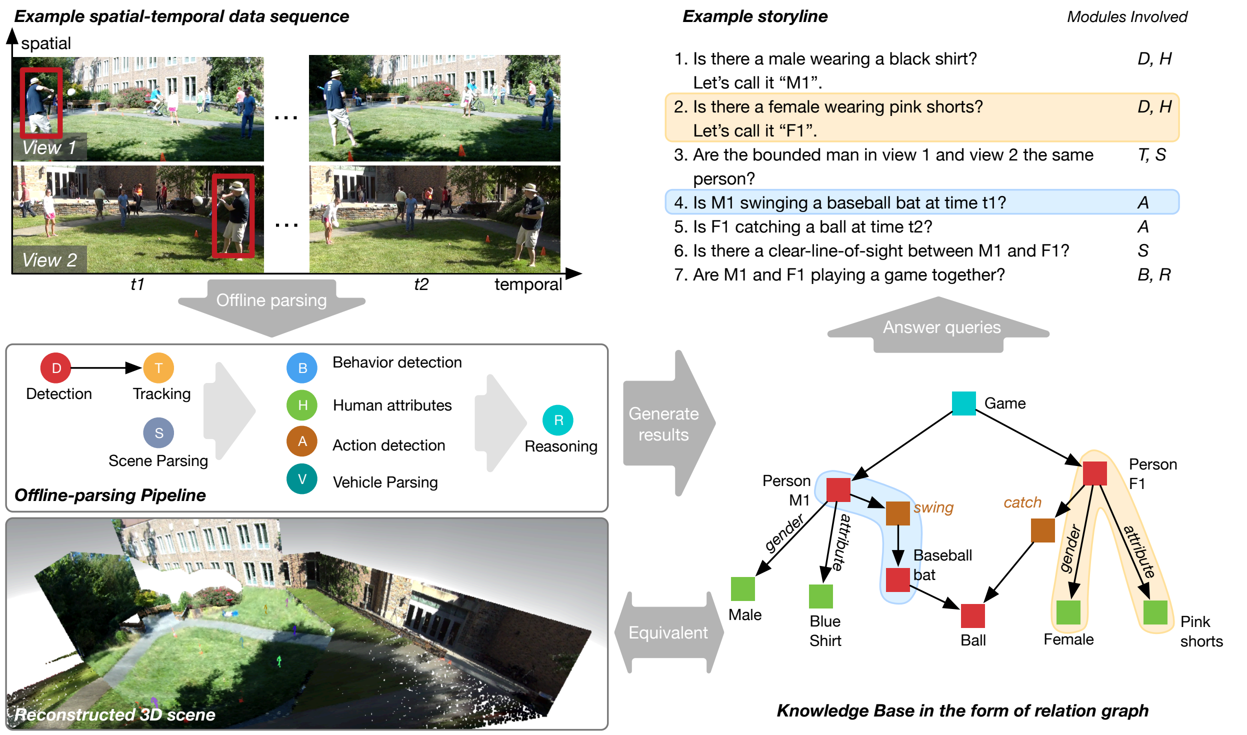
3.1 Introduction

Approaches proposed for image captioning and VQA are primarily based on the combination of convolutional neural network [LBD89, KSH12] and recurrent neural network like long short-term memory [HS97], which formulate the problem as sequence prediction or multi-way classification. In contrast to end-to-end approaches, we take an explicit approach to build a joint parsing system which integrates various vision modules. The architecture supports symbolic reasoning on results generated by individual modules. We are interested in whether a computer vision system can further unfold the intermediate representation to explicitly show how it derives the answer, and if so it enhances the "trust" that we have on the system that it has gain a correct understanding of the scene.

Our joint parsing system consists of three major components: an offline parsing pipeline which decompose the visual perception into multiple sub-tasks, a knowledge base which stores parsing results (including entities, properties, and relations between them), and a query engine which answers queries by searching the knowledge base. The system also features a flexible architecture and a visualization toolkit. Figure 3.1 shows an example of a full workflow of our system.

3.2 Parsing Pipeline

Offline parsing pipeline processes the multiple-view videos. Each view is first processed by a *single-view parsing pipeline* where video sequences from multiple cameras are handled



1. Is there a male wearing a black shirt? Let's call it "M1". — D, H
2. Is there a female wearing pink shorts? Let's call it "F1". — D, H
3. Are the bounded man in view 1 and view 2 the same person? — T, S
4. Is M1 swinging a baseball bat at time t1? — A
5. Is F1 catching a ball at time t2? — A
6. Is there a clear-line-of-sight between M1 and F1? — S
7. Are M1 and F1 playing a game together? — B, R

Figure 3.1: Illustration of our prototype vision system. *Top-left:* input videos with people playing baseball games. *Middle-Left:* Illustration of the offline parsing pipeline which performs spatial-temporal parsing in the input videos. *Bottom-Left:* Visualization of the parsed results. *Bottom-Right:* The knowledge base constructed based on the parsing results in the form of a relation graph. *Top-Right:* Example storyline and queries. Graph segments used for answering two of the queries are highlighted.

independently. Then *multiple-view fusion* matches tracks from multiple views, reconciles results from single-view parsing, and generates scene-based results for answering questions.

To take advantage of achievements in various sub-areas in computer vision, we organize a pipeline of modules, each of which focuses on one particular group of predicates by generating corresponding labels for the input data. Every module gets access to the original video sequence and products from previous modules in the pipeline. The implemented modules are described as follows. Most components are derived from the state-of-the-art methods at the time we developed the system last year and are pre-trained on other datasets.

Scene parsing generates a homography matrix for each sensor by camera calibration and also produces estimated depth map and segmentation label map for each camera view. The implementation is derived from [LZZ14].

Object detection [SWJ13, RHG15] processes the video frames and generates bounding boxes for major objects of interest.

Multiple object tracking [PRF11] generates tracks for all detected objects.

Human attributes [PZ15] classifies appearance attributes of detected human including gender, color of clothes, type of clothes, and accessories (e.g. hat, backpack, glasses).

Action detection detects human actions and poses in the scene. The implementation is derived form [XXZ15, YNL14, WKS11].

Behavior detection parses human-human, human-scene, and human-object interactions.

Vehicle parsing [WLZ15, HZ15, HR12] produces bounding boxes and fluent labels for specific parts of detected cars (e.g. fender, hood, trunk, windows, lights).

Multiple-view fusion merges the tracks and bounding boxes from multiple views based on appearance and geometry cues.

The middle-left part of Figure 3.1 shows the dependencies between these modules in the system.

3.3 Knowledge Base and Query Answering

We employ a generic graph-based data model to store knowledge. The detected objects, actions, attribute labels are all modeled as nodes, the connections between them are modeled as edges. In our implementation, the parsing results are stored into Resource Description Framework (RDF) graphs [W3Ca], in the from of triple expressions, which can be queried by a standard query language SPARQL [W3Cb]. Given that the questions are formal language, our query engine first parses the query and transforms the query into a sequence of SPARQL statements. Apache Jena [McB02] is used to execute these statements and to return answers derived from the knowledge base. Figure 3.2 shows the architecture of query engine.

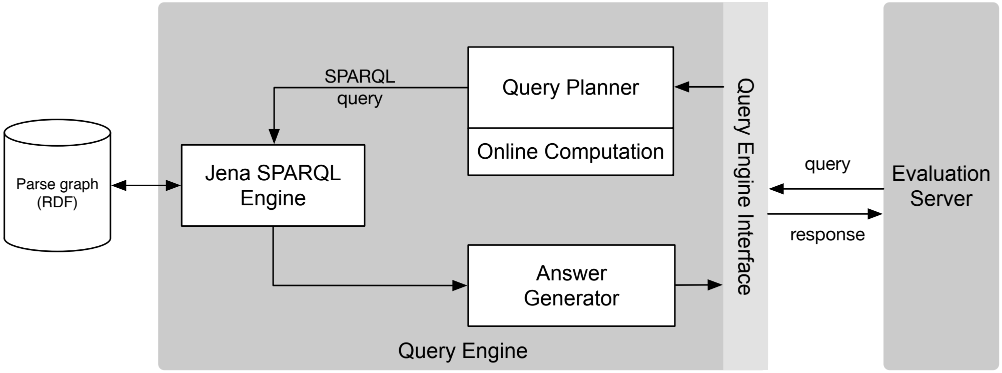


Figure 3.2: The architecture of query engine.

In practice, it is infeasible to pre-calculate all possible predicates and save each individual knowledge segment into the knowledge base. For example, pre-calculating all `following(x, y)` relationships would involve pair-wise combination across all detected humans. This strategy is obviously inefficient in that the portion of data being queried with this predicate is actually sparse. Alternatively, we designed an *online computation* protocol which evaluates binary and ternary relationships only at the testing time when such predicates appear in a query.

3.4 Online Computation

To make offline parsing efficient, we leave some parsing tasks to be only performed when a related predicate is received. In the current design, such parsing tasks usually involve predicates between two entities. At the offline parsing phase, enumerating all possible pairwise combinations and testing such predicates against all pairs can be computational expensive. Hence, this design is a trade-off between the parsing-time complexity and evaluation-time complexity.

For example, for predicate `together(x, y)`, which is true if two entities `x` and `y` are physically located nearby each other, the complexity of testing pairwise combination is $O(n^2)$ in a video sequence containing n objects. However, in the evaluation stage, the predicates in a query usually only involves a small number of objects constrained in a limited time range. Hence, delaying the evaluation of such predicates could reduce the computation efforts significantly.

We can interpret this architecture as a combination of *bottom-up* and *top-down* approaches. The offline parsing phase can be considered as a bottom-up parsing process in which we explore an initial set of knowledge including the positions (e.g. bounding box, contours), characteristics (e.g. color) and status (e.g. moving or stationary) of entities by perception modules. Whereas at the query answering stage, when encountering a query that involves a predicate that is not evaluated we explore it on the fly by online computation. This top-down process only performs specific computation on a limited subset of entities within a bounded time interval.

3.5 Implementation Details

The system is designed with two goals bearing in mind: first, we want to incorporate existing tasks in computer vision; second, the architecture shall be flexible enough for replacing a module with alternatives to pursuit incremental improvements later. To this end, we defined a set of APIs for each vision task and connect all the modules using remote procedure calls (RPC). This enables the system to only focus on the logical connection between modules and provides the implementation flexibility for individual components. This design allows us to use this system as an experiment platform by switching between alternative models and implementations for studying their effects and contributions to query answering. To make the system easy to use, we also developed a dashboard with visualization tools for rapid development and experiment.

We developed a SDK for modules implemented in various programming languages, including C++, Python, Java, and MATLAB to be used in our systems. In our system implementation, we use Apache Thrift [SAK07] as the underlying serialization and RPC framework. It allows us to define the shared data structure and service interface through an interface description language that abstracts away the dependencies on programming languages. By implementing bindings to multiple programming languages, modules written in different languages can be plugged into the system.

To support different use cases, every module can be executing in two modes:

(i) **Batch processing mode** allows a module to fetch inputs from the system (optionally intermediate results produced by other modules) and to produce a new pack of result managed by the system to be consumed by other modules in the pipeline. Execution are usually triggered from the dashboard.

(ii) **Online service mode** lets a module listen requests from end-user or other modules. This is particular useful in the online computing scenario discussed in the previous subsection. In addition, this mode allows frequently-used lightweight operations to be shared as on-demand services, which also help reducing the system's workload in managing intermediate results.

3.6 Evaluation

We evaluated our prototype system using a subset of the dataset (see the upper parts in Table 3.1) and 1,160 polar queries. During the evaluation, our system did not utilize the ground-truth answers after answering each query for consecutive queries. Among the 1,160 queries, 243 queries are object definitions, 197 (81%) of which are successfully detected. For non-definition queries, we either provided binary "true/false" answers or claimed "unable to respond" (when our implementation cannot handle or recognize some of the predicates involved in a query). Table 3.1 shows the accuracy as the ratio of correctly answered queries to number of the responded non-definition queries.

Figure 3.3 further breakdowns the accuracy by the category of predicates and the number of unique predicates in a query. Most queries have either one, two, or three predicates. This is a natural result of the choice to avoid over-complicating the queries. Queries with one predicate focus on various types of objects (people, car, etc.): most of these queries (243) are object definitions; the others (46) are about counting (e.g., "how many people are in the scene?"). Queries with two predicates mostly involve attributes and properties of single objects: one predicate of the two is used to define the object (usually person or automobile), the other unary predicate focuses on attributes. Queries with three predicates focus on binary relationships operating on two objects: two predicates are used to define the operands and the third predicate is for relationships. The results reveal that our prototype system performs well in object detection tasks and also indicate room for improvements for complex queries regarding spatial reasoning and interactions between entities.

	Office	Parking lot (winter)	Parking lot (fall)	Garden	Auditorium
Video length	17:35:36	8:14:42	4:27:44	4:15:56	8:53:24
# of cameras	12	12	11	8	11
# moving cameras	0	2	1	1	2
# IR cameras	0	1	1	0	1
# of queries	108	247	236	215	254
Definition queries	-	63	71	54	55
Non-definition queries	108	184	165	161	199
Respond rate	0.522	0.600	0.795	0.683	0.731
Accuracy	0.785	0.615	0.626	0.586	0.684

Table 3.1: Performance by data collection.

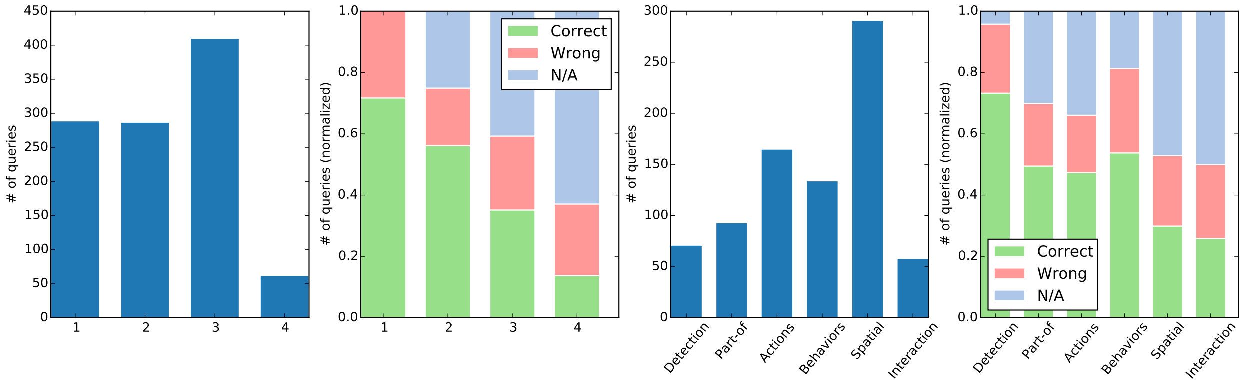


Figure 3.3: Experiment results breakdown. *Left*: accuracies by the number of unique predicates in a query. *Right*: accuracies by the category of predicates.

3.7 Summary

We discussed a scalable system design for visual scene understanding. It features a modular and decoupled architecture to integrate off-the-shelf computer vision modules. Such configuration allows us to leverage the advancements from research on various computer vision tasks and studies the performance of a joint computer vision system explicitly. The unified graph-based knowledge representation together with system SDK enable information sharing between multiple modules. With the system evaluated in the restricted visual Turing test, the strengths and weaknesses of the system are make explicit for future research.

In the following chapters, we will dive into two topics in detail. First, in Chapter 4, we will formulate a cross-view joint parsing problem that fuses view-centric proposals from multiple modules and cameras into a set of consistent scene-centric beliefs. Then in Chapter 5, we will discuss the underlying formalism of parse graph knowledge bases and discuss an alternative implementation of question answering with graph matching.

CHAPTER 4

Scene-centric Joint Parsing of Cross-view Videos

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 attributes from a scene-centric perspective. For example, Figure 4.1 shows a parse graph hierarchy that describes a scene where two people are playing a ball. In the first view,

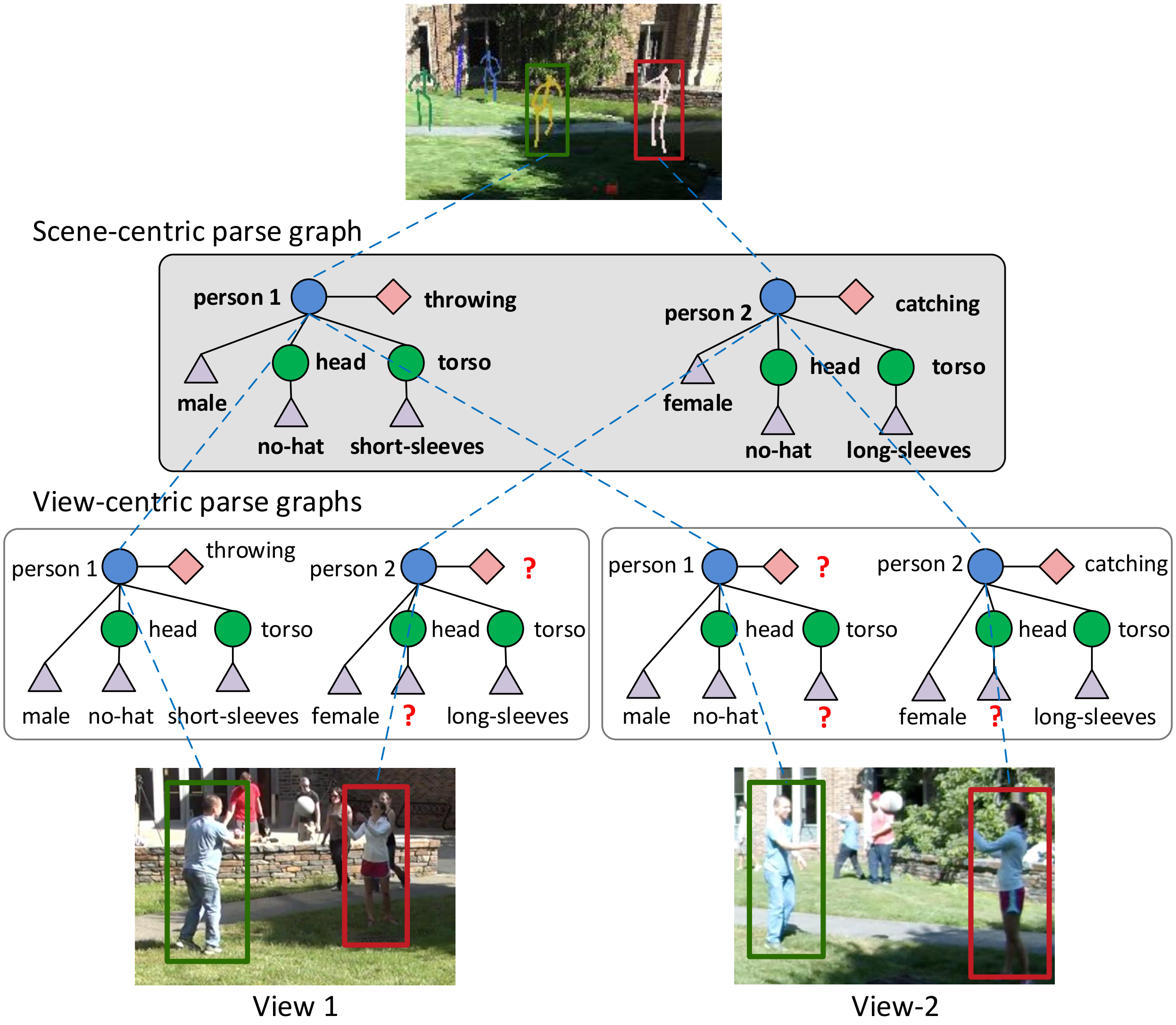


Figure 4.1: An example of the spatio-temporal semantic parse graph hierarchy in a visual scene captured by two cameras.

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 an 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.

4.2 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, Figure 4.2 shows an ontology graph that describes a domain including 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

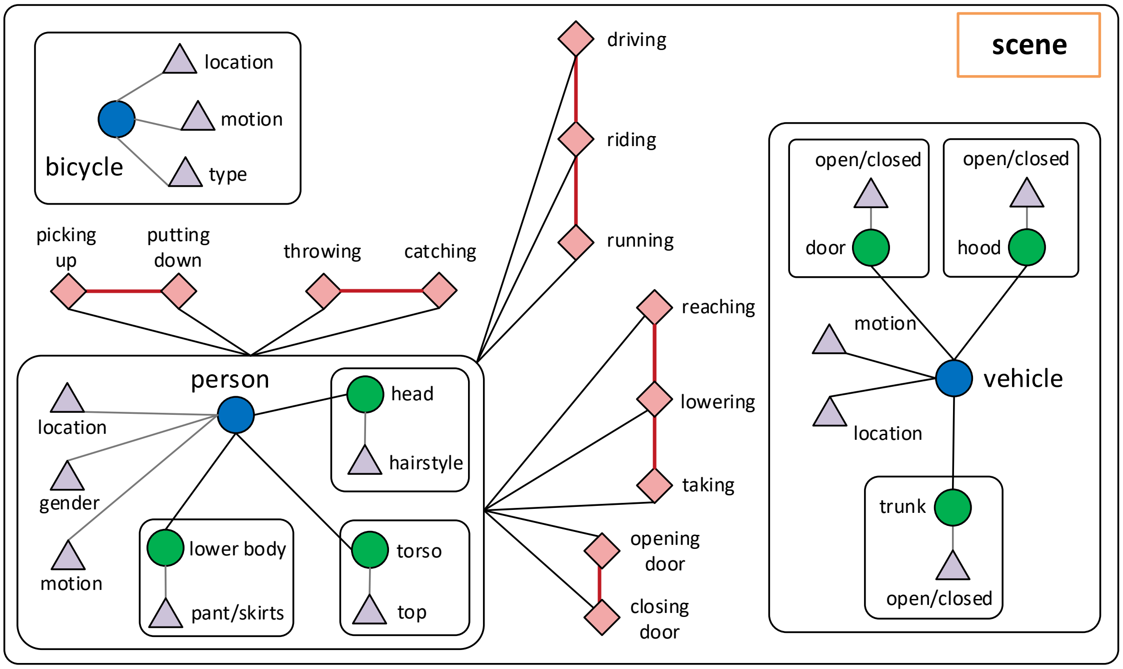


Figure 4.2: An illustration of the proposed ontology graph describing objects, parts, actions and attributes.

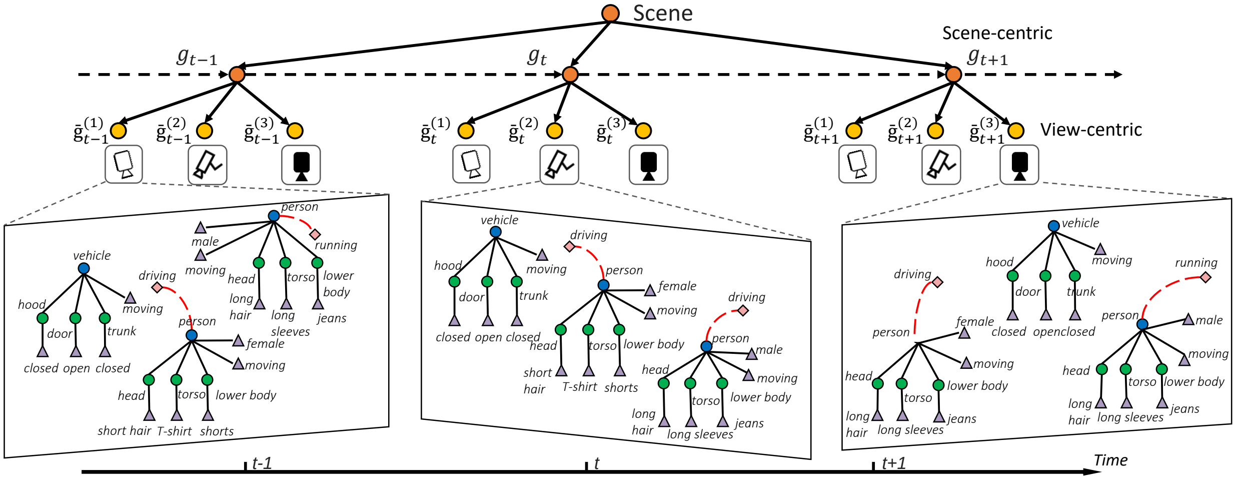


Figure 4.3: The proposed spatio-temporal parse graph hierarchy.

among action nodes denote their incompatibility. The ontology graph can be considered a compact AOG [LZZ14, WZZ16] without the compositional relationships and event hierarchy. In this work, 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 work, we refer to such subgraphs as *parse graphs*.

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 Figure 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, we use a tilde notation to represent the view-centric concepts $\tilde{x}$ corresponding to scene-centric concepts x.

4.3 Probabilistic Formulation

The task of joint parsing is to infer the spatio-temporal parse graph hierarchy

$$G = \langle \Phi, g, \tilde{g}^{(1)}, \tilde{g}^{(2)}, \ldots, \tilde{g}^{(M)} \rangle$$

from the input frames from video sequences $I = \{I_t^{(i)}\}$ captured by a network of M cameras , 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 4.4.

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

$$G^* = \arg\max_G p(I|G) \cdot p(G). \tag{4.1}$$

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

$$\begin{aligned} p(I|G) &= \prod_{i=1}^{M}\prod_{t=1}^{T} p(I_t^{(i)}|\tilde{g}_t^{(i)}) \\ &= \prod_{i=1}^{M}\prod_{t=1}^{T}\prod_{v\in V(\tilde{g}_i^{(t)})} p(I(v)|v), \end{aligned} \tag{4.2}$$

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

$$p(G) = p(g_1)\prod_{t=1}^{T-1} p(g_{t+1}|g_t)\prod_{i=1}^{M}\prod_{t=1}^{T} p(\tilde{g}_t^{(i)}|g_t), \tag{4.3}$$

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).

$$\begin{aligned} p(\tilde{g}|g) &= \frac{1}{Z}\exp\{-\mathcal{E}(g,\tilde{g})\} \\ &= \frac{1}{Z}\exp\{-w_1\mathcal{E}_S(g,\tilde{g}) - w_2\mathcal{E}_A(g,\tilde{g}) \\ &\qquad - w_3\mathcal{E}_{Act}(g,\tilde{g}) - w_4\mathcal{E}_{Attr}(g,\tilde{g})\}, \end{aligned} \tag{4.4}$$

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.

4.3.1 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.

$$\mathcal{E}_A(g,\tilde{g}) = \sum_{o \in g} ||\phi(o) - \phi(\tilde{o})||_2, \tag{4.5}$$

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:

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

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:

$$\mathcal{E}_{Act}(g, \tilde{g}) = \sum_{l \in g} -\log p(l|\tilde{p}), \tag{4.7}$$

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:

$$\mathcal{E}_{Attr}(g, \tilde{g}) = \sum_{a \in g} \mathbf{1}(a \neq \tilde{a}) \cdot \xi, \tag{4.8}$$

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

4.4 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.

4.4.1 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]:

$$\alpha(G \to G') = \min\left(1, \frac{q(G' \to G) \cdot p(G'|I)}{q(G \to G') \cdot p(G|I)}\right), \tag{4.9}$$

where $p(G|I)$ the posterior is defined in Eqn. (4.1).

4.4.2 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 Figure 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 algorithms.

4.5 Experiments

4.5.1 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.

The **TUM Kitchen** dataset [TBB09][2] is an action recognition dataset that contains

[1] http://bitbucket.org/merayxu/multiview-object-tracking-dataset

[2] http://ias.in.tum.de/software/kitchen-activity-data

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.

4.5.2 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 Eqn. (4.2). 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, 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

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.

	CAMPUS						TUM Kitchen								
Methods	Run	PickUp	PutDown	Throw	Catch	Overall	Reach	Taking	Lower	Release	OpenDoor	CloseDoor	OpenDrawer	CloseDrawer	Overall
view-centric	0.83	0.76	0.91	0.86	0.80	0.82	0.78	0.66	0.75	0.67	0.48	0.50	0.50	0.42	0.59
baseline-vote	0.85	0.80	0.71	0.88	0.82	0.73	0.80	0.63	0.77	0.71	0.72	0.73	0.70	0.47	0.69
baseline-mean	0.86	0.82	1.00	0.90	0.87	0.88	0.79	0.61	0.75	0.69	0.67	0.67	0.66	0.45	0.66
scene-centric	0.87	0.83	1.00	0.91	0.88	**0.90**	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.

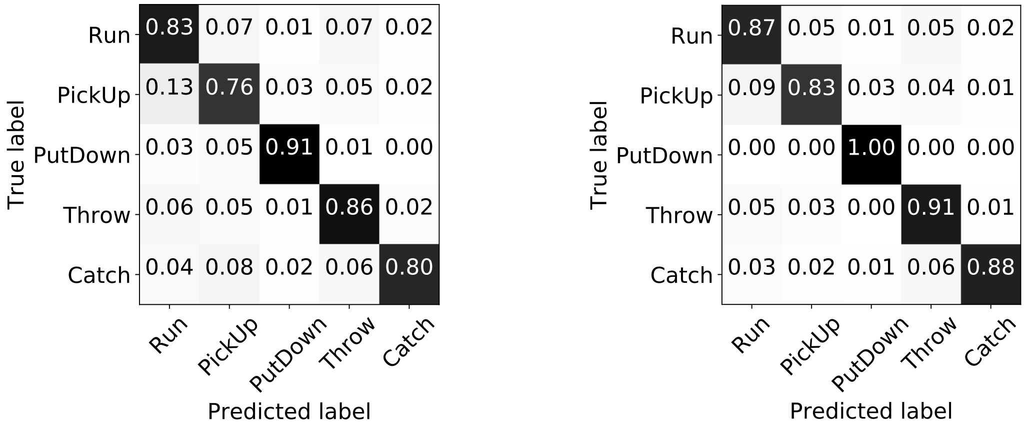


Figure 4.4: Confusion matrices of action recognition on view-centric proposals (left) and scene-centric predictions (right).

(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, Low-

	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.

ering, Releasing, OpenDoor, CloseDoor, OpenDrawer, and CloseDrawer. We measure both individual accuracies for each category as well as the overall accuracies across all categories. 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. Figure 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 (Figure 4.5). Observing an entity from more cameras generally leads to better performance, while too many conflicting observations may also cause degraded performance. Figure 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

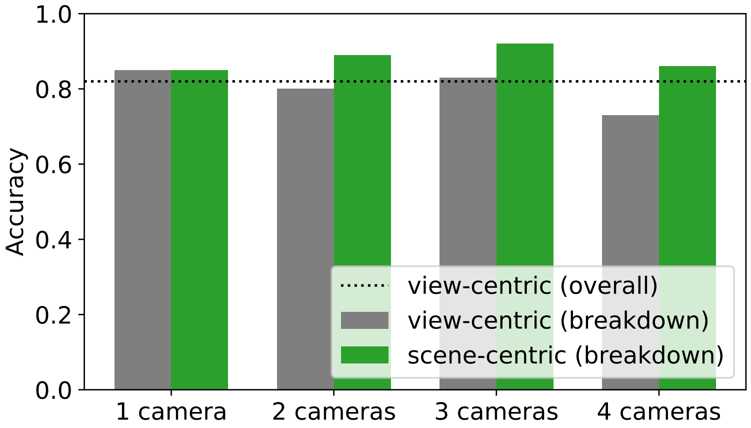


Figure 4.5: The breakdown of action recognition accuracy according to the number of camera views in which each entity is observed.

collected for both CAMPUS and TUM Kitchen dataset. View-centric proposals and score are obtained from an attribute grammar model as in [PNZ16]. 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.

4.5.3 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,

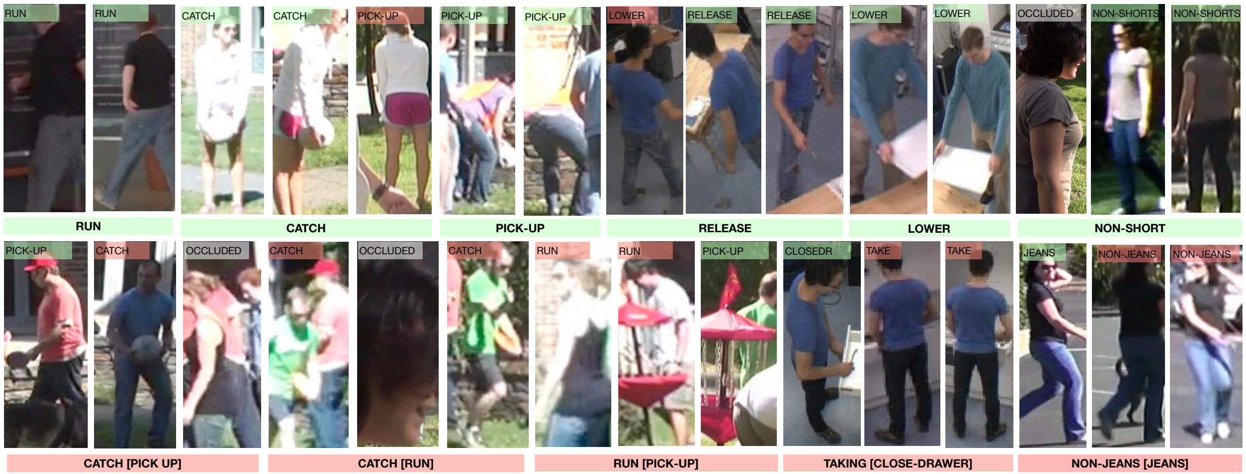


Figure 4.6: Success (1st row) and failure examples (2nd row) of view-centric (labels overlaid on the images) and scene-centric predictions (labels beneath the images) of action and attribute recognition tasks. For failure examples, true labels are in the bracket. "Occluded" means that the locations of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

more efficient MCMC algorithms may be adopted. For example, the mini-batch acceptance testing technique [CSP16] has demonstrated several order-of-magnitude speedups.

4.6 Summary

In this chapter, we focused on the joint parsing problem of fusing inconsistent and noisy view-centric proposals from various modules and camera-views into a consistent set of scene-centric beliefs about the visual scene. We described a parse graph hierarchy as a formal knowledge representation for scene understanding in cross-view videos. Joint parsing is formulated as an MAP problem of inferring the structure and values of the parse graph hierarchy given the initial proposals from view-centric modules. A probabilistic model is developed to capture the appearance and geometry constraints among objects observed at multiple views and the semantic constraints among different properties of objects.

CHAPTER 5

Parse Graph Knowledge Base and Question Answering

5.1 Introduction

We have discussed a general scenario called restricted visual Turing test and a joint parsing framework that leverages pre-trained computer vision models for visual scene understanding. In this chapter, we focus on formally describe a structured storage for parse graphs as a knowledge base that bridges the parsing phase of the system and additional applications that use language as interface, such as question answering.

Parse graphs, as a form of knowledge representation, is an important mean of information sharing and reusing among multiple intelligent agents for communication and cooperation. In this chapter, we are interested in developing a general principle of constructing a parse graph knowledge base for visual scene understanding using a property graph model. The structure of the knowledge base are closely related to the first-order logic. Specifically, predicates are represented by nodes and arguments of predicates are modeled as edges.

The parse graph knowledge base is a semantic bridge. Rather than viewing language as the outputs of a model as in [FHS10, AAL15], we think of language as an interface for information exchange. In our framework, every query to our knowledge base can be written as a first-order logic statement. By converting the statement into a graph fragment, we can retrieve answers by graph matching in the knowledge base. Figure 5.1 illustrates the idea. With a common ontology that defines the scope and schema of the knowledge base, multiple agents or applications can interact with the parse graph knowledge base.

Our main contributions include developing a principled method connected to first-order logic for storing spatio-temporal parse graphs in a knowledge base using property graph

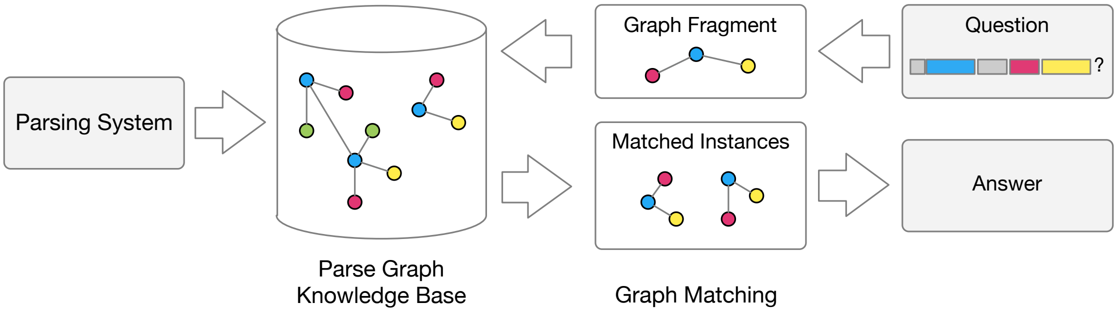


Figure 5.1: Graph matching in the parse graph knowledge base.

model. In addition we develop a system that uses the knowledge base for question answering with graph matching.

This chapter is organized as follows. We discuss preliminaries in Section 5.2. Then we describe the parse graph knowledge base construction in Section 5.3. In Section 5.4, we present graph matching techniques for question answering. Finally, we discuss potential future use cases on reasoning in Section 5.5.

5.2 Preliminaries

5.2.1 Ontology and Parse Graphs

Ontology is an explicit specification of domain conceptualization [Gru95]. In our work, it defines the scope of our representation on scenes and events. Concretely, it characterizes a set of plausible objects, actions, and attributes in visual scenes understanding applications. As discussed in Chapter 4, we define an ontology as a graph that contains nodes representing objects, parts, actions, attributes respectively and edges representing the relationships between nodes. To capture semantic structures of objects in the real world, we also include two special types of relationships: `IsA` for class taxonomy hierarchy and `PartOf` for compositional relationships between object and semantic parts.

While an ontology describes plausible elements, only a subset of these concepts can be true for a given instance at a given time. Therefore, the ontology can be viewed as the metadata or schema for parse graphs, which is the agent's view of the world after the input images or videos are parsed. In the context of visual scene understanding, the ontology summarizes the labels that the modules in the system can potentially produce. For question answering, the ontology characterizes the scope of questions that the system can handle. On the other hand, parse graphs are produced by the system and its sub-modules. Answers to questions are generated from a knowledge base that stores parse graphs.

Ontology is closely related to the system organization. In general, top-down and bottom-up approaches can be adopted. The former organizes sub-modules of the system after an ontology is determined at the first place (by the design from domain expert or automatically generated from web or external databases), while the latter grows the ontology from the capability of all system components. In practice, we take a hybrid approach that starts from an initial ontology designed by domain experts and makes adjustments as the domain requirements or the capabilities of system components change.

5.2.2 RDF Triples

One popular choice for graph-structured knowledge representation is using a set of *triples*, each defined as (subject, predicate, object). The Semantic Web [DMV00] community uses Web Ontology Language (OWL) [Gro12] and Resource Description Framework (RDF) [W3Ca, BG14] to pursuit semantic interoperability of the Web. SPARQL [W3Cb] is a standard tool to query RDF knowledge bases. However, although entities are logically connected, triple-based graph databases typically store each triple as an individual artifact instead of storing the graph as a connected structure. In addition, each node in the RDF triple representation is atomic. When representing data with internal structures, such as a list of properties, the RDF triple model requires using multiple triples that results in an significantly increased size of the knowledge base.

5.2.3 Property Graph Model

In this work, we adopt the property graph model as an alternative to store parse graphs. Like ordinary graphs, a property graph consists of a set of **nodes** and directed **edges** (also called **relationships**), each of which connects two end nodes. However, every node and edge in a property graph can be associated with a list of properties in the form of key-value pairs. In addition, nodes and edge can be optionally attached with one or more labels for richer semantic representations in a dialet called labeled property graph. In practice, we use Neo4j [Dev12] and Cypher, its corresponding query language, as underlying tools for managing and querying parse graph knowledge bases, respectively.

5.3 Parse Graph Knowledge Base

When building parse graph knowledge bases using the property graph model, we aim to (i) map an ontology into a property graph knowledge base in a principled way, (ii) store all concepts in scene-centric and view-centric parse graphs into the knowledge base without loss of information, (iii) develop a method to retrieve grounded understanding of visual scenes from the parse graph knowledge base. We address the first two goals in this section and the last goal in the next section.

We start with formally defining the property graph structure of an ontology. We first describe a potential modeling ambiguity issue and then present how to resolve it by establishing formal connections with first-order logic. With a principled way to map an ontology to a property graph, we then present how to store scene-centric and view-centric concepts in a parse graph hierarchy discussed in the previous chapters.

5.3.1 Graph Structure

Given an object concept in an ontology, it is straightforward to represent that concept with a single node with the property graph model. However, given an relationship, there could exist multiple seemly equivalent ways to represent it. For example, consider `Driving(person,` vehicle) where Driving is a binary relationship whose agent is a person and patient is a vehicle. Figure 5.2 shows two modeling choices. As one option (Figure 5.2(a)), driving can be modeled as a node with two out-going edges each pointing to one participant in this activity. Alternatively, driving is modeled as a directed edge that connects the agent and the patient, as in Figure 5.2(b).

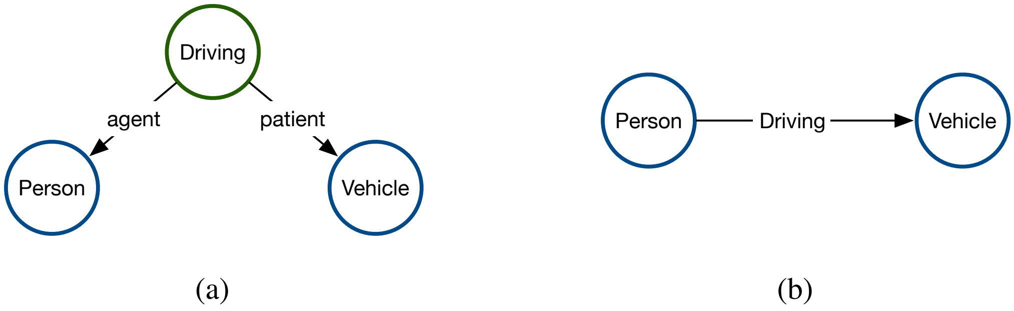


Figure 5.2: Two equivalent ways to model binary relationships: (a) As a node with two out-going edges. (b) As an edge connecting two nodes.

In this example, the two options give the same capacities in expressing this relationship and the option (b) uses exactly one node and one edge less than (a). However, in the context of our work, we always prefer option (a) where relationships are modeled as nodes, rather than edges. The disadvantage of using edges is that edges inherently constraint every relationship to have two and only two participants. Consider a ternary relationship Together(person1, person2, person3) which describes a possible world where three persons are spatially close to each other. A single labeled edge will not be sufficient to connect all three person nodes; whereas together, if modeled as a node, can be connected with all three entities each with an edge.

Now we describe a formal principle behind this modeling decision. The fundamental idea is to establish a correspondence between ontology and first-order logic. First-order logic syntax contains three basic symbols: constant (representing concrete objects), predicates (representing relationships), and functions (mappings between objects). They can be used to compose statements that describe possible worlds. For an given ontology, we consider every concept of objects and relationships (includes attributes, actions, and other spatio-temporal relationships) in the ontology as a predicate.

Parse graph contains grounded concepts. Every grounded object concept is considered as a constant, every grounded relationship is a statement that evaluates to true given the operands. When building a labeled property graph knowledge base from parse graphs, for each grounded concept (objects and relationships), we create a node in the knowledge base and use edges to link relationship nodes to the corresponding object nodes where the relationship predicates evaluate to true. All nodes are labeled with the concept name in ontology. Table 5.1 lists the correspondence between ontology concepts and first-order logic syntax. Figure 5.3 illustrates the primary subgraph structures in the knowledge base.

Ontology concepts	First-order logic	Example
Objects	Predicate	`Person(x)`
Attributes	Predicate (unary)	`IsMale(x)`
Action	Predicate (unary)	`Running(x)`
Relationship	Predicate (binary)	`Driving(x, y)`
Relationship	Predicate (n-ary)	`Together(x, y, z)`
Grounded objects	constants	`p1, v3`
	true predicate statements	`Person(p1), Vehicle(v3)`
Grounded relationships	true predicate statements	`Driving(p1, v3)`

Table 5.1: Correspondence between ontology concepts and first-order logic syntax. Every concept in the ontology is considered a predicate. Grounded objects are considered constants.

In addition to semantic attributes and relationships defined in ontology for visual understanding, we define two binary relationships to represent object taxonomy and part hierarchy, respectively. As an exception, these binary relationships are modeled as edges for conciseness.

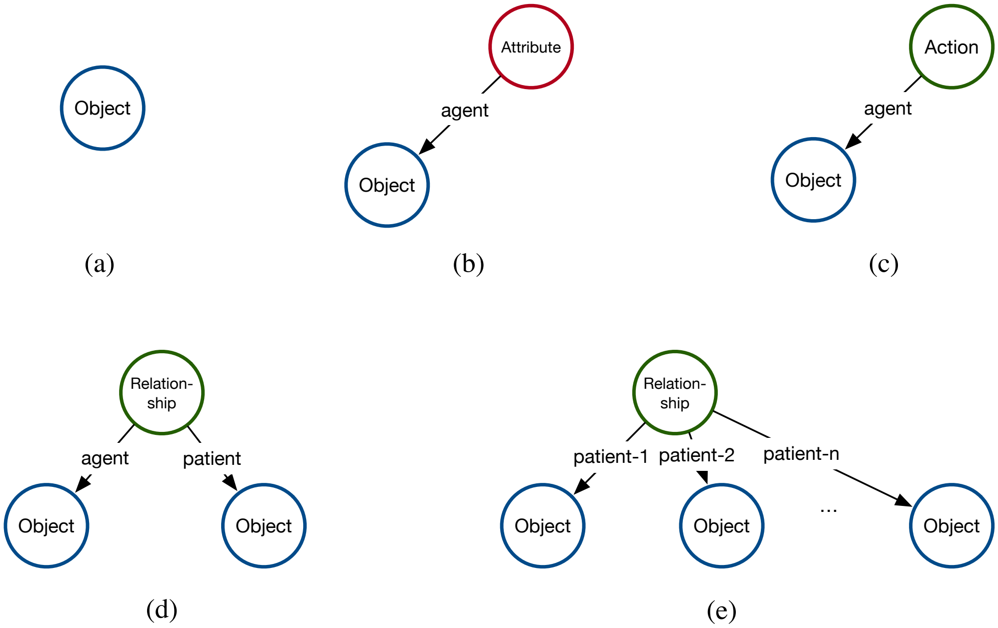


Figure 5.3: Primary subgraph structures in property graph knowledge bases. Every grounded concepts will be stored as a node. (a) An object. (b) An attribute. (c) An action. (d) A binary relationship. (e) A n-ary relationship.

Object Taxonomy. We introduce an

```
IsA(low-level concept, high-level concept)
```

relationship to represent the taxonomy relationship between low-level concepts and high-level concepts. For example, Figure 5.4(a) shows an graph fragment that represents a vehicle taxonomy. This relationship is particularly useful for answering queries formulated with only high-level concepts while the parse graph contains only low-level concepts.

Part Hierarchy. Spatial parse graphs (S-pg) represents the compositional relationships of a scene [ZZ11] or human [RPZ13]. We introduce a binary relationship to model this part hierarchy

```
PartOf(part, whole).
```

Figure 5.4(b) shows an example that decomposes a detected person into a part hierarchy.

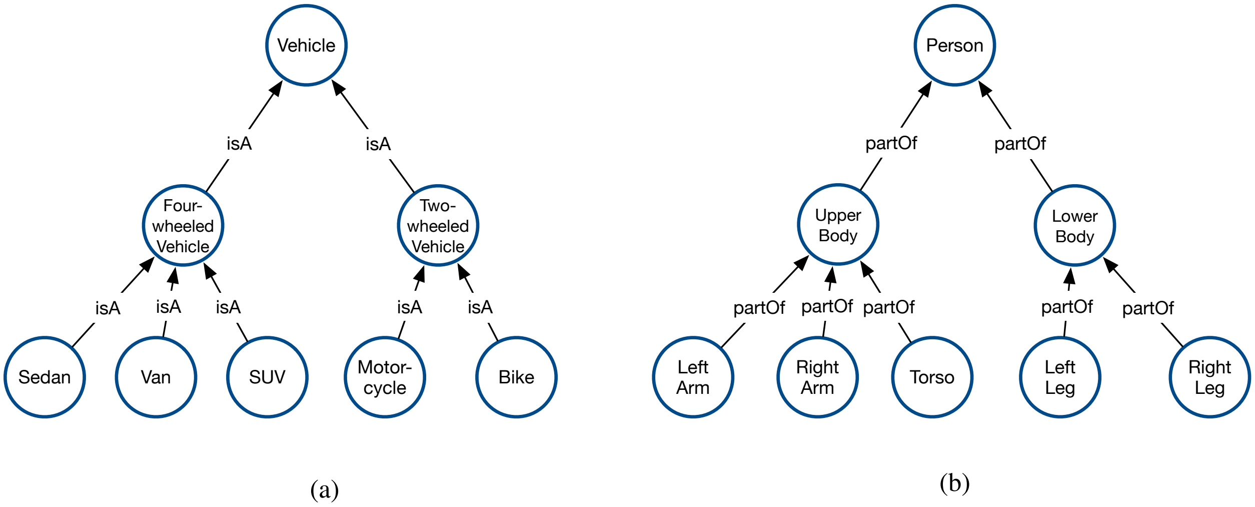


Figure 5.4: Examples of (a) class taxonomy hierarchy and (b) part hierarchy.

Note that the taxonomy and part hierarchy relationships are transitive. Formally,

$$\forall \texttt{x, y, z} \;\; \texttt{IsA(x, y)} \wedge \texttt{IsA(y, z)} \Rightarrow \texttt{IsA(x, z)}, \tag{5.1}$$

likewise for PartOf relationships.

5.3.2 Property List

With each node in a parse graph knowledge base representing a grounded concept, we store grounding details in the internal structure of each node as a list of properties. Concretely, for each object and its bounding box node, the properties characterize the grounding details such as scene-centric locations or view-centric bounding box coordinates, detection labels, and confidence scores. For each attribute, action, and relationship node, the properties include the start and end timestamps, the camera view it is observed, additional labels and scores. Figure 5.5 shows example properties for a various types of nodes. In practice, the exact list of properties to include depends on the application and domain specification.

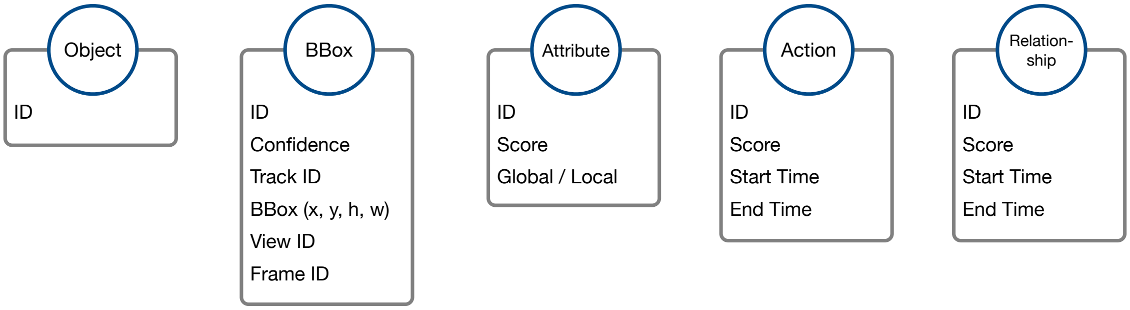


Figure 5.5: Example properties associated with different types of nodes.

5.3.3 Scene-centric and View-centric Hierarchy

Temporal information is retained by creating a key-frame nodes for every key frame where the concepts is grounded to data by computer vision modules. The granularity can be varied according the application requirements. In practice, we typically store a grounded node about every 10 frames. To characterize that a set of nodes grounded at multiple different timestamps have the same identity, we create a entity node that is connected to every grounded key-frame nodes with an edge (optionally labeled with `observedAt` for rich semantic).

In multi-view scenarios, we generalize the concept of key-frame nodes to view-centric nodes by attaching additional grounding information that characterizes from which camera view the concept is grounded. Hence, a view-centric key-frame node represents a local belief of the scene from a specific camera view at a fixed time stamp, while the entity nodes captures the system's scene-centric belief. The view-centric and scene-centric information together contributes to a comprehensive understanding of a scene, since view-centric nodes contain details about what local predictions were made by the underlying modules given a concrete piece of data and scene-centric nodes present an aggregated view, which helps answering queries such as counting the number of persons without duplicates. Figure 5.6 illustrates how scene-centric and view-centric representation are modeled with property graph.

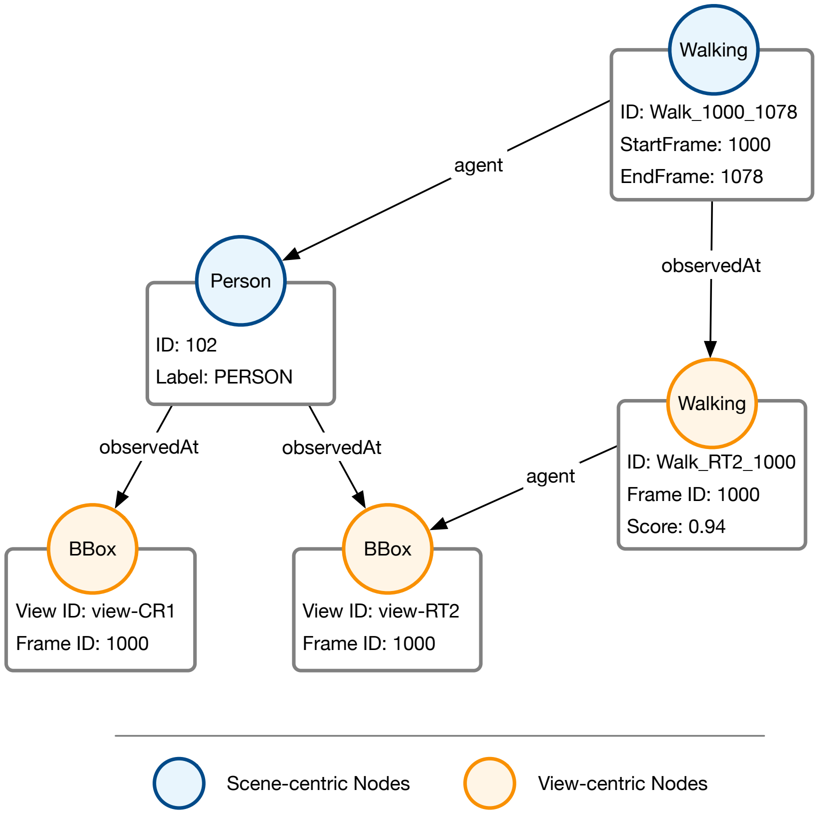


Figure 5.6: Scene-centric and view-centric representations.

5.4 Question Answering by Graph Matching

In this section, we present a graph-matching based method to retrieve grounded understanding of visual senses from parse graph knowledge bases in question answering scenarios. In contrast to open-ended question answering, we consider a constrained question answer scenario where the question domain is restricted by an ontology. Since we use an ontology to specify the possible objects and relationships in visual scenes, all questions can be asked are also characterized by the same ontology. Fundamentally, we view every question as a set of fragments of the ontology. By doing so, the answer to the question is all the parse graph fragments in the knowledge base that matches the pattern of the question graph.

Formally, we define a question q to be a set of graph fragments that are subgraphs of the ontology

$$q \triangleq \{g : g \subset G_{\text{ontology}}\}. \tag{5.2}$$

Accordingly, we define the answer $\mathcal{A}(q)$ to a question q to be a set of graph fragments in the parse graph knowledge base that is isomorphic to question graph

$$\mathcal{A}(q) \triangleq \bigcup_{g \in q} \{a : a \simeq g, a \subset pg\}. \tag{5.3}$$

In our system, we support both formal language questions and natural language questions. We first discuss the idea of translating the questions into formal first-order logic statement that corresponds to a graph patterns. Then we provide implementation details of utilizing graph matching to retrieve answers from a parse graph knowledge base.

5.4.1 Formal Language Questions

We begin with formal language questions written in the form of first-order logic statement, where each question is expressed by a conjunction of predicates

$$\exists x_1, \ldots, x_m \quad p_1(x_1) \wedge \ldots \wedge p_k(x_i, x_j) \ldots \wedge p_n(x_m), \tag{5.4}$$

where the quantified variables specifics the objects to be fetched[1]. The answer to the question would be a set of bindings between the fetching variables in the query and constants in knowledge base so that all predicates evaluate to true. With the correspondence principle described in the previous section, the formal language query can be converted to a graph pattern by creating a node for every predicates and linking relationships and its object operands with edges.

Objects. When querying the existence of objects without additional predicates, we simply define the query to be the object predicate itself. The corresponding graph fragment is simply a single node. For instance, `Person(x)` and `Vehicle(y)` queries all the person nodes and vehicle nodes, respectively.

Attributes. Attributes predicates impose additional constraints on the objects in queries. Formally, an attribute predicate shall take the the same variable as the object predicate which it modifies. For example, the following formal language statement queries persons

[1] When there is no ambiguity, we omit the quantifier and fetching variables for brevity.

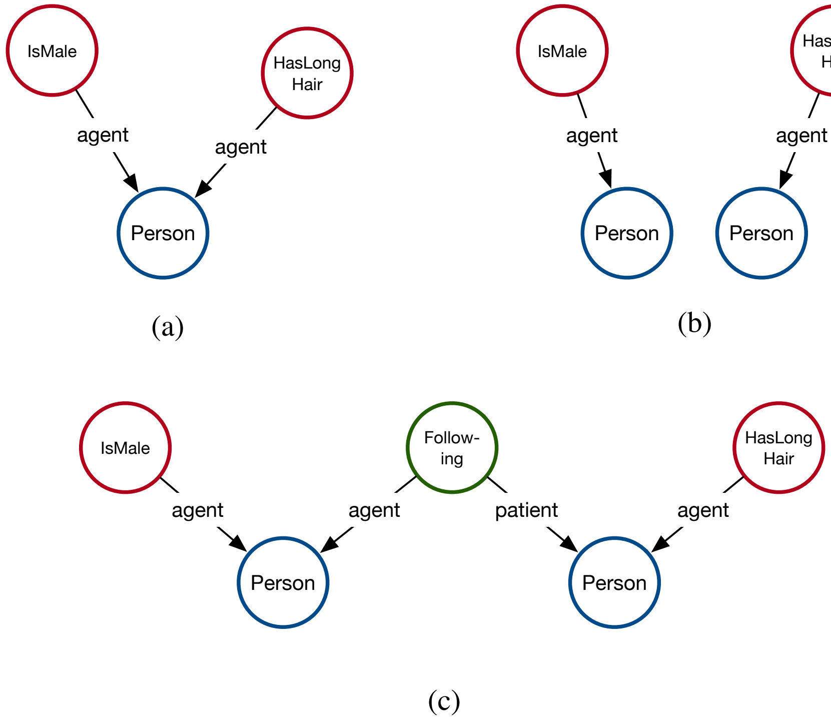
who possess two different attributes:

$$\mathtt{Person(x)} \land \mathtt{IsMale(x)} \land \mathtt{HasLongHair(x)}, \tag{5.5}$$

whereas a slightly modified version queries two persons with different attributes:

$$\mathtt{Person(x)} \land \mathtt{IsMale(x)} \land \mathtt{Person(y)} \land \mathtt{HasLongHair(y)}. \tag{5.6}$$

Figure 5.7(a) and (b) shows the graph fragments corresponding to these two queries, respectively.



(a)

(b)

(c)

Figure 5.7: Graph fragments correspond to attribute queries. (a) Query 5.5. (b) Query 5.6. (c) Query 5.8.

Actions and relationships. Unary action predicates is similar to attributes predicates as it only has one agent. For example, the following query uses the action predicate together with the object predicate to fetch all walking persons:

$$\mathtt{Person(x)} \land \mathtt{Walking(x)}. \tag{5.7}$$

Binary action predicates takes two arguments and therefore the corresponding node connects to two object node in the query graph (Figure 5.7(c))

$$\texttt{Person(x)} \wedge \texttt{IsMale(x)} \wedge \texttt{Person(y)} \wedge \texttt{HasLongHair(y)} \wedge \texttt{Following(x, y)}. \tag{5.8}$$

5.4.2 Natural Language Questions

Natural language questions also fit in our framework of graph-matching based question answering so that the answers can be retrieved from the same parse graph knowledge. Recall that, in contrast to open-ended question answering or human-machine dialogue, the domain is restricted and has a formal structure with respect to an ontology. Given the formalism defined in the formal language and its underlying first-order representation, our approach is to cast the natural language questions into formal graph patterns.

Rather than focusing on the linguistic details of the input natural language questions, we pursuit a structural understanding of the natural language query so that a correspondence between the natural language and formal structure of the parse graph knowledge base can be established. We adopt the text parsing approach in [HLJ09, TML14] where the natural language input is first parsed into a dependency tree representation [DM08] with the terminal terms mapped to concepts in the ontology. The connectivity between concepts are directly informed by the dependencies tree. For simple queries that mainly contain attributes and unary action predicates, tracking a small set of dependencies edges (such as `nsubj`, `attr`, `amod`) can reduce the structure to the corresponding graph form that complies with the ontology. Figure 5.8 shows examples of dependency trees and the corresponding graph pattern for natural language queries. For more versatile natural language questions, a semantic extraction grammar can be learned as shown by [HLJ09].

5.4.3 Implementation Details

In this section, we describe the implementation details of our query answering system with Neo4j [Dev12] and its the corresponding query language Cypher.

Being a declarative query language, Cypher specifies a query by defining a graph pattern

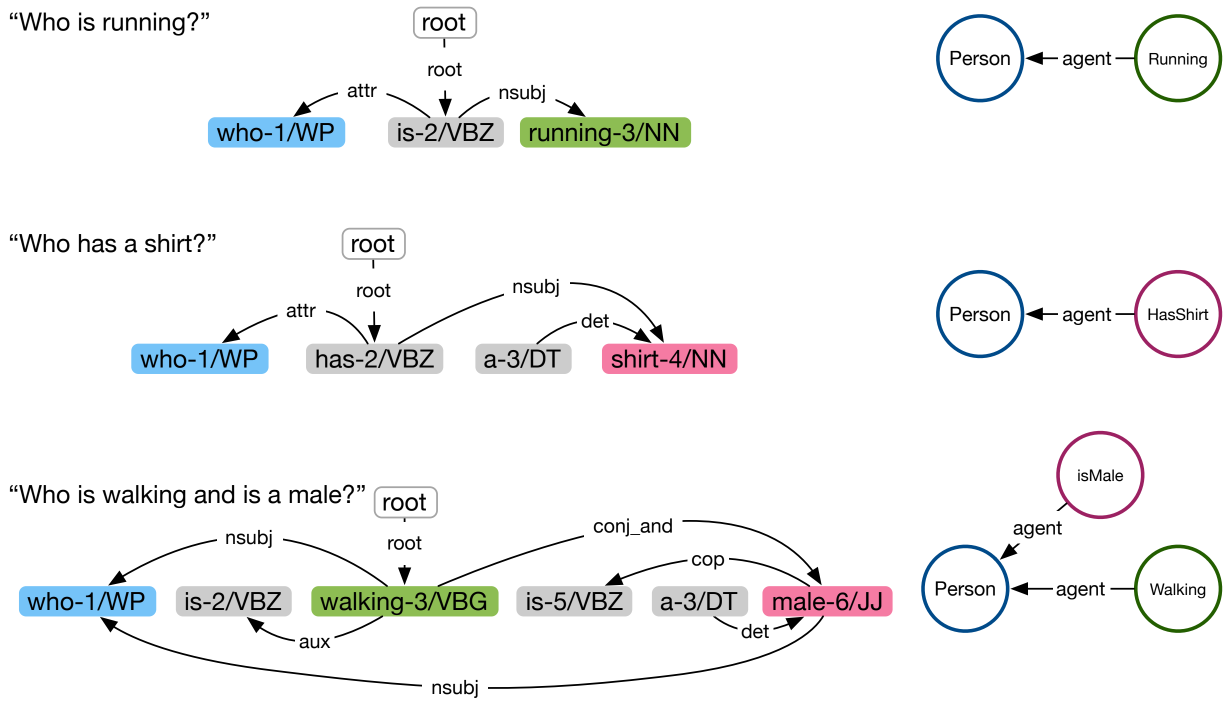


Figure 5.8: Example dependency trees and query graph fragment for natural language questions.

to be matched in the graph database. In the scope of our work, we primarily use the following clauses to compose queries:

- The `MATCH` clause specifies the pattern to be found in the database. It has an intuitive and expressive syntax to describe graph patterns. Concretely, bracket and squared bracket, such as `(:Person)` and `[:Agent]`, represent a labeled node and a labeled edge, respectively. Dashes together with greater-than or less-than signs, such as `-[:Agent]->`, represents an directed edge.

- The `WHERE` clause specifies conditions for filtering the results.

- The `RETURN` clause specifies which nodes and edges to be fetched after executing the query. All properties associated with the matched nodes and edges are included. This is particularly helpful for analyzing the how the concepts are grounded to the original input data. In the cases where unique results are desired, a `DISTINCT` keyword can be included in the `RETURN` clause.

We characterize the primary graph patterns used in our question answering system as follows.

5.4.3.1 Scene-centric queries

Scene-centric queries is one of the main types of queries in our system. These queries specifies graph patterns with scene-centric nodes, i.e. objects, scene-centric actions, and attributes. The answers to such queries reflect the global understanding of the scene. Typical use cases including counting entities that satisfies a condition, retrieving clips that contains certain actions.

Objects. Query for one type of objects is a simple match clause consisting only one node. For example, `Person(x)` directly translates to the a Cypher query as follows:

```
MATCH (x:Person)
RETURN x
```

Attributes. Each attribute query is expressed by an attributes nodes connected to the object nodes it modifies. Multiple attributes about the same object in a conjunction statement are represented as a connected graph fragment. For example, the first-order query at (5.5) can be converted into a Cypher query as follows.

```
MATCH (:IsMale)-[:Agent]->(x:Person)<-[:Agent]-(:HasLongHair)
RETURN x
```

Actions and Relationships. Similar to the attribute queries, unary actions connects to the agent of the action with an out-going edge. Optionally, a `WHERE` clause can be used to filter the matching pattern using global timestamps in the visual scene. As a concrete example, the following Cypher query finds all walking persons between frame 1000 and 1020 specified by the first-order equivalent:

$$\texttt{Person(p)} \wedge \texttt{Walking(p; time=[1000, 1020])}$$

```
MATCH (w:Walking)-[:Agent]->(p:Person)
WHERE w.StartFrame >= 1000 AND w.EndFrame <= 1020
RETURN p, w
```

Binary relationships have more than one out-going edges. As an example, formal language query

$$\texttt{Following(x, y)} \wedge \texttt{Person(x)} \wedge \texttt{Person(y)}$$

corresponds to a Cypher query consisting of a three-node pattern as follows.

```
MATCH (f:Following)-[:Agent]->(x:Person),
      (f:Following)-[:Patient]->(y:Person)
RETURN f, x, y
```

In applications, various types of predicates in an conjunction can be connected to the same graph fragment. For example, the following query contains an object node, an action node, and an attribute node.

```
MATCH (w:Walking)-[:Agent]->(p:Person)<-[:Agent]-(m:IsMale)
WHERE w.StartFrame >= s AND w.EndFrame <= e
RETURN p, w, m
```

5.4.3.2 View-centric queries

View-centric nodes at key frames contains detailed grounding information for each concept. We use view-centric queries to retrieve these view-specific information, such as the coordinates of a bounding box, the confidence score of a particular prediction.

For example, to retrieve all bounding boxes of a walking person between frame 1000 and 1020 observed from `view-RT2`, the following view-centric query statement includes a view-centric bounding box node labeled as `BBox`:

```
MATCH (w:Walking)-[:Agent]->(p:Person)-[:ObservedAt]->(b:BBox)
WHERE w.StartFrame >= 1000 AND w.EndFrame <= 1020 AND
      b.viewId = "view-RT2"
RETURN p, b, w
```

5.4.3.3 Taxonomy and part hierarchy queries

For questions formulated with high-level concepts, we use the syntax `-[:IsA*1..n]->` to allow for matching nodes that are up to n `IsA` edges away.

As a concrete example, the following query retrieves all moving vehicles. All subcategories of vehicle nodes (connected to a `Vehicle` node via multiple intermediate concepts with `IsA` edges) match with the pattern.

```
MATCH (m:Moving)-[:Agent]->(x)-[:IsA*]->(v:Vehicle)
RETURN x, m
```

Similarly, for part-of hierarchy, parts of a particular object at all granularity levels can be retrieved by using the `-[:PartOf*]->` syntax. Concretely, the following Cypher statement returns all detected parts of an object bounding box whose id is `100`.

```
MATCH (p)-[:PartOf*]->(b:BBox)
WHERE b.id = "100"
RETURN b, p
```

5.5 Reasoning Potentials

The correspondence between parse graphs and first-order logic allows additional capabilities beyond retrieval and taxonomic reasoning. In this section, we discuss a potential direction of extending the graph-based question answering from retrieval to reasoning with additional first-order clauses that capture commonsense knowledge.

The connection between the formulation of parse-graph based knowledge graph and first-order logic can support interesting higher level tasks such as reasoning with theorem proving. To see this, let knowledge base Δ_{pg} be the collection of grounded facts in parse graphs. Consider a query statement α, to find out whether the statement α is true in the knowledge base, we either directly query the knowledge base via retrieval,

$$\alpha \subset \Delta_{pg},$$

or we verify if it can be inferred from the knowledge base that the statement α is true

$$\Delta \vDash \alpha.$$

where $\Delta = \Delta_{pg} \cup \Delta_{\mathrm{c}}$ is the augmented knowledge base that consists of grounded concepts in the parse graphs Δ_{pg} and additional set of clauses Δ_{c} that captures commonsense knowledge regarding the problem domain.

Concretely, consider an example parse graph fragment and snapshot in Figure 5.9, the knowledge Δ_{pg} from the parse graph can be summarized as the following clauses

```
BaseballGame(P1, P2),
Person(P1), IsFemale(P1), Catching(P1),
Person(P2),  IsMale(P2), Swinging(P2),
TShirt(S), isBlue(S), Wearing(P1, S).
```

with `P1, P2, S` being object constants.

Pre-trained computer vision models may have failed to detect a baseball bat due to resolution or occlusion. To see how first-order logic reasoning can help with inferring such missing concepts, assume we have the following first-order clause that captures the commonsense knowledge that performing a swinging action implies holding a bat:

$$\forall p \quad \texttt{Swinging(p)} \Rightarrow \exists b \quad \texttt{Bat(b)} \wedge \texttt{Holding(p, b)}.$$

With this additional clause, the formal language query statement α:

$$\texttt{Person(x)} \wedge \texttt{TShirt(y)} \wedge \texttt{Wearning(x, y)} \wedge \texttt{IsBlue(y)} \wedge \texttt{bat(b)} \wedge \texttt{Holding(x, b)}.$$

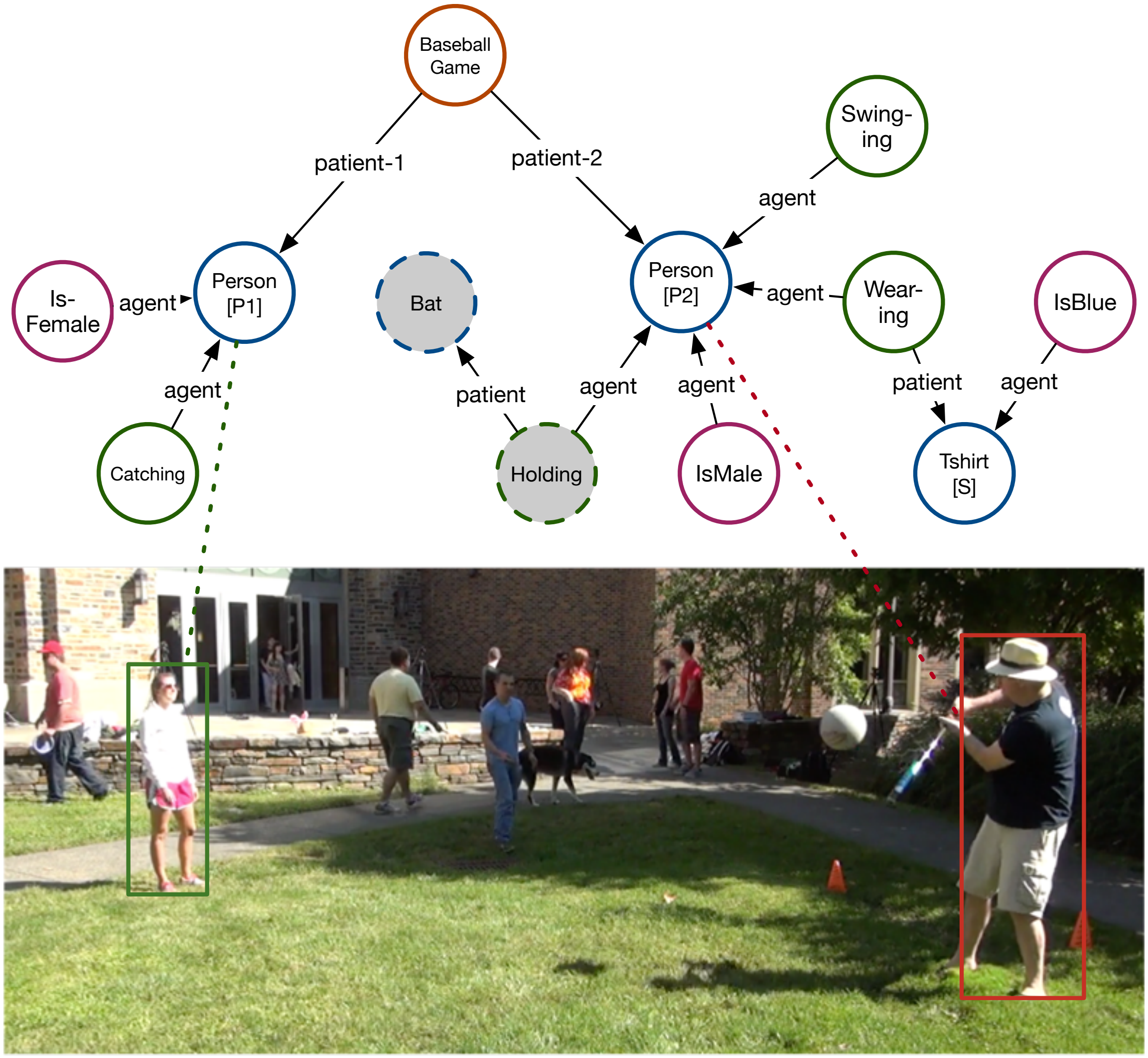


Figure 5.9: Example parse graph fragment of a baseball game that has two participants. Shaded nodes with dashed border represent concepts that are not grounded to data but can be inferred with additional clauses capturing commonsense knowledge. Dotted lines connect person nodes to the grounded bounding boxes.

evaluates to true with grounding {`x/P2, y/S, b/f(P2)`} using the modus ponens inference rule. Specifically, `x` grounds to the person `P2`, `y` grounds to the T-shirt `S`, whereas the inferred object `b` does not ground to any detected constants but can be inferred from the existence of person `P2`.

Nevertheless, the set of commonsense first-order clauses is domain-dependent and specific to applications. Discovering a set of clauses automatically from data or external sources remains an challenging task which is beyond the scope of our discussion.

5.6 Summary

In this chapter, we discussed the formalism for constructing parse graph knowledge bases. We adopted the labeled property graph model to capture the semantic structure of parse graphs and grounding details for each node. Parse graphs share the same local structure with ontology graphs. By casting questions into graph fragments of the ontology graph and answers as matched graph fragments in parse graphs, the question answering problem can be reduced to a graph matching problem. We showed a straight-forward implementation with Neo4j and Cypher. The correspondence to the first-order logic eliminates the modeling ambiguity and provides symbolic reasoning potentials.

CHAPTER 6

Conclusion

The computer vision community has been long focusing on classic tasks such as object detection, human attributes classification, action recognition. As the performance of the state-of-the-art methods are getting improved, it is increasingly important to organize the individual pieces into an integral system that can under the visual scene from a holistic joint perspective beyond the original individual tasks. In this dissertation, we explored the problem of joint visual scene parsing in a restricted visual Turing test scenario that encourages explicit concept grounding. We build a scalable and modular computer vision system that leverages pre-trained individual modules in various tasks to parse visual scenes jointly.

Firstly, we described a restricted visual Turing test scenario that evaluates computer vision systems across a wide task spectrum with a domain ontology and explicitly tests the grounding of concepts with formal queries. We presented a benchmark for evaluating long-range recognition and event reasoning in videos captured from a network of cameras via query answering. Given a set of videos of a scene and a sequence of storyline-based queries, the task is to provide answers either simply in binary form "true/false" or in natural language. Queries consist of view-centered queries which can be answered from a particular camera view and scene-centered queries which involve joint inference across different cameras. The data and queries distinguish us from visual question answering in images and video captioning as we emphasize a joint spatial, temporal, and causal understanding by utilizing scene-centered representation and storyline-based queries.

Secondly, we proposed a scalable joint parsing system that leverages off-the-shelf computer vision modules to parse scene and events in cross-view videos. The system defines a unified knowledge representation for information sharing and is extendable to new tasks and domains with its modules reconfigured. To aggregate information from multiple modules and multiple camera views, we proposed a joint parsing method 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.

Thirdly, based on the parse graph hierarchy, we discussed constructing parse graph knowledge bases and implementation details of graph-matching based query answering. We described a principled way originated from first-order logic to model concepts in a domain-specific ontology into nodes and edges in labeled property graphs. By casting the questions to graph fragments, the question answering problem is reduced to a graph pattern matching problem. Utilizing the property graph model and a declarative query language, our system efficiently stores parse graphs together with the grounding details for each node. Although our current system implements question answering as retrieval, we showed that it can be extended to reasoning with first-order resolution.

Finally, we highlight the advantages of our joint parsing system and potential future directions from three perspectives as follows.

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.

Interpretable Interface. Our joint parsing system not only aim at a comprehensive understanding of the scene, moreover, the unified parse graph representation is an interpretable interface of the intelligence agent to users. In particular, we consider the following properties an explainable interface shall have apart from the correctness of results:

- *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.

We argue that the parse graph hierarchy satisfies the four criteria above. By casting questions into graph structures and performing graph matching, the answers returned in the form of parse graphs naturally ensure its relevance to questions. In contrast to answering yes/no or providing resulting video sequences solely, the parse graphs with nodes grounded to specific data fragment serve as self-explanatory traces regarding how the answers are concluded. The answers retrieved from parse graph knowledge base are guaranteed to be consistent since the parse graph hierarchy is the single source of truth in the system and its structure is constraint by the ontology of the domain, which defines the capability of an agent explicitly.

Reasoning Potentials. We presented a retrieval-based question answering method on top of our parse graph knowledge base. The formalism of parse graph knowledge bases originating from first-order logic enables further reasoning potentials. Specifically, with additional first-order clauses that capture domain-specific commonsense knowledge incorporated, refutation theorem-proving can be implemented to infer concepts that are not originally grounded. However, discovering a proper set of commonsense knowledge clauses remains a challenging problem.

REFERENCES

[AAL15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. "VQA: Visual Question Answering." In *IEEE International Conference on Computer Vision*, 2015.

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

[AGB07] Sharon Alpert, Meirav Galun, Ronen Basri, and Achi Brandt. "Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[ARD16a] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. "Learning to compose neural networks for question answering." *arXiv preprint arXiv:1601.01705*, 2016.

[ARD16b] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. "Neural module networks." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 39–48, 2016.

[BEP08] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. "Freebase: a collaboratively created graph database for structuring human knowledge." In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250. ACM, 2008.

[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.

[BG14] Dan Brickley and R.V. Guha. "Resource description framework (RDF) schema specification.", 2014.

[BJC13] Damian Borth, Rongrong Ji, Tao Chen, Thomas Breuel, and Shih-Fu Chang. "Large-scale visual sentiment ontology and detectors using adjective noun pairs." In *Proceedings of the 21st ACM international conference on Multimedia*, pp. 223–232. ACM, 2013.

[BJJ10] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samuel White, and Tom Yeh. "VizWiz: nearly real-time answers to visual questions." In *ACM Symposium on User Interface Software and Technology*, 2010.

[BM14] Or Biran and Kathleen McKeown. "Justification narratives for individual classifications." In *IEEE International Conference on Machine Learning Workshops*, 2014.

[CLV06] Mark G Core, H Chad Lane, Michael Van Lent, Dave Gomboc, Steve Solomon, and Milton Rosenberg. “Building explainable artificial intelligence systems.” In *AAAI Conference on Artificial Intelligence*, 2006.

[CM08] Michel Chein and Marie-Laure Mugnier. *Graph-based knowledge representation: computational foundations of conceptual graphs*. Springer Science & Business Media, 2008.

[CSG13] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. “NEIL: Extracting visual knowledge from web data.” In *Proc. 14th International Conference on Computer Vision*, volume 3, 2013.

[CSP16] Haoyu Chen, Daniel Seita, Xinlei Pan, and John Canny. “An Efficient Minibatch Acceptance Test for Metropolis-Hastings.” *arXiv preprint arXiv:1610.06848*, 2016.

[CSS09] Wongun Choi, Khuram Shahid, and Silvio Savarese. “What are they doing?: Collective activity classification using spatio-temporal relationship among people.” In *IEEE International Conference on Computer Vision Workshops*, 2009.

[DBL11] Jia Deng, Alexander C. Berg, and Fei-Fei Li. “Hierarchical semantic indexing for large scale image retrieval.” In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[DDS09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “ImageNet: A large-scale hierarchical image database.” In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[Dev12] Neo4J Developers. “Neo4j.” *Graph NoSQL Database [online]*, 2012.

[DM08] Marie-Catherine De Marneffe and Christopher D Manning. “The Stanford typed dependencies representation.” In *Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation*, pp. 1–8. Association for Computational Linguistics, 2008.

[DMV00] Stefan Decker, Sergey Melnik, Frank Van Harmelen, Dieter Fensel, Michel Klein, Jeen Broekstra, Michael Erdmann, and Ian Horrocks. “The semantic web: The roles of XML and RDF.” *IEEE Internet computing*, **4**(5):63–73, 2000.

[EEV14] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. “The pascal visual object classes challenge: A retrospective.” *International Journal of Computer Vision*, **111**(1):98–136, 2014.

[EVW] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. “The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results.” http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html.

[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.

[FFP06] Li Fei-Fei, Robert Fergus, and Pietro Perona. "One-shot learning of object categories." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(4):594–611, 2006.

[FGM10] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. "Object Detection with Discriminatively Trained Part-Based Models." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(9):1627–1645, 2010.

[FHS10] Ali Farhadi, Seyyed Mohammad Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David A. Forsyth. "Every Picture Tells a Story: Generating Sentences from Images." In *European Conference on Computer Vision*, 2010.

[FP05] Li Fei-Fei and P. Perona. "A Bayesian Hierarchical Model for Learning Natural Scene Categories." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[GD05] K. Grauman and T. Darrell. "The Pyramid Match Kernel: Efficient Learning with Sets of Features." In *IEEE International Conference on Computer Vision*, 2005.

[GGH15] Donald Geman, Stuart Geman, Neil Hallonquist, and Laurent Younes. "Visual Turing test for computer vision systems." *Proceedings of the National Academy of Sciences*, **112**(12):3618–3623, 2015.

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

[GKS17] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. "Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering." In *CVPR*, volume 1, p. 9, 2017.

[Gro12] W3C OWL Working Group. "OWL 2 Web Ontology Language Document Overview (Second Edition).", 2012.

[Gru95] Thomas R Gruber. "Toward principles for the design of ontologies used for knowledge sharing?" *International journal of human-computer studies*, **43**(5-6):907–928, 1995.

[HAR16] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. "Generating visual explanations." In *European Conference on Computer Vision*, 2016.

[HLJ09] Asaad Hakeem, Mun Wai Lee, Omar Javed, and Niels Haering. "Semantic video search using natural language queries." In *Proceedings of the 17th ACM international conference on Multimedia*, pp. 605–608. ACM, 2009.

[HR12] Mohsen Hejrati and Deva Ramanan. "Analyzing 3D Objects in Cluttered Images." In *Annual Conference on Neural Information Processing Systems*, 2012.

[HS97] Sepp Hochreiter and Jurgen Schmidhuber. "Long Short-Term Memory." *Neural Computation*, **9**(8):1735–1780, 1997.

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

[HZ15] Wenfeng Hu and Shuyuan Zhu. "Learning 3D object templates by quantizing geometry and appearance spaces." 2015.

[HZR15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification." In *IEEE International Conference on Computer Vision*, 2015.

[JJM16] Allan Jabri, Armand Joulin, and Laurens van der Maaten. "Revisiting visual question answering baselines." In *European conference on computer vision*, pp. 727–739. Springer, 2016.

[JKS15] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A Shamma, Michael S Bernstein, and Li Fei-Fei. "Image retrieval using scene graphs." In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 3668–3678. IEEE, 2015.

[KPD11] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. "Baby talk: Understanding and generating simple image descriptions." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[KS16] Hema S Koppula and Ashutosh Saxena. "Anticipating human activities using object affordances for reactive robotic response." *IEEE transactions on Pattern Analysis and Machine Intelligence*, **38**(1):14–29, 2016.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks." In *NIPS*, pp. 1097–1105, 2012.

[KZG17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalanditis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. "Visual Genome: Connecting Language and Vision

Using Crowdsourced Dense Image Annotations." *International Journal on Computer Vision*, **123**(1):32–73, 2017.

[KZM12] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and Antonio Torralba. "Undoing the Damage of Dataset Bias." In *European Conference on Computer Vision*, 2012.

[LBD89] Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, R. E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. "Backpropagation Applied to Handwritten Zip Code Recognition." *Neural Computation*, **1**(4):541–551, 1989.

[LCC12] Meghann Lomas, Robert Chevalier, Ernest Vincent Cross II, Robert Christopher Garrett, John Hoare, and Michael Kopack. "Explaining robot actions." In *ACM/IEEE International Conference on Human-Robot Interaction*, 2012.

[LCV05] H Chad Lane, Mark G Core, Michael Van Lent, Steve Solomon, and Dave Gomboc. "Explainable artificial intelligence for training and tutoring." Technical report, Defense Technical Information Center, 2005.

[LMB14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft COCO: Common objects in context." In *European Conference on Computer Vision*, 2014.

[LMR15] L. Leal-Taixe, A. Milan, I. Reid, S. Roth, and K. Schindler. "MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking." *arXiv preprint arXiv:1504.01942*, 2015.

[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] Hugo Liu and Push Singh. "ConceptNeta practical commonsense reasoning toolkit." *BT technology journal*, **22**(4):211–226, 2004.

[LS10] Joerg Liebelt and Cordelia Schmid. "Multi-view object class detection with a 3D geometric model." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[LSP06] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories." 2006.

[LZZ14] Xiaobai Liu, Yibiao Zhao, and Song-Chun Zhu. "Single-view 3D scene parsing by attributed grammar." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[McB02] Brian McBride. "Jena: A semantic web toolkit." *IEEE Internet computing*, **6**(6):55–59, 2002.

[MF14a] Mateusz Malinowski and Mario Fritz. “A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input.” In *Annual Conference on Neural Information Processing Systems*, 2014.

[MF14b] Mateusz Malinowski and Mario Fritz. “Towards a Visual Turing Challenge.” *arXiv preprint arXiv:1410.8027*, 2014.

[MRR53] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. “Equation of State Calculations by Fast Computing Machines.” *Journal of Chemical Physics*, **21(6)**:1087–1092, 1953.

[MXY15] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. “Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN).” *International Conference on Learning Representations*, 2015.

[PBH13] L.D. 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.

[PNZ16] Seyoung Park, Bruce Xiaohan Nie, and Song-Chun Zhu. “Attribute And-Or Grammar for Joint Parsing of Human Attributes, Part and Pose.” *arXiv preprint arXiv:1605.02112*, 2016.

[PR12] Hamed Pirsiavash and Deva Ramanan. “Detecting activities of daily living in first-person camera views.” In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[PRF11] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. “Globally-optimal greedy algorithms for tracking a variable number of objects.” In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1201–1208, 2011.

[PZ15] Seyoung Park and Song-Chun Zhu. “Attributed Grammars for Joint Estimation of Human Attributes, Part and Pose.” *IEEE International Conference on Computer vision*, 2015.

[QWL15] Hang Qi, Tianfu Wu, Mun-Wai Lee, and Song-Chun Zhu. “A Restricted Visual Turing Test for Deep Scene and Event Understanding.” *arXiv preprint arXiv:1512.01715*, 2015.

[RA11] MS Ryoo and JK Aggarwal. “Stochastic representation and recognition of high-level group activities.” *International Journal of Computer Vision*, **93**(2):183–200, 2011.

[RDS15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. “ImageNet Large Scale Visual Recognition Challenge.” *International Journal of Computer Vision*, pp. 1–42, 2015.

[RHG15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." In *Annual Conference on Neural Information Processing Systems*, 2015.

[RKZ15] Mengye Ren, Ryan Kiros, and Richard Zemel. "Exploring Models and Data for Image Question Answering." *Annual Conference on Neural Information Processing Systems*, 2015.

[RPZ13] Brandon Rothrock, Seyoung Park, and Song-Chun Zhu. "Integrating Grammar and Segmentation for Human Pose Estimation." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[RQT13] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. "Translating Video Content to Natural Language Descriptions." In *IEEE International Conference on Computer Vision*, pp. 433–440, 2013.

[RRT15] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. "A Dataset for Movie Description." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[RRW13] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. "Grounding action descriptions in videos." *Transactions of the Association for Computational Linguistics*, **1**:25–36, 2013.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos 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. ACM, 2016.

[SAK07] Mark Slee, Aditya Agarwal, and Marc Kwiatkowski. "Thrift: Scalable cross-language services implementation." *Facebook White Paper*, **5**(8), 2007.

[SH13] Robert Speer and Catherine Havasi. "ConceptNet 5: A large semantic network for relational knowledge." In *The Peoples Web Meets NLP*, pp. 161–176. Springer, 2013.

[Sow76] John F Sowa. "Conceptual graphs for a data base interface." *IBM Journal of Research and Development*, **20**(4):336–357, 1976.

[SWJ13] Xi Song, Tianfu Wu, Yunde Jia, and Song-Chun Zhu. "Discriminatively trained and-or tree models for object detection." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[SXR15] Tianmin Shu, Dan Xie, Brandon Rothrock, Sinisa Todorovic, and Song-Chun Zhu. "Joint inference of groups, events and human roles in aerial videos." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[SZS12] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. "Ucf101: A dataset of 101 human actions classes from videos in the wild." *arXiv preprint arXiv:1212.0402*, 2012.

[TBB09] Moritz Tenorth, Jan Bandouch, and Michael 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.

[TML14] Kewei Tu, Meng Meng, Mun Wai Lee, Tae Eun Choe, and Song-Chun Zhu. "Joint Video and Text Parsing for Understanding Events and Answering Queries." *IEEE MultiMedia*, **21**(2):42–70, 2014.

[Tur50] A. M. Turing. *Mind*, **59**(236):433–460, 1950.

[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.

[VC10] Roberto Vezzani and Rita Cucchiara. "Video surveillance online repository (visor): an integrated framework." *Multimedia Tools and Applications*, **50**(2):359–380, 2010.

[VFM04] Michael Van Lent, William Fisher, and Michael Mancuso. "An explainable artificial intelligence system for small-unit tactical behavior." In *National Conference on Artificial Intelligence*, 2004.

[W3Ca] W3C. "Resource Description Framework.".

[W3Cb] W3C. "SPARQL 1.1 Overview.".

[WKS11] Heng Wang, Alexander Klaser, Cordelia Schmid, and Cheng-Lin Liu. "Action recognition by dense trajectories." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[WLZ15] Tianfu Wu, Bo Li, and Song-Chun Zhu. "Learning And-Or Models to Represent Context and Occlusion for Car Detection and Viewpoint Estimation." *IEEE Trans on Pattern Analysis and Machine Intelligence*, 2015.

[WNX14] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. "Cross-view Action Modeling, Learning and Recognition." In *IEEE Confererence on Computer Vision and Pattern Recognition*, 2014.

[WZZ16] Ping Wei, Yibiao Zhao, Nanning Zheng, and Song-Chun Zhu. "Modeling 4D Human-Object Interactions for Joint Event Segmentation, Recognition, and Object Localization.", 2016.

[XLL16] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Song-Chun Zhu. "Multi-view people tracking via hierarchical trajectory composition." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[XLQ17] Yuanlu Xu, Xiaobai Liu, Lei Qin, and Song-Chun Zhu. "Multi-view people tracking via hierarchical trajectory composition." In *AAAI Conference on Artificial Intelligence*, 2017.

[XLZ13] Yuanlu Xu, Liang Lin, Wei-Shi Zheng, and Xiaobai Liu. "Human Re-identification by Matching Compositional Template with Cluster Sampling." In *IEEE International Conference on Computer Vision*, 2013.

[XMH14] Yuanlu Xu, Bingpeng Ma, Rui Huang, and Liang Lin. "Person Search in a Scene by Jointly Modeling People Commonness and Person Uniqueness." In *ACM Multimedia Conference*, 2014.

[XXZ15] Bruce Xiaohan Nie, Caiming Xiong, and Song-Chun Zhu. "Joint Action Recognition and Pose Estimation From Video." In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1293–1301, 2015.

[YNL14] Benjamin Z Yao, Bruce X Nie, Zicheng Liu, and Song-Chun Zhu. "Animated pose templates for modeling and detecting human actions." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **36**(3):436–452, 2014.

[YYL10] Benjamin Z Yao, Xiong Yang, Liang Lin, Mun Wai Lee, and Song-Chun Zhu. "I2t: Image parsing to text description." *Proceedings of the IEEE*, **98**(8):1485–1508, 2010.

[ZFF14] Yuke Zhu, Alireza Fathi, and Li Fei-Fei. "Reasoning about Object Affordances in a Knowledge Base Representation." In *European Conference on Computer Vision*, 2014.

[ZGB16] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. "Visual7W: Grounded Question Answering in Images." In *Advances of Cognitive Systems*, 2016.

[ZM06] Song-Chun Zhu and David Mumford. "A Stochastic Grammar of Images." *Foundations and Trends in Computer Graphics and Vision*, **2**(4):259–362, 2006.

[ZPR14] Ning Zhang, Manohar Paluri, Marc'Aurelio Rantazo, Trevor Darrell, and Lubomir Bourdev. "PANDA: Pose Aligned Networks for Deep Attribute Modeling." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[ZWY17] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. "Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints." In *EMNLP*, 2017.

[ZWZ15] Jun Zhu, Tianfu Wu, Song-Chun Zhu, Xiaokang Yang, and Wenjun Zhang. "A Reconfigurable Tangram Model for Scene Representation and Categorization." *IEEE Transactions on Image Processing*, **25**(1):259–362, 2015.

[ZZ11] Yibiao Zhao and Song-Chun Zhu. "Image parsing via stochastic scene grammar." *Advances in Neural Information Processing Systems*, 2011.