# Understanding Complex Human Behaviour in Images and Videos

by

Wongun Choi

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in the University of Michigan
2013

Doctoral Committee:

Assistant Professor Silvio Savarese, Chair
Assistant Professor Ryan M. Eustice
Professor Jeffrey A. Fessler
Professor Benjamin Kuipers

# ACKNOWLEDGEMENTS

I owe sincere gratitude to all the people who have made huge influence on this thesis.

Firstly and most importantly, I am grateful to my advisor, Professor Silvio Savarese, without whom this thesis would not have been possible. Silvio's knowledge in the literature, fresh ideas, astonishing intuitions and especially his passion toward computer vision research made him a great advisor. Especially, I would also like to thank him for his patience in listening to my random (sometimes ignorant) ideas and bearing with me for last 5 years. His sincere trust and support have kept me motivated during my Ph.D study.

I'm also very much grateful to Dr. Caroline Pantofaru for her insightful suggestions and technical discussions we had during my internship and afterwards. I also appreciate her for giving sincere advice and support which had a big influence on my career. She has been a great mentor.

I am deeply grateful to my wife, Yunjeen Kim, who gave me all the trust and support during my Ph.D. study. She has been a good listener, cheerful mentor, lovely entertainer, and the main source of energy during this stressful period. Thank you my dear for bearing with me during these years!

Finally, I'd like to thank all my colleagues, Sid Yingze Bao, Johnny Yu-Wei Chao, Min Sun, Byungsoo Kim, Jie Li, Jingen Liu, Liang Mei, Khuram Shahid, Changkyu Song, Ryan Tokola, Yi-Hsuan Tsai, Yu Xiang, and Zhen Zeng, for their help and insightful discussions on various research topics. I also appreciate all my friends,

Manchul Han, Jiho Yoon, Jungwoo Kim, Yuneun Lee, Dongjun Lee, Daeyon Jung, Donghwan Kim, Kihyuk Sohn, Seunghwan Lee, and many others, for making my life in Ann Arbor be happier.

# TABLE OF CONTENTS

# LIST OF FIGURES

**Figure**

# LIST OF TABLES

**Table**

# LIST OF APPENDICES

**Appendix**

# ABSTRACT

Understanding Complex Human Behaviour in Images and Videos

by

Wongun Choi

Chair: Silvio Savarese

Understanding human motions and activities in images and videos is an important problem in many application domains, including surveillance, robotics, video indexing, and sports analysis. Although much progress has been made in classifying single person's activities in simple videos, little efforts have been made toward the interpretation of behaviors of multiple people in natural videos. In this thesis, I will present my research endeavor toward the understanding of behaviors of multiple people in natural images and videos. I identify four major challenges in this problem: i) identifying individual properties of people in videos, ii) modeling and recognizing the behavior of multiple people, iii) understanding human activities in multiple levels of resolutions and iv) learning characteristic patterns of interactions between people or people and surrounding environment. I discuss how we solve these challenging problems using various computer vision and machine learning technologies. I conclude with final remarks, observations, and possible future research directions.

# CHAPTER I

# Introduction

Activities performed by humans, for the most part, have an underlying purpose that characterizes the way individuals interact with other humans (inter-human) or surrounding objects (human-object) (Fig.1.1). When people want to communicate with others, they stand or sit close to each other, looking at each other's faces, and speak. This characterizes the activity of *talking*. On the other hand, when a person plans to prepare a food, he or she may stand in front of a kitchen countertop, holding a knife at hand, and cutting up ingredients. This characterizes the activity of *cooking*. Such inter-person or person-object interactions are ubiquitously observed in various types of human activities, and thus understanding the patterns of such interactions is vital in discovering the underlying purpose of human activities in visual media. Specifically, analyzing the interactions enables the identification of activities that are inherently ambiguous when individuals are considered in isolation (I call these activities *complex*, as opposed to the *simple* activities that can be well defined by the properties of an individual in isolation, such as running or punching). Just looking at the single individuals below, it is very hard to tell whether the person is cooking or using a computer (Fig.1.1) or if they are in a queue or they are talking (Fig.1.1); they all appear to be in a similar posture and appearance. However, the contextual information from interactions can provide critical information to tell whether the

person is cooking or the people are standing in a queue.

Recognizing complex activities in visual media (images and videos) is valuable in numerous video surveillance scenarios wherein it is imperative to track individuals and to interpret and describe their behavior in various degrees of semantic resolution. Moreover, modeling complex activities plays a critical role in related research areas such as robotics and autonomous navigation, as well as in applications where the content in large video repositories must be indexed, searched and organized. Also, the tasks of sports analysis and psychological examinations can be greatly eased with the use of a visual human activity recognition system. Further, it can provide tools for analyzing and studying typical or anomalous spatial-temporal collective behaviors in biology (insects, animals) or biomedicine (cells) and help construct an ontology of human or animal complex behaviors.

Although significant improvement has been made in recognizing simple activities thanks to the advancement of local image features in computer vision (*Dollar et al.* (2005); *Laptev and Lindeberg* (2003); *Niebles et al.* (2008); *Liu et al.* (2011a)), the problem of recognizing complex activities is far from being solved due to several critical challenges. The same human activities are often revealed in a very different form (*intra-class variation*). The number of participating people varies and the appearance or posture of an individual person can greatly differ over different instances of the same activity, e.g., two people standing and talking or three people talking while sitting on chairs. Also, the variable viewpoint of the visual media often make the activity recognition even more challenging (*viewpoint variation*). Humans appearance and the configuration of people and the surrounding environment can appear in a very different way, depending on the viewpoint. Accurately estimating the underlying individual properties and the structure of interaction in different viewpoints is very challenging. Also, typically a scene contains many visual elements that are irrelevant to the activities performed by people (*scene clutter*), such as a computer

Figure 1.1: **Top two rows**: Four different activities performed by humans; talking, queuing, using computer and cooking ordered in clock-wise direction. These activities are best described by the interaction between people or person and objects. **Bottom row**: If a person is seen in isolation (i.e., without considering objects or the other persons he/she is interacting with, that is, without considering the surrounding context) an activity cannot be characterized appropriately (see the last row).

beside a person talking to others, an outlier person who is not participating in a group activity, etc. Finally, variations in image capturing processes, such as camera motion, illumination changes and so forth, make it hard to identify the properties of humans and complex activities, e.g., whether there is a person or not when the scene is dark, whether a person or the camera is moving, etc.

In order to cope with these challenges and effectively recognize complex human activities, it is important to have 1) an accurate estimation of basic individual properties to model interactions and underlying activities; 2) a proper and invariant abstract representation to encode the inter-person or human-object interactions; 3) a robust

and coherent model that allows one to reason about complex human activities at different levels of resolution, and 4) an algorithmic tool to learn important types of interaction from a set of visual data.

In this thesis, my goal is to design algorithmic tools for understanding complex human activities in images and videos. To achieve this goal, I divide the whole problem into four different parts. In chapter II, I discuss our method for estimating 3D trajectories of multiple targets from moving cameras. As the key contribution, I propose a coherent probabilistic model and algorithm that can estimate the motion of the camera and all targets jointly. Chapter III discusses the framework for recognizing collective activities based on the novel *crowd context* concept, which encodes the semantic relationship between people using the trajectories obtained by the method described in chapter II. In chapter IV, I present the unified framework for tracking multiple targets and recognizing activities. The framework seamlessly combines the target tracking and activity recognition problems presented in chapters II and III. In chapter V, I propose a new algorithmic model that can discover a discriminative set of interactions between objects, between objects and a space, and between objects and a scene from training data to aid in indoor scene understanding, given a single image. Although the model is not tested for finding the interaction between humans and objects or the surrounding environment, it can be naturally extended to learn such interactions. Finally, in chapter VI, I conclude the thesis with discussions of my observations in these lines of research and possible future research directions.

## 1.1 Themes

The following themes appear commonly throughout most chapters.

### 1.1.1 Contextual Model for Visual Recognition

In various visual recognition tasks, contextual relationships play an important role in robustly understanding the visual properties of images and videos (*Oliva and Torralba* (2007); *Hoiem et al.* (2008)). Such contextual relationships can appear in various forms in different recognition tasks. For example, knowing the geometric structure of the scene provides strong constraints on possible locations of objects in images, and activities performed by people are strongly governed by the spatial configuration of objects and people. In this thesis, I propose various mathematical models that can leverage useful contextual relationships to better understand visual properties of images and videos. In chapter II, I propose a probabilistic model that can leverage the geometric context to better estimate camera parameters and localize targets in 3D space. Also, I find that reasoning about contextual relationship among people (interactions) is highly beneficial to obtain a robust estimation of targets' trajectories. Chapter III discusses models for encoding contextual relationships among people to recognize collective activities. I also show that reasoning about the relationship among trajectories, individual activities, pairwise interactions, and collective activity provides better recognition of all than if they are investigated in isolation (chapter IV). Finally, in chapter V, I propose a coherent model that can discover the contextual relationship between objects, space and scene type to estimate the geometric and semantic properties of an image.

### 1.1.2 Joint Inference Algorithm

Many visual recognition problems require an efficient inference algorithm to find the optimal solution in a high dimensional space. Especially when we deal with the models encoding complex contextual relationships among many variables, it is critical to have an efficient algorithm to find mutually consistent representations of the scene. In this thesis, I investigate a variety of inference algorithms to solve complex

vision problems in an efficient way. In chapter II, I propose a Markov Chain Monte Carlo (MCMC) particle filtering algorithm to find multiple targets' trajectories in 3D space, as well as camera parameters in a joint fashion. I show that the algorithm is capable of efficiently generating a robust and consistent estimation of all variables. A novel iterative algorithm is proposed to solve the joint target tracking and recognizing activities in multiple levels of hierarchy (chapter IV). The problem is divided into two parts to leverage efficient existing algorithms: belief propagation (*Felzenszwalb and Huttenlocher* (2006)) for joint activity classification and branch and bound method for trajectory estimation. In chapter V, I propose a compositional algorithm based on MCMC sampling to find the optimal configuration of a scene, that is known to be NP-hard.

### 1.1.3 Learning the Model and Associated Model Parameters

Recent computer vision algorithms enjoyed much success in complex visual recognition tasks thanks to the improvements in machine learning and statistics, especially discriminative learning methods. Thus, learning good discriminative models encoding characteristic patterns from visual data became a primal interest in computer vision research. In this thesis, I investigate different discriminative learning algorithms to learn a robust model for visual recognition. In order to learn characteristic patterns of crowd context (chapter III), I propose using Support Vector Machine (SVM) for learning parameters associated with a rigid descriptor and Random Forest (RF) to learn both structure and parameters of the crowd context. I found that RF-based learning algorithm provides better discrimination power, as it is able to learn the optimal structure of the crowd context. In more complex problems (chapters IV and V), I investigate the use of Structural Support Vector Machine (SSVM) to examine the structured relationship among different variables. As a critical contribution, I propose a novel learning algorithm that can discover a useful set of 3D geometric

phrases (3DGPs) from a dataset in an unsupervised way. The algorithm finds an initial set of 3DGPs performing data mining and obtains a refined set of 3DGPs by iterating over pruning step and parameter learning step using SSVM.

### 1.1.4 Dataset collection and Experimental Evaluation

Having a good dataset is critical for both learning the underlying model (parameters and structure) and evaluating the accuracy of algorithms in comparison to the other state-of-the-art methods. As a critical contribution, I provide new datasets for multiple target tracking in chapter II, collective activity recognition in chapter III, and indoor scene understanding in chapter V. The datasets contain large amounts of videos and images that can be used as standard benchmarks in related research problems. In order to evaluate the performance of proposed algorithms using the datasets, I incorporate various evaluation metrics. A confusion table and mean classification accuracy are offered for the evaluation of various classification tasks (chapters III, IV, and V). I analyze detection accuracy measures such as precision vs. recall and false-positive-per-image vs. miss rate curve (chapters II and V).

## 1.2 Tracking multiple targets

The trajectories of individual people provide cues as to their location and motion in the scene that are critical for understanding interactions and relationships among people. These interactions are best characterized by spatial relationships in 3D space. Thus, a robust algorithm that can estimate the 3D location and motion of humans is vital to the successful recognition of human interactions (chapter III). To that end, the first part of my thesis focuses on designing a robust tracking algorithm that is able to track multiple people in 3D space.

In previous approaches, camera parameters are first estimated using 3D reconstruction techniques such as Structure from Motion (SfM) or Simultaneous Localiza-

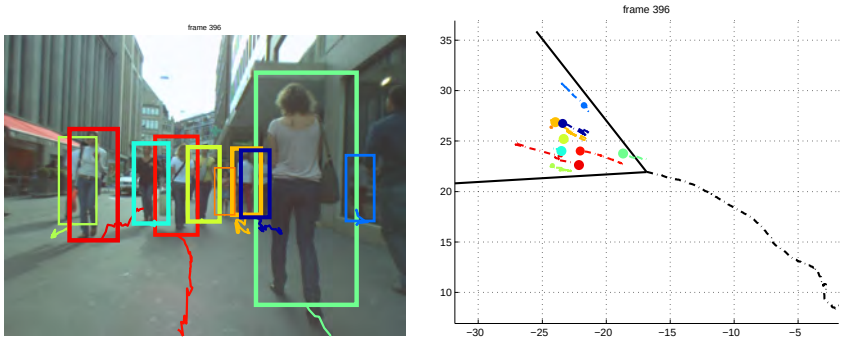



Figure 1.2: Tracking multiple targets with a moving platform is an important problem in many applications, such as activity recognition, robotics, and autonomous vehicle, wherein understanding individual motion is critical. In chapter II, I introduce our novel algorithm that is capable of tracking multiple people in 3D space with a single moving camera. The left and right figures show the estimated trajectories of individuals (colored bounding boxes and lines) as well as camera motion (visible cone and dotted line) in image and 3D space (top view), respectively. Different colors represent different identities of people.

tion and Mapping (SLAM) and the trajectories of individuals are estimated using these camera parameters. Such 3D reconstruction techniques tend to fail when there is only a small baseline change in the camera motion and are often computationally expensive. To counter these problems, I propose a novel model that encodes the motion of individuals and the camera in a coherent framework and estimates all of the unknown parameters (camera motion, trajectories, etc) in a joint fashion. Not only does this model allow us to efficiently estimate camera parameters and individual trajectories in 3D, but it also provides a generalized framework that can yield more accurate trajectory information using various interaction models, such as group and repulsion interactions between people. The optimal solution for the proposed model is found by using a Reversible Jump Markov Chain Monte Carlo method that can efficiently discover unknown numbers of people and their associated 3D trajectories. We show that the proposed algorithm can produce reliable and robust camera trajectories, as well as trajectories of individuals in 3D space in two challenging scenarios. These include a mobile system equipped with a monocular camera moving in

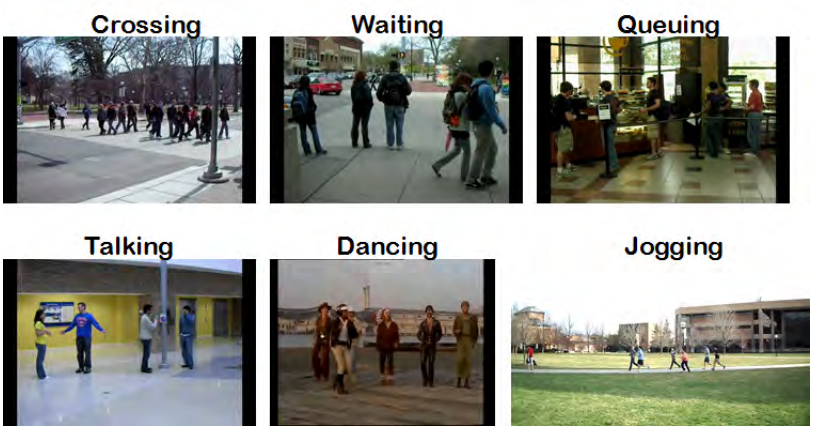



Figure 1.3: Frames extracted from the dataset on collective activities introduced in chapter III. I propose a new concept *crowd context* to recognize collective activities that include crossing, waiting, queuing, talking, dancing, and jogging.

a crowded outdoor environment and a robot agent equipped with a Microsoft Kinect (RGB and depth image sensor) exploring an indoor environment.

## 1.3 Learning the crowd context

I propose a new aspect of human activity recognition problem that I call *collective activity* recognition: collective activities are defined or reinforced by the coherent behavior of multiple people. These activities include "queuing in a line" or "talking" (see Fig.1.3 for examples of collective activities). I focus on designing a mathematical representation that can be used to robustly recognize these activities. Since collective activities are defined inherently by the coherent behavior among a number of people, it is often the case that the actions of participating individuals are interdependent and some coherency between these actions may exist. I propose a new concept (*crowd context*) that encodes such cohesive behavior in a mathematical representation and I introduce two different ways to capture the crowd context. In the first approach, I tackle this problem by introducing a descriptor that can encode the spatio-temporal relationships between people. To the best of my knowledge, this is the first attempt

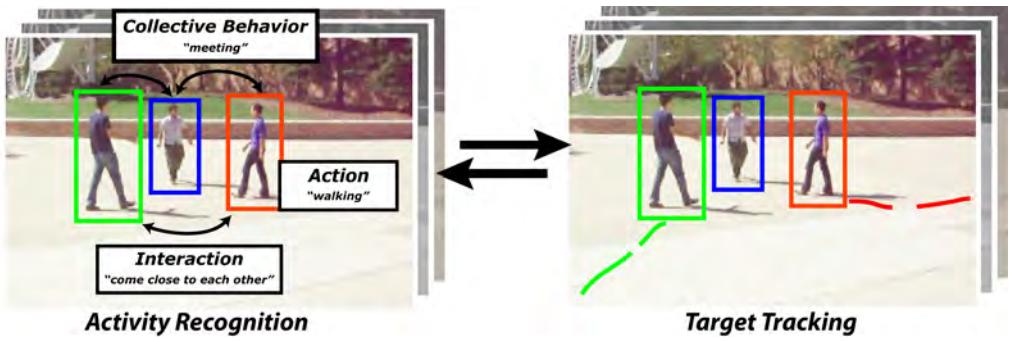



Figure 1.4: In chapter IV, I introduce a new algorithmic model that can reason about target trajectories and activities of people in multiple level of resolutions in a joint fashion. The relationship between variables characterizing target trajectories as well as activities at different levels (atomic activities, pair-wise interactions and collective activities) is encoded into a coherent energy-based model. I show that estimating the variables in a joint fashion is critical to a better understanding of activities as well as individual trajectories, rather than estimating them in isolation.

to capture the collective behavior of people using a descriptor that encapsulates the contextual relationships of individuals in time and space. I evaluate the effectiveness of the approach using a newly proposed dataset that includes video sequences of people walking, crossing, talking, standing, queuing, dancing and jogging. The dataset is now widely used by the vision community as a standard benchmark. While promising, I observe that this approach has several limitations: it requires much engineering to tune the necessary parameters specifying the structure of the descriptor and would not scale up well to a large number of activities. In order to alleviate these limitations, I propose a new method based on the Random Forest framework to automatically learn the most discriminative structure of descriptors from data. In this extension, I show that the newer method can learn meaningful structures of descriptors that capture characteristic patterns of each collective activity and thus can enhance its classification accuracy.

## 1.4 Unified model for tracking and activity recognition

In chapter IV, I explore the idea that contextual information provided by the collective behavior of multiple interacting individuals can make the tracking and activity recognition process more accurate and robust than if these problems are solved in isolation. It is clear that having accurate trajectories can help improve the understanding of high level activities, but how would we use general types of high level activity information to reinforce trajectory estimation? Recent studies by *Pellegrini et al.* (2009); *Leal-Taixe et al.* (2011) showed that the social force model - "a measure for the internal motivations of the individuals to perform certain actions (movements)" (*Helbing and Molnar* (1995)) - can improve the accuracy of multiple target tracking by considering interactions among people. However, the types of interactions considered in the social force model are limited to a few hand-designed interactions such as repulsion and attraction. The usage of general semantically meaningful interactions such as "facing each other" and "approaching" has been unexplored. I propose a novel hierarchical graphical model that encodes the relationships between trajectories, individual activities, pairwise interactions and collective activities into a joint framework. The model propagates information in both bottom-up (from trajectories to collective activities) and top-down (from collective activities to trajectories) fashion, allowing the different levels of the hierarchy to help each other. To the best of my knowledge, I am the first to show that general semantic activities can help trajectory estimation. I solve this challenging problem using a combination of belief propagation and a novel branch and bound algorithm equipped with linear programming. The model parameters representing the relationships between high level activities and trajectories are learned from a set of training data using a structured max-margin learning algorithm to guarantee discriminative power.

Figure 1.5: Typical examples of indoor images. The type of a scene provides a useful contextual cue to regularize presence of an object: sofas often appear in living rooms but not in bed rooms. The location of objects in a space is strongly constrained by the geometric structure of the space; e.g., a bed cannot appear inside a wall. Also, objects often co-appear in a characteristic 3D spatial configuration; e.g., side tables at the side of a bed and a coffee table between two sofas. Learning such interactions among objects, space and scene is critical to robustly understand a configuration of images.

## 1.5 Learning discriminative set of interactions

In chapter V, I propose an algorithm to tackle the problem of general scene understanding, leveraging on various types of interactions between humans, objects, scenes and spaces. Truly understanding a scene involves integrating information at multiple levels, as well as studying the interactions between scene elements (Fig.1.5). For instance, a scene-object interaction describes the way a scene type influences an objects presence and vice versa (i.e., it is highly likely to see a bed in bed rooms but not in dining rooms). An object-layout interaction describes the way the geometric scene layout (e.g., the 3D configuration of walls, floor and observer's pose) biases the placement of objects in the image and vice versa (e.g., objects cannot appear inside a wall). An object-object interaction describes the way objects and their poses affect

each other (a dining table suggests that a set of chairs will be found around it). Combining predictions at multiple levels into a global estimate can improve each individual prediction. I have shown that learning frequently co-occurring spatial configurations of objects (e.g., table and chairs), which we call *3D geometric phrases*, can improve the detection of each individual object and in turn lead to better scene classification and layout estimation. Although experiments are not performed on human activity recognition problems, the method can be easily extended to learn characteristic interactions between humans and between humans and objects. I envision that studying interactions is critical in many visual recognition tasks, and my work provides an important foundation for automatically learning useful interactions from visual data.

# CHAPTER II

# Estimating Trajectories of Multiple People with a Moving Camera

Understanding how people move through the world is a key problem in computer vision. There is currently a wealth of video data of people available from the Internet, from indoor mobile robot platforms, and from car-mounted sensors, to name a few sources. Accurately detecting and tracking people in video can facilitate action understanding for better video retrieval. Tracking in real time from a mobile robot can form the basis for human-robot interaction and more efficient robot performance in human environments. Detecting and tracking pedestrians from a car can help people drive safely, and it will keep people safe in the presence of autonomous cars. In this chapter, we tackle the problem of detecting and tracking multiple people as seen from a moving camera. Our goal is to design an algorithm that can be adapted for the wide array of applications in which person tracking is needed.

In practice, unfortunately, person tracking is extremely difficult. Examples of the data we wish to tackle are displayed in Fig. 2.1. The first challenge evident in these images is that people's appearances vary widely, and people change their appearance in different environments, which complicates person detection. Despite excellent advances in detection (*Dalal and Triggs* (2005); *Felzenszwalb et al.* (2010)),

This chapter is based on the publications (*Choi and Savarese* (2010); *Choi et al.* (2011a, 2013b))

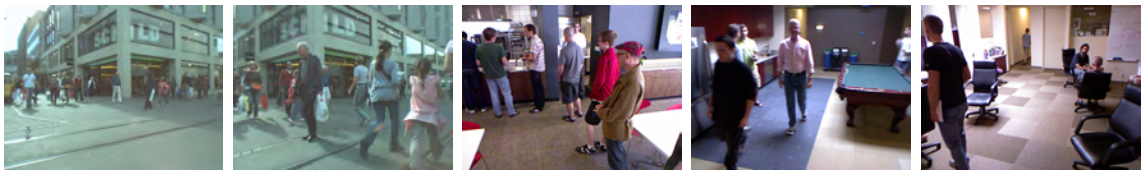


Figure 2.1: Typical examples of outdoor and indoor tracking scenarios. Correspondences between video frames are difficult to compute due to camera motion and multiple dynamic subjects. People are difficult to detect due to occlusions and limited field of view. Indoor environments are especially challenging as people tend to adopt various types of poses (standing, leaning, sitting, etc). We aim at providing a general framework for tracking multiple people in a wide variety of difficult situations.

it is still far from trivial to detect people in a variety of poses, wearing a variety of clothing, and in cluttered environments full of occlusions. A tight field-of-view that truncates people, as well as high contrast illumination, are additional difficulties often encountered in indoor environments. To improve the odds of finding people, our system combines multiple different detection cues, and allows additional cues to be added or removed as needed.

Another challenge is the complexity of the motion patterns of multiple people in the same scene. Tracking a single person is sufficiently difficult as they move wilfully and unpredictably. Tracking multiple people, however, is complicated by their interactions; assuming independence between targets' motions is insufficient. As in Fig. 2.1, people stay out of each other's personal space and never occupy exactly the same space. On the other hand, people may choose to move together as a group for awhile. To model these interactions, we propose placing constraints between the targets' motions, partially removing the independence assumption. In the chapter IV, we generalize this concept and introduce a framework where high level collective activity understanding helps estimate the interactions among individuals and, in turn, obtain more robust tracking results.

In the scenarios we wish to address, the camera is moving as well. Estimating camera motion and reconstructing a scene is a well-studied problem when the scene is

stationary (*Izadi et al.* (2011)), however the scenes described herein contain multiple large dynamic elements. Background subtraction is also likely to fail in these scenes. To tackle this issue, our tracker is capable of separating stationary and dynamic features in the scene, allowing it to estimate the camera motion and separate it from the motion of the targets.

Given that the mobile platform on which the camera is mounted needs to react to people's positions online, for example to plan to drive around them, our tracking method is capable of near real-time performance at 5-10 frames per second.

To address the issues discussed above, we propose a principled method for tracking multiple people and estimating a camera's motion simultaneously. Our contributions are as follows. First, we propose a novel model which can naturally explain the process of video generation from a single moving camera. Second, we propose a motion model that can capture the interactions between targets. Third, we introduce a principled method for fusing multiple person detection methods to build a more robust and adaptable tracker. Finally, our system is flexible enough to operate on video data alone, and to integrate depth data when it is available. This unified tracking framework is made efficient through the use of Reversible Jump - Markov Chain Monte Carlo (RJ-MCMC) particle filtering.

We demonstrate our method using the challenging ETH tracking dataset (*Ess et al.* (2008)) which contains video data taken from onboard a vehicle driving through a city, as seen in the left two images in Fig. 2.1. In addition, we contribute a dataset of color and depth image (RGB-D) data taken from onboard a robot moving in an indoor environment, as seen in the right three images in Fig. 2.1.

Throughout the chapter, we discuss our tracking framework as it applies to the task of tracking people. However, the framework is general and could be applied to other tracking tasks by replacing the detection components and changing the target motion interaction model.

## 2.1 Background

The method introduced in this chapter is designed to track multiple people in the world from a moving camera. To solve this problem a number of challenges must be overcome, including coping with the varying appearance of people as they deform over time, occlusions among people and between people and the environment, possibly missing detections, and the difficulties of estimating a moving camera's position. In this section, we discuss the related work designed to overcome one or more of these challenges.

**Tracking by Online Learning:** To track an object whose appearance is changing over time requires an adaptable object model. A number of related works seek to address this problem through online learning, learning the appearance model of a specific target and applying the model to track that target (*Comaniciu and Meer* (2002); *Avidan* (2007); *Ramanan et al.* (2007); *Bibby and Reid* (2008); *Kwak et al.* (2011)). For example, *Comaniciu and Meer* (2002) used color histograms created from user-initialized bounding boxes, and tracked those models with the mean-shift algorithm. *Avidan* (2007) showed promising results on tracking a single target using a boosting-based learning framework. A common issue for these methods is tracker drift. In addition, they all require that a target's initial position be provided manually.

**Human Detection:** One solution for improving tracker drift and enabling automatic track initialization is the use of person detectors. Over the last decade, algorithms for detecting humans have improved a great deal (*Viola et al.* (2003); *Leibe et al.* (2004); *Dalal and Triggs* (2005); *Tuzel et al.* (2007); *Wu and Nevatia* (2007); *Ferrari et al.* (2008); *Felzenszwalb et al.* (2010)). Modern human detection methods are quite reliable when applied to large pedestrians in simple scenes that include minimal crowding, occlusion and clutter. Methods by *Ferrari et al.* (2008) and *Felzenszwalb et al.* (2010) are also able to detect humans in non-pedestrian poses with reason-

able accuracy. However, in real-world environments which are crowded, include large amounts of occlusion and clutter, as well as wide pose variation, none of these methods is satisfactory. For this reason, we combine a number of person detection cues into our system.

**Tracking-by-detection:** Thanks to the improvement in human detection methods, the tracking problem can be reformulated as a tracking-by-detection problem such as in *Wu and Nevatia* (2007); *Breitenstein et al.* (2009); *Khan et al.* (2005); *Andriluka et al.* (2008); *Wojek et al.* (2009); *Choi and Savarese* (2010). This approach can generate reliable tracking results if the camera is kept stationary. Multi-target tracking problems can either be formulated to estimate target locations online, such as in the works of *Wu and Nevatia* (2007) and *Breitenstein et al.* (2009), or to find the globally optimal association among detections at different time stamps, such as is done by *Pirsiavash et al.* (2011), *Zhang et al.* (2008) and *Shitrit et al.* (2011) using a linear programming framework. However, most of the methods which do not explicitly consider camera motion are prone to failure when a camera moves since the camera motion and target motions become intertwined.

**Tracking with a Moving Camera:** To address the challenges of tracking from a moving platform, several approaches (*Wojek et al.* (2009, 2011); *Ess et al.* (2008, 2009)) have recently been proposed. *Wojek et al.* (2009, 2011) proposed a probabilistic framework to detect multiple people in a busy scene by combining multiple detectors and explicitly reasoning about occlusions among people. However, they did not associate detections between frames, so no tracking was performed. In addition, they relied on odometry readings from the car on which the camera was mounted to obtain the camera position. Our work performs data association to track people and is capable of estimating the camera motion. Our work is most similar in spirit to the work by *Ess et al.* (2008, 2009), which combines multiple detectors to estimate camera odometry and track multiple people at once. Unlike *Ess et al.* (2008, 2009), we

track targets and estimate camera motion in a unified framework and do not require stereo information.

## 2.2 System Overview

A pictorial overview of the problem is presented in Fig. 2.2. Given a stream of sensor inputs (i.e. images) $\{I_t\}$, the high-level goal of our system is to determine people's trajectories, $\{Z_t^i\}$, in a world coordinate system, while simultaneously estimating the camera's motion, $\Theta_t$. To stabilize the camera's parameter estimation, a number of features which are hypothesized to be stationary, $\{G_t^j\}$, are extracted from the scene.

A system diagram is presented in Fig. 2.3. The core of the system is the RJ-MCMC particle filter tracker, which generates proposals for subjects' track states and the camera state, and evaluates proposals given both observations from the scene and a motion model.

There are three key ingredients in making such a system perform well for person tracking. The first is the observation model and cues that are used which must

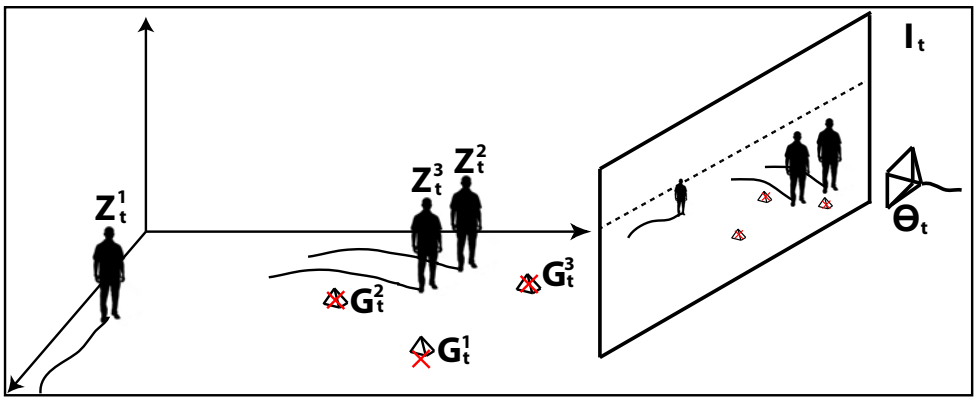



Figure 2.2: Given sensor inputs (i.e. images) $I_t$ over time, the goals of this work are 1) to track targets to obtain their trajectories, $\{Z_t^i\}$, in a world coordinate system, and 2) to estimate the camera's motion, $\Theta_t$. Stationary geometric features from the scene, $\{G_t^j\}$, are used to guide the camera parameter estimation.

account for the large variation in both people's appearance and scene statistics. The second is the motion model which must account both for people's unexpected motions as well as interactions between people. The third is the sampling procedure for the RJ-MCMC tracker, which must efficiently sample the space of possible trajectories while also accounting for people's erratic movements.

In the following sections, we will first describe the mathematical model for our system, and then describe each of the system components in detail.

## 2.3 Model Representation

We model the tracking problem using a sequential Bayesian framework, which seamlessly integrates both the estimation of camera motion and multiple target tracking. The camera parameters $\Theta_t$, a set of targets' states $Z_t$ and a set of geometric features' states $G_t$ in each time frame are modeled as random variables and the relationships among them are encoded by a joint posterior probability. With this model, the tracking and camera estimation problem is formulated as finding the maximum-a-posteri (MAP) solution of the joint probability. In this section, we explain our probabilistic formulation using the notation summarized in Table 2.1. Here, we keep the mathematical formulation general and explain the details in subsequent sections.

A configuration of all the variables at time $t$ is represented by $\Omega_t = \{\Theta_t, Z_t, G_t\}$. To find the most probable configuration, we estimate the MAP solution of $P(\Omega_t|I_{0,\ldots,t})$, which can be factored as:

$$P(\Omega_t|I_{0,\ldots,t}) \propto \underbrace{P(I_t|\Omega_t)}_{(a)} \int \underbrace{P(\Omega_t|\Omega_{t-1})}_{(b)} \underbrace{P(\Omega_{t-1}|I_{0,\ldots,t-1})}_{(c)} d\Omega_{t-1} \tag{2.1}$$

The first term (Eq. 2.1(a)) represents the **observation likelihood** of the model configuration at time $t$, $\Omega_t$, given the sensor input at time $t$, $I_t$. This measures

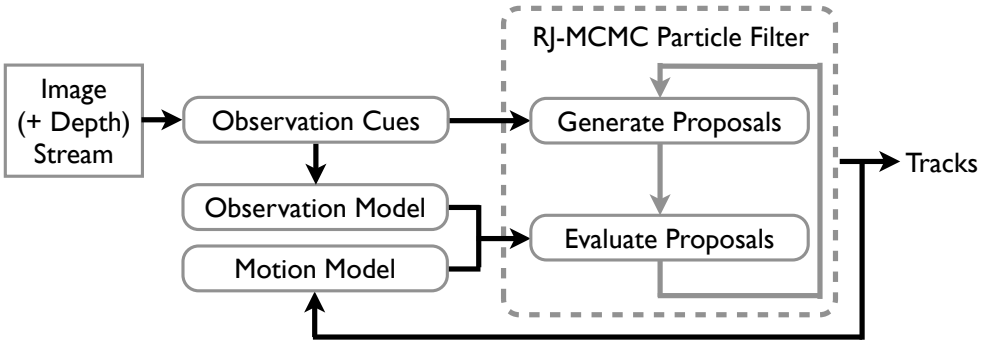



Figure 2.3: System overview. Given an input sequence, our system outputs trajectories for both targets and the camera's motion in 3D world coordinates. By employing a diverse set of observation cues to generate detection hypotheses and an observation model, our system adapts to various scenarios in a principled manner. RJ-MCMC Particle Filtering efficiently samples the set of possible trajectories, allowing for online use. Note that the system is versatile enough to cope with either monocular camera data or RGB-D sensor data.

the compatibility of a hypothetical configuration with the sensor input. The second term (Eq. 2.1(b)) is the **motion model**, which captures both smoothness of the trajectory over time, as well as the target interactions. The third term (Eq.2.1(c)) is the posterior probability at time $t-1$.

By assuming that the posterior probability at the initial time is available, the posterior probability at an arbitrary time $t$ can be calculated from the posterior probabilities from time 1 to $t-1$ sequentially. The best model configuration $\Omega_t$ is then the MAP solution.

One important characteristic of this model is that it allows the number of targets and features to vary. Tracks can be initiated when people enter the scene, and terminated automatically when people leave. The entrance or exit of a person $i$ either introduces or removes a variable into the target set $Z_t$. Similarly, a static geometric feature $j$ can go out of the scene when the camera moves or if it is occluded by a dynamic element. This also enables the model to decide which target or feature hypothesis is actually valid as the introduction of a false detection hypothesis will result in a lower joint probability. Despite this changing dimensionality of $\Omega_t$, we can esti-

mate the posterior using Reversible Jump Markov Chain Monte Carlo (RJ-MCMC) particle filtering (*Khan et al.* (2005)).

## 2.4 Observation Likelihood

The observation likelihood is a measure for evaluating which configuration $\Omega_t$ best matches the input data $I_t$. Given a hypothesis for the configuration, $\hat{\Omega}_t$, evaluation is broken into two steps: 1) project each hypothesized target and geometric feature into the input space, and then 2) evaluate the observation likelihood given the input data as in Eq. 2.2. Our input data is an image, so step 1 is equivalent to using the camera projection function $f_\Theta$ with the estimated camera parameters, $\hat{\Theta}_t$. This method can be generalized to other input modalities (e.g. lidar).

$$P(I_t|\Omega_t) = \prod_i P(I_t|Z_t^i,\Theta_t) \prod_j P(I_t|G_t^j,\Theta_t) \tag{2.2}$$

$$\underbrace{P(I_t|Z_t^i,\Theta_t) = P(I_t|f_{\Theta_t}(Z_t^i))}_{target\ observation},\ \underbrace{P(I_t|G_t^j,\Theta_t) = P(I_t|f_{\Theta_t}(G_t^j))}_{feature\ observation} \tag{2.3}$$

### 2.4.1 Camera model

We consider two different types of camera projection functions: a simplified camera projection function (*Hoiem et al.* (2008)) and a general pinhole camera projection

| | |
|---|---|
| $\Theta_t$ | camera parameters at time $t$ |
| $Z_t^i$ | a target's state at time $t$ (location and velocity in 3D). |
| $G_t^j$ | a geometric feature's state at time $t$ (location in 3D). |
| $Z_t$ | $= \{Z_t^0, Z_t^1, ..., Z_t^N\}$, the set of all targets' states at time $t$ |
| $G_t$ | $= \{G_t^0, G_t^1, ..., G_t^N\}$, the set of all geometric features' states at time $t$ |
| $\Omega_t$ | $= \{\Theta_t, Z_t, G_t\}$, the set of all random variables at time $t$ |
| $I_{0...t}$ | all sensor inputs upto time $t$ |
| $f_\Theta$ | the camera projection function parameterized by $\Theta$ |

Table 2.1: Notation definitions

function (*Hartley and Zisserman* (2000)).

**Simplified camera model:** The simplified camera model (*Hoiem et al.* (2008)) assumes that all objects of interest rest on the ground plane. Given the image location of the horizon and the camera height, the model estimates objects' 3D locations from the top and bottom of their bounding boxes in the image (see *Hoiem et al.* (2008) for details).

The camera $\Theta$ is parameterized with the following variables: focal length $f$, image center $u_c$, horizon line $v_h$, yaw angle $\phi$, velocity $\mu$, camera height $h_\Theta$ and 3D location $(x_\Theta, z_\Theta)$. For parameters $\Theta$ and object location $Z$, the projection function $f_\Theta$ is defined as:

$$Z_0 = \begin{bmatrix} R(\phi) & 0 \\ 0 & 1 \end{bmatrix} Z + \begin{bmatrix} x_\Theta \\ z_\Theta \\ 0 \end{bmatrix}, \; X = f_\Theta(Z_0) = \begin{bmatrix} \frac{fx_Z}{z_Z} + u_c \\ \frac{fh_\Theta}{z_Z} + v_h \\ \frac{fh_Z}{z_Z} \end{bmatrix} \tag{2.4}$$

where $Z_0$ represents the location of a target in the current camera coordinates, and $X = (x, y, h)$ is the corresponding bounding box in the image plane with a fixed aspect ratio. The projection function for geometric features is defined similarly (and is identical to the projected location for a target's feet.)

**Pinhole camera model:** If additional 3D input is available (i.e. a depth image), we employ a pinhole camera model to obtain a more accurate camera projection. Following the general pinhole camera model, the camera parameterization includes the focal length $f$, the 3D location $(x, y, z)$ and the orientation angles $(roll, pitch, yaw)$. See (*Hartley and Zisserman* (2000)) for details.

### 2.4.2 Target Observation Likelihood

Given the projection of a target's hypothesized location into the image, the observation likelihood measures both the validity of the target, as well as the accuracy of the location. The localization is modeled directly via the observation likelihood $P(I_t|f_{\Theta_t}(Z_t^i))$.

It is more difficult for the validity measure to adjust to the possibility that the target does not actually exist at all. The measure we would like to use is the ratio of the likelihoods $P(I_t|f_{\Theta_t}(Z_t^i))/P(I_t|f_{\Theta_t}(\emptyset))$, which allows the dimensionality of the target states variable $Z_t^i$ to vary. However, since the likelihood of the empty set is ambiguous, we instead model the ratio by taking a soft max $g(\circ)$ of the hypothesis likelihood, as in Eq. 2.6. The soft max makes the measure robust to sporadic noise.

In order to accommodate the wide array of data inputs and tracking scenarios we wish to address, our system combines a number of different detectors to evaluate the observation likelihood. This is one of the key ingredients in our approach. Each single detector has its strengths and weaknesses. For example a face detector is extremely reliable when a frontal face is presented, but uninformative if a person shows his back to the camera. We propose to combine the ensemble of detectors by using a weighted combination of detection responses as in Eqs. 2.5 and 2.6. Our experimental analysis shows that this helps make our system more robust and reliable (Sec.5.5). For simplicity, we adopt the log likelihood $l_j$ instead of the likelihood $P_j$ for each detector $j$ with weight $w_j$.

$$P\left(I_t|f_{\Theta_t}(Z_t^i)\right) \propto exp\left(\sum_j w_j l_j\left(I_t|f_{\Theta_t}(Z_t^i)\right)\right) \tag{2.5}$$

$$\frac{P\left(I_t|f_{\Theta_t}(Z_t^i)\right)}{P\left(I_t|f_{\Theta_t}(\emptyset)\right)} = exp\left(\sum_j g\left(w_j l_j(I_t|f_{\Theta_t}(Z_t^i))\right)\right) \tag{2.6}$$

We combine seven detectors to generate the observation likelihood: 1) a pedestrian

detector, 2) an upper body detector, 3) a target-specific detector based on appearance model, 4) a detector based on upper-body shape from depth, 5) a face detector, 6) a skin detector, and 7) a motion detector. The model is flexible enough to allow the addition of other observation modules as necessary for other applications. A description of each observation measurement follows.

**Pedestrian and Upper Body Detectors**

The first two observation cues are based on the distribution of gradients in the image, encoded by the Histogram of Oriented Gradient detector (HOG) by *Dalal and Triggs* (2005). We incorporate two HOG detection models, an upper body detector and a full body detector as trained in *Dalal and Triggs* (2005) and *Ferrari et al.* (2008), respectively. Using both models allows us to cope with lower body occlusions, different pose configurations, as well as different resolutions of people in images.

To obtain a detection response, the HOG detector performs a dot product between the model parameter $w$ and the HOG feature $h$, and thresholds the value (above zero). Both the positive detections and confidence values are used to model the observation likelihood from the HOG detector, as inspired by *Breitenstein et al.* (2009) (see Fig.2.4). The positive detector outputs and confidence value terms in the observation likelihood are as follows:

$$l_{Det^+}(I_t|f_{\Theta_t}(Z_t^i)) = \begin{cases} N(d_t^i; f_{\Theta_t}(Z_t^i), \Sigma_d) & \text{if } d_t^i \text{ exists} \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

$$l_{Det^c}(I_t|f_{\Theta_t}(Z_t^i)) = w \cdot h(f_{\Theta_t}(Z_t^i)) \tag{2.8}$$

where $N(\circ; \mu, \Sigma)$ is a multivariate normal distribution with mean $\mu$ and covariance $\Sigma$, $d_t^i$ is the positive detector output corresponding to $Z_t^i$, $h(\circ)$ represents the HOG feature from the region $\circ$ and $w$ is the linear detector model parameter vector. This

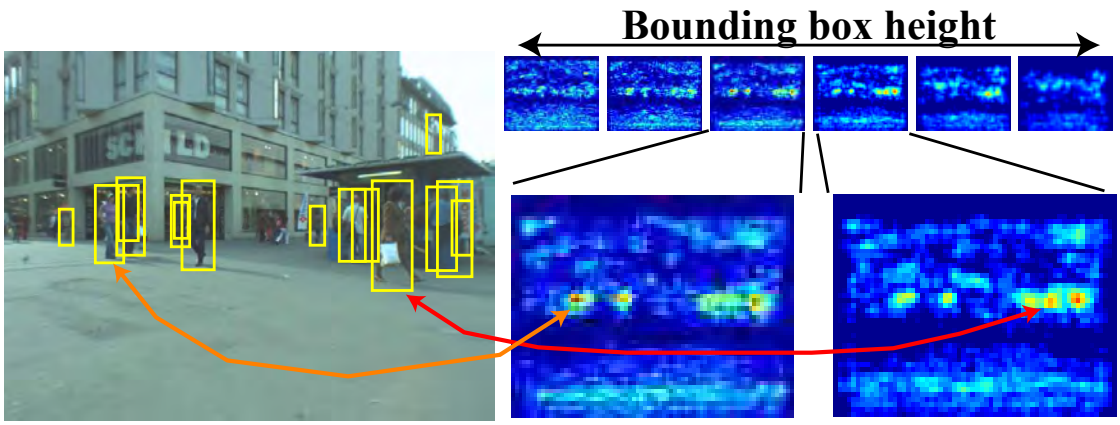



Figure 2.4: **Left:** HOG (*Dalal and Triggs* (2005)) positive detections. **Right:** the corresponding confidence maps for each candidate window size (measured by the bounding box heights). Each target projected into the image plane corresponds to one $(x, y, scale)$ position in the corresponding confidence map.

detector can be easily replaced by a more sophisticated approach as demonstrated in our experiments (we use the Deformable Parts Model (*Felzenszwalb et al.* (2010)) for the experiments on the ETH dataset).

### Face Detector

The ability to detect frontal faces (*Viola and Jones* (2003)) has proven to be useful for tracking. In our system, we employ the Viola-Jones face detector (*Viola and Jones* (2003)) as implemented in OpenCV[1]. This method detects faces reliably given a face size of greater than 24 pixels and minimal blur. The face detector likelihood is calculated as the maximum overlap ratio between all the face detection outputs $X_t^k$ and the projection of target state $Z_t^i$ into the image:

$$l_{Face} = \max_k OR(X_t^k, T_f(f_{\Theta_t}(Z_t^i))) \tag{2.9}$$

where $T_f$ is the face portion of the image projection and $OR(\cdot, \cdot)$ is the overlap ratio (intersection over union) between two rectangles.

[1] http://opencv.willowgarage.com/wiki/

### Skin Color Detector

The next cue used is skin color. If a person exists in a location $Z_t^i$, then pixels corresponding to the face region are likely to be observed even if the face is observed from the side view (face profile). To detect pixels with skin color appearance, we threshold each pixel in HSV color space and apply a median filter on the skin image $I_{Skin}$, an image of binary pixels that indicate skin region. The observation likelihood is obtained by computing the percentage of skin pixels lying in the predicted face region of a hypothesis:

$$l_{Skin} = \frac{1}{|T_f(f_{\Theta_t}(Z_t^i))|} \sum_{(x,y)\in T_f(f_{\Theta_t}(Z_t^i))} I_{Skin}(x,y) \tag{2.10}$$

where $|\cdot|$ represents the area of a bounding box and $I_{Skin}$ is the filtered binary skin image.

### Depth-based Shape Detector

Observations can also be extracted from the depth image. Each pixel in a depth image specifies the distance of the pixel from the camera in the world coordinate system. In a depth image, the head-and-shoulders outline of a person is clearly distinguishable as shown in Fig.2.5. This can be converted into the observation likelihood $l_{Shape}$ by taking the Hamming distance between a binary template of the head-and-shoulder region, with a thresholded version of the depth image projection region of $Z_t^i$, as in Fig.2.5. Then the likelihood term becomes:

$$l_{Shape}(I_t|Z_t^i) = \tau_s - d(S_{temp}, S(f_{\Theta_t}(Z_t^i); I_t)) \tag{2.11}$$

where $\tau_s$ is a threshold, $S_{temp}$ is the template, $S(f_{\Theta_t}(Z_t^i); I_t)$ is the shape vector of $Z_t^i$, and $d(\cdot,\cdot)$ is the distance between template and shape vector of $Z_t^i$.

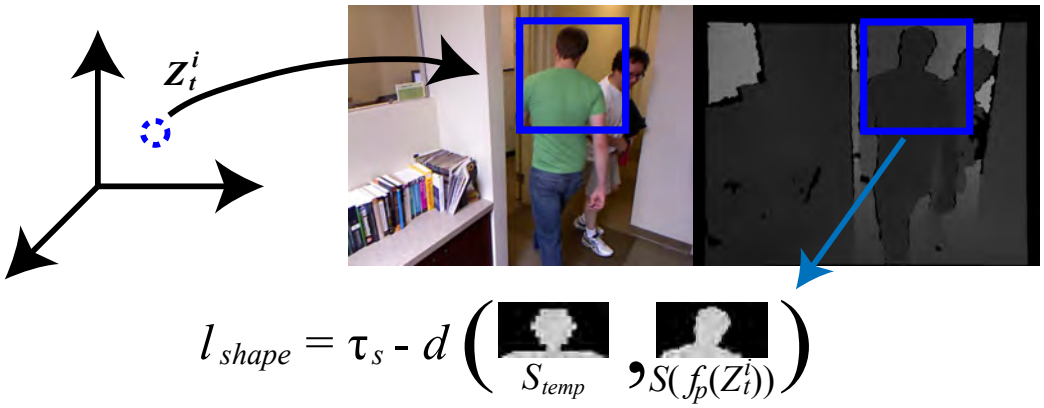



Figure 2.5: The shape vector is computed from the top half of the depth image bounding box. A binary vector of the person's head and shoulder area is compared to a template using the Hamming distance.

### Motion Detector

The presence of motion in a scene is a strong indicator of the presence of a person, especially indoors. Given depth information, motion can be efficiently identified by using a change detector in 3D. In our implementation, we use the octree-based change detection algorithm between the point clouds in consecutive frames, as described in (*Kammerl* (2011)). A binary motion image is obtained by projecting the moving points into the image plane and thresholding. The likelihood is then computed as the ratio of moving pixels lying in the body region of a hypothesis.

$$l_{Motion} = \frac{1}{|f_{\Theta_t}(Z_t^i)|} \sum_{(x,y)\in f_{\Theta_t}(Z_t^i)} I_{Motion}(x,y) \tag{2.12}$$

where $I_{Motion}$ is the binary motion image.

### Target Specific Appearance-based tracker

A detector often fails to detect the target even when it is present (false negatives). Appearance-based tracking (*Comaniciu and Meer* (2002); *Avidan* (2007); *Bibby and Reid* (2008)) can be used to help link consecutive detections. By limiting the use

of appearance-based tracking to a small number of consecutive frames, issues due to tracker drift can be minimized. We employ a color-based tracking algorithm (*Comaniciu and Meer* (2002)) to provide target-specific tracking information in each time frame. Denote the output for the tracker for target $Z^i$ at time $t$ as $Y_t^i$. Then the color-tracker observation likelihood term is:

$$l_{Tr}(I_t|f_{\Theta_t}(Z_t^i)) = N(Y_t^i; f_{\Theta_t}(Z_t^i), \Sigma_{tr}) \tag{2.13}$$

Note that for many of these cues, such as face detection, skin color detection and motion detection, a positive observation increases the likelihood that a person is present, but the lack of observation does not decrease the likelihood that a person is present.

### 2.4.3 Geometric Feature Observation Likelihood

In addition to detecting and localizing targets, we also want to compute the camera's location and orientation in the world. As in previous tracking work (i.e. the KLT tracker (*Tomasi and Kanade* (1991))), this is accomplished by detecting stationary features in the world which we call *geometric features*.

Observing geometric features can be interpreted as a generative process in which features in the world are projected onto the image plane and then detected by an interest point detector. The detection process is noisy, so the observation likelihood is modeled as a normal distribution centered on the projection of the feature, $f_{\Theta_t}(G_t^j)$. Since some of the hypothesized features may become occluded between frames, or may in fact be non-stationary features, we introduce a uniform background model for invalid features.

Let the interest point corresponding to a geometric feature $G_t^j$ be $\tau_t^i$. Then the

likelihood can be written as:

$$P(I_t|f_{\Theta_t}(G_t^j)) = \begin{cases} N(\tau_t^i; f_{\Theta_t}(G_t^j), \Sigma_G) & \text{if } G_t^j \text{ is valid} \\ K_B & \text{if } G_t^j \text{ is invalid} \end{cases} \tag{2.14}$$

Through the combination of the Gaussian component for valid features and the uniform component for invalid features, the inference process rejects outliers and estimates camera motion more robustly.

## 2.5 Motion Prior

We now discuss the motion prior term $P(\Omega_t|\Omega_{t-1})$ in Eq. 2.1. The motion model encodes smooth transitions between configurations through time via three components: 1) a camera motion prior, 2) a target motion prior and 3) a geometric feature motion prior, as follows:

$$P(\Omega_t|\Omega_{t-1}) = \underbrace{P(\Theta_t|\Theta_{t-1})}_{camera}\underbrace{P(Z_t|Z_{t-1})}_{targets}\underbrace{P(G_t|G_{t-1})}_{geom.features} \tag{2.15}$$

These three motion priors are discussed in detail below.

### 2.5.1 Camera Motion Prior

The motion of a camera over a short period of time can be assumed to be smooth both in position and rotation, and so can be modeled using a linear dynamic model with constant velocity.

For scenarios in which the simplified camera model is used, a constant perturbation model is employed for the horizon, camera height, velocity, and yaw angle, i.e. $\phi_{t+1} = \phi_t + \epsilon$ (where $\epsilon$ is an error term that accounts for uncertainty.) The location update

is:

$$x_{t+1} = x_t + v_t cos(\phi_t) + \epsilon,\ z_{t+1} = z_t + v_t sin(\phi_t) + \epsilon \tag{2.16}$$

The constant perturbation model is used with the pinhole camera model for all location-related camera parameters $(x, y, z, roll, pitch, yaw)$. The internal camera parameters (focal length, skewness, optical center, etc.) are assumed to be provided for both parameterizations.

### 2.5.2 Target Motion Prior

The motion model for the moving targets includes two factors: the existence of a target at time $t$, $P_e$, and the smoothness of its motion, $P_m$. The former encodes the probability of the person's presence at adjacent time stamps; a person is more likely to exist at time stamp $t$ if they existed at time stamp $t-1$, and vice versa. Then full target motion model is:

$$P(Z_t|Z_{t-1}) = P_{Ex}(Z_t|Z_{t-1})P_{Motion}(Z_t|Z_{t-1}) \tag{2.17}$$

In this work, we consider two possible ways to model the targets' motions: i) independent motion and ii) interactions between people affect their motion. The independence assumption has been traditionally used to simplify model inference. However, recent studies (*Scovanner and Tappen* (2009); *Pellegrini et al.* (2009); *Khan et al.* (2005)) suggest that modeling the interaction between targets enhances tracking accuracy significantly. We now describe the two terms in Eq.2.17.

#### Existence Prior ($P_{Ex}(Z_t|Z_{t-1})$)

The existence prior is modeled by two binomial probabilities, the first of which is parameterized by the probability of a target staying in the scene from one time to another ($p_s^t$). The second is parameterized by the probability of a new target entering

the scene ($p_e^t$).

$$P_{Ex}(Z_t|Z_{t-1}) = \prod_i P_{Ex}(Z_t^i|Z_{t-1}^i) \tag{2.18}$$

$$P_{Ex}(Z_t^i|Z_{t-1}^i) = \begin{cases} p_s^t & \text{if } i \text{ exists at } t-1 \text{ and } t \\ 1-p_s^t & \text{if } i \text{ exists at } t-1 \text{ but not } t \\ p_e^t & \text{if } i \text{ exists at } t \text{ but not } t-1 \\ 1-p_e^t & \text{if } i \text{ does not exist at either time} \end{cases} \tag{2.19}$$

**Independent Targets ($P_{Motion}(Z_t|Z_{t-1})$)**

The motion prior based on independent targets can be expressed as:

$$P_{Motion}(Z_t|Z_{t-1}) = \prod_i P_{Motion}(Z_t^i|Z_{t-1}^i) \tag{2.20}$$

The motion prior for a particular target, $P_{Motion}(Z_t^i|Z_{t-1}^i)$, can be modeled by a constant velocity model, giving the update rule:

$$Z_t^i = Z_{t-1}^i + \dot{Z}_{t-1}^i dt, \;\; \dot{Z}_t^i = \dot{Z}_{t-1}^i + \epsilon_Z \tag{2.21}$$

where $\epsilon_Z$ is a process noise for individual target's motion that is drawn from a normal distribution.

**Interacting Targets**

In real world crowded scenes, targets rarely move independently. Often, targets stay out of each other's personal space and they never occupy the same space. At other times, some targets may choose to move together as a group for awhile. One of the contributions of our work is to introduce such interactions into the motion model through a *repulsion model* and a *group model*.

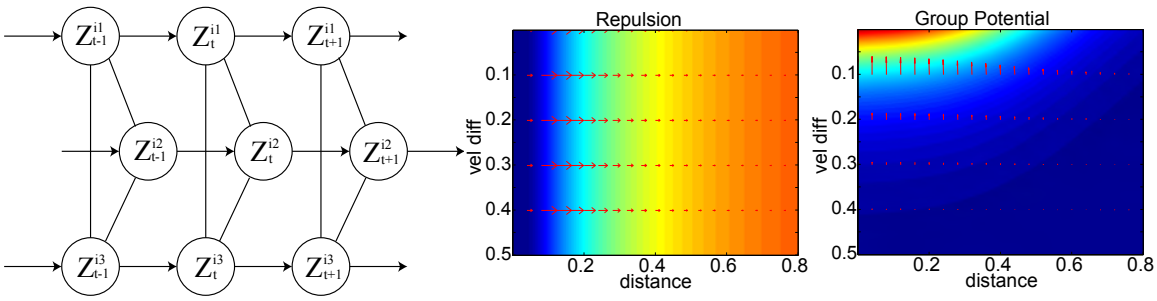



Figure 2.6: **Left:** the interactions between people are modelled by pairwise potentials, linking their positions and velocities in the motion model. **Middle and Right:** the potential functions for repulsion and group interaction, respectively, over different distances (x-axis) and velocities (y-axis) in the jet color map. Function gradients are shown as red arrows. The repulsion model pushes people apart, while the group motion enforces similar velocity. The potentials were generated with $c_r = 2, s_g = 3, t_g = 1, c_g = 10$.

We model target interactions by using an MRF as in Fig.2.6. In particular, interactions are captured by pairwise potentials between the current targets' states. Since two targets cannot both repel and form a group at the same time, a hidden mode variable $\beta_t^{i1,i2}$ selects between the interaction modes. The model then becomes:

$$P(Z_t|Z_{t-1}) = \prod_{i1<i2} \psi(Z_t^{i1}, Z_t^{i2}; \beta_t^{i1,i2}) \prod_{i1<i2} P(\beta_t^{i1,i2}|\beta_{t-1}^{i1,i2}) \prod_{i1=1}^{N} P(Z_t^{i1}|Z_{t-1}^{i1}) \tag{2.22}$$

where $\psi(Z_t^{i1}, Z_t^{i1}; \beta_t^{i1,i2})$ is the pairwise potential.

**Mode variable:** The mode variable selects the interaction type for a given pair of targets. The transition probability $P(\beta_t^{i1,i2}|\beta_{t-1}^{i1,i2})$ is modeled as $p_\beta$ if $\beta_t^{i1,i2} = \beta_{t-1}^{i1,i2}$ or $1 - p_\beta$ otherwise. The mode variable selects the interaction type such that:

$$\psi(Z_t^{i1}, Z_t^{i2}; \beta_t^{i1,i2}) = \begin{cases} \psi_g(Z_t^{i1}, Z_t^{i2}) & \text{if } \beta_t^{i1,i2} = 1 \\ \psi_r(Z_t^{i1}, Z_t^{i2}) & \text{otherwise} \end{cases} \tag{2.23}$$

**Repulsion:** The repulsion potential pushes apart targets that are too close together. Let $r_{i1,i2}$ be the distance between two targets in 3D, and let $c_r$ control the repulsion force. Then the repulsion potential is:

$$\psi_r(Z_t^{i1}, Z_t^{i2}) = e^{-\frac{1}{c_r r_{i1,i2}}} \tag{2.24}$$

The repulsion between two targets is higher as they get close to each other, and approaches 0 when they are far apart. Two targets will push each other away unless they have a group relationship and $\beta_t^{ij} = 1$.

**Group Motion:** Targets can also interact as a group, remaining at the same small distance from each other and moving in the same direction, $\dot{Z}_t^{i1} \approx \dot{Z}_t^{i2}$. By modelling the proximity between targets using a sigmoid function of their distance, we obtain:

$$\psi_g(Z_t^{i1}, Z_t^{i2}) = \frac{1}{1 + e^{s_g(r_{i1,i2} - t_g)}} e^{-c_g \left\| \dot{Z}_t^{i1} - \dot{Z}_t^{i2} \right\|} \tag{2.25}$$

where $c_g$ controls the velocity similarity, $s_g$ controls the sigmoid slope, and $t_g$ controls the distance.

The proposed interaction model improves localization when the reliability of the detection results is affected by noise. For example, detectors have trouble distinguishing between two people in close proximity whose bounding boxes overlap. In this case, the repulsion model will keep the hypotheses separate. On the other hand, the group interaction model provides constraints on the location of neighboring targets if at least one of the targets is confidently detected. Our model can be naturally extended to incorporate other interaction types, such as people approaching for a handshake.

### 2.5.3 Geometric Feature Motion Prior

The geometric features' motion prior captures whether the features are valid and whether their positions are consistent with those in previous times. To estimate the camera motion robustly, the inference must separate the stationary background features from dynamic ones. Let $P_{Val}$ be the validity prior and $P_{Cons}$ be the consistency prior. Then:

$$P(G_t|G_{t-1}) = P_{Val}(G_t|G_{t-1})P_{Cons}(G_t|G_{t-1}) \tag{2.26}$$

Similar to the target existence prior, the validity prior is modeled by two binomial probabilities which are parameterized by the probability of staying in the scene, $p_s^g$, and the probability of entering the scene, $p_e^g$. This encodes the intuition that a valid (stationary) geometric feature will likely remain valid in the next time stamp. The validity prior becomes:

$$P_{Val}(G_t|G_{t-1}) = \prod_j P_{Val}(G_t^j|G_{t-1}^j) \tag{2.27}$$

$$P_{Val}(G_t^j|G_{t-1}^j) = \begin{cases} p_s^g & \text{if j is valid at } t-1 \text{ and } t \\ 1-p_s^g & \text{if j is valid at } t-1 \text{ but not } t \\ p_e^g & \text{if j is valid at } t \text{ but not } t-1 \\ 1-p_e^g & \text{if j is not valid at } t \text{ and } t-1 \end{cases} \tag{2.28}$$

Since the features are defined as stationary 3D world points, a single feature's consistency prior $P_{Cons}(G_t|G_{t-1})$ can be modeled by an indicator function $I$ to enforce the stationary assumption:

$$P_{Cons}(G_t|G_{t-1}) = \prod_j I(G_t^j = G_{t-1}^j) \tag{2.29}$$

Overall, the target and geometric features motion priors ensure that all of the

configuration variables change smoothly, but can also appear, disappear and interact. This makes our model both robust and flexible.

## 2.6 Tracking with RJ-MCMC

We have thus far discussed how to evaluate proposed tracking states through the observation likelihood and the motion model, terms (a) and (b) of Eq. 2.1, and the left half of the system diagram in Fig. 2.3. We now need to explore the space of these hypotheses to find the MAP solution to the posterior distribution $P(\Omega_t|I_{1,\ldots,t})$. Unfortunately, the structure of the posterior is extremely complex because: i) both targets and geometric features may change their cardinality in time which, in turn, changes the dimensionality of $\Omega_t$, ii) $\Omega_t$ has high dimensionality and iii) the interaction model couples states together. As a result, traditional methods for obtaining MAP solutions are difficult to apply.

To efficiently explore the configuration space and obtain the MAP solution, we use the Reversible Jump Markov Chain Monte Carlo Particle filtering method (RJ-MCMC) introduced by *Khan et al.* (2005) (see the right half of the diagram in Fig. 2.3). The RJ-MCMC algorithm enables the addition and removal of targets via random jump proposal moves between dimensions. Unlike Khan et al., however, our goal is to estimate the camera motion and identify target interaction as well as track multiple moving targets, so we need to explore the combined configuration state space. To this end, an important contribution of this work is the introduction of additional jump proposal moves to the RJ-MCMC algorithm.

### 2.6.1 RJ-MCMC sampling

As in Eq. 2.1, the goal of tracking is to find the state that maximizes the posterior configuration:

$$\hat{\Omega}_t = \underset{\Omega_t}{\operatorname{argmax}} P(\Omega_t | I_{1,\ldots,t}) \tag{2.30}$$

We apply RJ-MCMC to obtain the posterior $P(\Omega_t | I_{1,\ldots,t})$. At each timestep, we approximate the posterior by a number of samples:

$$P(\Omega_t | I_{1,\ldots,t}) \approx \{\Omega_t^{(r)}\}_{r=1}^{N} \tag{2.31}$$

where $N$ is the number samples and $\Omega_t^{(r)}$ is the $r^{th}$ sample. These samples can be obtained by performing RJ-MCMC sampling on the posteriors from 1 to $t$. Given the set of samples at time $t-1$, the posterior distribution at $t$ can be approximated as:

$$P(\Omega_t | I_{1,\ldots,t}) \propto P(I_t | \Omega_t) \sum_r P(\Omega_t | \Omega_{t-1}^{(r)}) \tag{2.32}$$

In this section, we explain the details of our proposal distribution and sampling. Section 2.6.3 explains the acceptance ratio for the Metropolis-Hastings algorithm. In the remainder of this section, we assume that a weak detection hypothesis $X_t$ and the correspondences between targets and detections are available to guide the sampling. The detections are necessary to help initiate targets and bias sampling. Notice our algorithm is capable of accommodating missing detections and false positives as well.

### 2.6.2 Proposal Moves

As explained in Sec. 5.2, the configuration variable is composed of three components, $\Omega_t = \{Z_t, G_t, \Theta_t\}$. Sampling from the whole configuration variable's space results in very slow convergence to the steady state distribution due to high dimensionality. Thus, instead, we randomly choose one variable to sample at a time. More

specifically, one of targets, geometric features or camera parameters is randomly chosen and its state is randomly perturbed to propose a new sample. Following the Metropolis-Hasting rule, the proposed sample is accepted or rejected to construct the Markov Chain, $\{\Omega_t^{(0)}, \Omega_t^{(1)}, ...\Omega_t^{(N)}\}$.

Let the proposal distribution be $Q(\Omega_t', \Omega_t)$. Also, let $Q_Z$ be the target proposal that is perturbed with probability $q_z$, $Q_G$ be the geometric feature proposal that is perturbed with probability $q_g$ and $Q_\Theta$ the camera proposal which is perturbed with probability $q_\Theta$. Then:

$$Q(\Omega_t', \Omega_t) = q_Z Q_Z(\Omega_t', \Omega_t) + q_G Q_G(\Omega_t', \Omega_t) + q_\Theta Q_\Theta(\Omega_t', \Omega_t) \tag{2.33}$$

For example, assume that the geometric proposal is randomly chosen. Then, upon perturbation, the new configuration will be $\Omega_t^{(r+1)} = \{Z_t^{(r)}, G_t^{(r+1)}, \Theta_t^{(r)}\}$. Only a single geometric feature's state will be changed in $G_t^{(r+1)}$, and the remaining terms will remain unchanged.

**Target Proposal $Q_Z$**

The target proposal $Q_Z$ generates a new sample $Z_t^{(r+1)}$ from the current sample $Z_t^{(r)}$. The information contained in $Z_t^{(r)}$ includes the status of each target's presence and state, which has variable dimensionality depending on the number of targets present. Thus, the proposal distribution must allow efficient exploration of a space with varying dimensionality. This efficient exploration is accomplished through the use of *jump moves*.

We define a set of six reversible jump moves: *Stay*, *Leave*, *Add*, *Delete*, *Update* and *Interaction Flip*. Each move is designed to act as a reversible counterpart of another move in the set (this guarantees that the Markov Chain satisfies the *detailed balance* condition). For example *Stay* and *Leave* counteract each other. During exploration,

one of the six moves is chosen randomly with probabilities of $q_S$, $q_L$, $q_A$, $q_D$, $q_U$, and $q_I$, respectively. Below, we describe each jump type.

**Stay**: Let $S_t^{(r)}$ be the set of targets that existed in $Z_{t-1}$ but are not in sample $Z_t^{(r)}$. The *stay* move inserts one of these targets, $i$, into sample $Z_t^{(r+1)}$. The specific target to insert is chosen with uniform probability. Unlike *Khan et al.* (2005) (which samples from only the previous posterior $P(Z_t^i|Z_{t-1}^i)$), we sample the new target location from a mixture distribution of $P(Z_t^i|X_t^i)$ and $P(Z_t^i|Z_{t-1}^i)$, where $X_t^i$ is a corresponding detection. This makes the sampling process more robust to accommodate moving targets. If no detection is available for target $i$, the new proposal is sampled from the previous posterior distribution. The proposal is then:

$$Q_S(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|S_t^{(r)}|} Q_i(Z_t^{i(r+1)}) & \text{if } i \text{ in } S_t^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{2.34}$$

where $Q_i(Z_t^{i(r+1)})$ is equal to $P(Z_t^i|Z_{t-1}^i)$ when there is no corresponding detection, and $\frac{1}{2}[P(Z_t^i|Z_{t-1}^i) + P(Z_t^i|X_t^i)]$, otherwise.

**Leave**: If a target *Stays* in sample $Z_t^{(r)}$, the *Leave* move proposes to remove the target from the new sample $Z_t^{(r+1)}$. This is the reverse of *Stay*. Let $L_t^{(r)}$ be the set of targets that exist in $Z_t^{(r)}$ and existed in $Z_{t-1}$. From this set, a target $i$ is selected with uniform probability and removed. The proposal is then:

$$Q_L(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|L_t^{(r)}|} & \text{if } i \text{ in } L_t^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{2.35}$$

**Add**: This proposal initiates a new target from the new detections, $X_t^{new}$, which do not correspond to any existing targets. Let $A_t^{(r)} = X_t^{new} \backslash Z_t^{(r)}$ be the new detections

that are not in the current target set. From this set, one target $i$ is randomly selected with a uniform probability. The new location of target $i$, $Z_t^{i(r)}$, is proposed from the distribution $P(Z_t^{i(r)}|X_t^i)$. The corresponding proposal distribution can be written as

$$Q_A(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|A_t^{(r)}|} P(Z_t^{i(r)}|X_t^i) & \text{if } i \text{ in } A_t^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{2.36}$$

**Delete**: *Delete* is the reverse jump move of *Add*. Among new detections in the previous sample, $D_t^{(r)} = X_t^{new} \cap Z_t^{(r)}$, one target $i$ is randomly drawn with uniform probability and removed.

$$Q_D(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|D_t^{(r)}|} & \text{if } i \text{ in } D_t^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{2.37}$$

**Update**: *Update* proposes a new location for a target. From the targets in sample $Z_t^{(r)}$, a target $i$ is randomly selected and a new location is proposed from the distribution $Q(Z_t^{i(r+1)}; Z_t^{i(r)}) \sim \mathcal{N}(Z_t^{i(r)}, \Sigma_U)$. Note that one *Update* move can be "reversed" by another *Update* move. The proposal can be expressed as

$$K_t^{(r)} = L_t^{(r)} \cup D_t^{(r)}$$
$$Q_U(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|K_t^{(r)}|} Q(Z_t^{i(r+1)}; Z_t^{i(r)}) & \text{if } i \text{ in } K_t^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{2.38}$$

note that $K_t^{(r)}$ is the set of target indices that exist in the current target set $Z_t^{(r)}$.

**Interaction Flip**: The final target proposal considers pairs of targets and their interactions. *Interaction Flip* proposes an alternative interaction mode for a selected pair of targets, $\beta_t^{i1,i2}$. Among all possible pairs of targets in a sample $Z_t^{(r)}$, a pair of

targets $(i1, i2)$ is randomly selected and the mode of interaction is flipped between repulsion and group interaction, or vice versa, with probabily $p_f$.

$$
\begin{aligned}
&Q_I(Z_t^{(r+1)}; Z_t^{(r)}) \\
&= \begin{cases} \frac{2Q(\beta_t^{i1,i2,(r+1)}; \beta_t^{i1,i2,(r)})}{|K_t^{(r)}|(|K_t^{(r)}|-1)} & \text{if } i1, i2 \text{ in } K_t^{(r)},\ i1 \neq i2 \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{2.39}
$$

**Geometric Feature Proposal $Q_G$**

Similarly to the target states in $Z_t$, the geometric features' states stored in $G_t$ also need to be updated. $G_t$ is also a high dimensional vector with a variable dimensionality. Thus, we use the same scheme as that used for targets to update it. To update the geometric feature states, we use the proposal moves: *Stay*, *Leave* and *Update*. As for the target proposals, one of the proposals is randomly chosen with probability of $q_S$, $q_L$ and $q_U$, respectively. Note that since the validity of features can only be defined by comparing their location in different time frames, we do not use the *Add* and *Delete* moves in feature proposals. That is, in order to verify whether a feature is stationary or not, we assume we observe it for at least two adjacent frames. All the newly introduced features are automatically added into the feature set in the time frame, and the validity of features is examined by comparing the observed position and the predicted position using *Stay* and *Leave* moves in the subsequent frames.

**Stay**: Similarly to the *Stay* move for target proposals, the *Stay* move proposes to keep feature $j$ that was in $G_{t-1}$ but is not in $G_t^{(r)}$. With a slight abuse of notation, let $S_t^{(r)}$ be the set of features which are in $G_{t-1}$ but not in $G_t^{(r)}$ and let one of these be chosen with uniform probability. The location of the feature is drawn from $P_c(G_t^{j(r+1)}|G_{t-1}^j)$,

which gives:

$$Q_S(G_t^{(r+1)}; G_t^{(r)}) = \begin{cases} \frac{1}{|S_t^{(r)}|} P_c(G_t^{j(r+1)} | G_{t-1}^j) & \text{if } j \text{ in } S_t^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{2.40}$$

**Leave**: The *Leave* move for geometric features follows the same structure as the *Leave* move for target proposals. Let $L_t^{(r)}$ be the set of features that exist in both $G_t^{(r)}$ and $G_{t-1}$.

$$Q_L(G_t^{(r+1)}; G_t^{(r)}) = \begin{cases} \frac{1}{|L_t^{(r)}|} & \text{if } i \text{ in } L_t^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{2.41}$$

**Update**: Similarly to the target *Update* proposal, we randomly select a geometric feature and propose a new location for the feature by adding gaussian noise. Since geometric features are defined to be static, it is not necessary to explore different locations for an existing feature. The motion consistency prior for the features is defined to be an indicator function. As a result, any new state $G_t^{j(r+1)}$ that is different from $G_{t-1}^j$ will have 0 probability and, thus, perturbations are only applied to newly added features, $N_t = G_t^{(r)} \setminus G_{t-1}$.

$$Q_G(G_t^{(r+1)}; G_t^{(r)}) = \begin{cases} \frac{1}{|N_t^{(r)}|} Q(G_t^{j(r+1)}; G_t^{j(r)}) & \text{if } j \text{ in } N_t \\ 0 & \text{otherwise} \end{cases} \tag{2.42}$$

Similarly to the update proposal for targets, $Q(G_t^{j(r+1)}; G_t^{j(r)})$ is modeled by a normal distribution $\mathcal{N}(G_t^{j(r+1)}; G_t^{j(r)}, \Sigma_G)$ parameterized by $\Sigma_G$.

**Camera Proposal $Q_\Theta$**

The final component we need to sample is the camera state. Since there is only one camera, we can model the camera state proposal $Q_\Theta(\Theta_t^{(r+1)}; \Theta_t^{(r)})$ by a simple normal distribution $\mathcal{N}(\Theta_t^{(r+1)}; \Theta_t^{(r)}, \Sigma_\Theta)$.

### 2.6.3 Acceptance Ratio

Following the Metropolis Hastings algorithm, we compute the acceptance ratio of the new sample $\Omega_t^{(r+1)}$ by the product of the three ratios:

$$a = \frac{P(I_t|\Omega_t^{(r+1)})}{P(I_t|\Omega_t^{(r)})} \frac{P(\Omega_t^{(r+1)}|I_{1,2,\ldots t-1})}{P(\Omega_t^{(r)}|I_{1,2,\ldots t-1})} \frac{Q(\Omega_t^r; \Omega_t^{r+1})}{Q(\Omega_t^{r+1}; \Omega_t^r)} \tag{2.43}$$

The first term expresses the ratio between the image likelihoods; the second term is the ratio between approximated predictions; the last term encodes the ratio between proposal distributions. Since we change the state of only one target's presence or location at a time, most of the factors can be cancelled out in the above computation. This characteristic makes the algorithm efficient and capable of processing videos in real-time.

## 2.7 Experimental Evaluation

We demonstrate our proposed algorithm using two different types of data inputs and three datasets. The first dataset is a part of the ETH dataset (*Ess et al.* (2008)) that includes the sequences *ETH-Linthescher* and *ETH-Bahnhof* (seq02 and seq03 in *Ess et al.* (2008)). This data consists of video sequences recorded with a moving camera in densely populated urban streets with pedestrians. The videos have a frame rate of ∼14Hz and a resolution of 640×480 pixels.

The second and third datasets are collected using a Kinect RGB-D camera[2] and consist of video sequences associated with depth maps (RGB-D). Both of the datasets

[2]NITE natural interaction middleware, http://www.primesense.com/?p=515

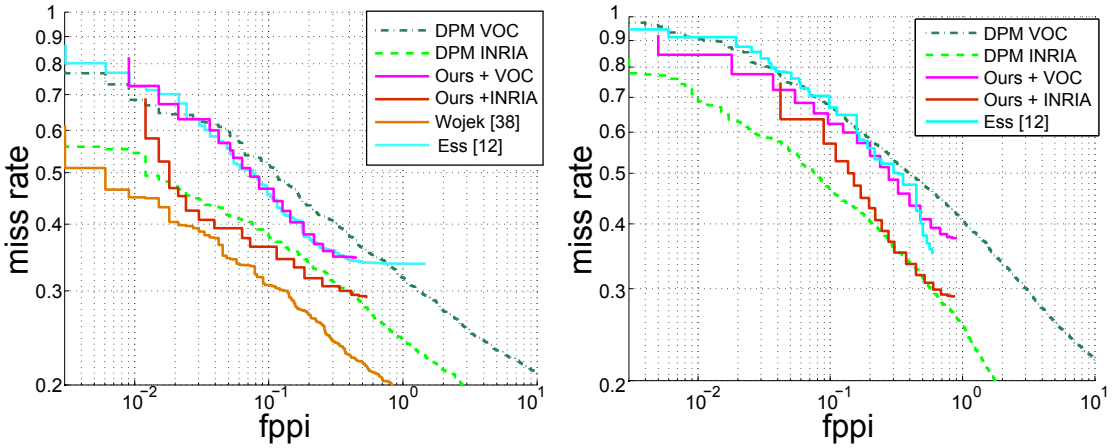



Figure 2.7: FPPI vs miss-rate curves for the sequences **Left:** *ETH-Linthescher, Seq2* and **Right:** *ETH-Bahnhof, Seq3*.

contain longer video segments and tracks than previous datasets, making data association and camera motion estimation more difficult. The first RGB-D dataset (we call it the *Kinect office dataset*) is acquired using a static Kinect mounted approximately 2 meters high (and tilted down) in an office. This set contains 17 videos, typically 2 to 3 minutes long. People in these scenes take on different poses (e.g. sitting on a chair, standing up), are observed from different view points (front, side, 3/4 rotation) and are subject to various degrees of occlusions, inter-occlusions and self-occlusions. The second RGB-D dataset (we call it the *Kinect mobile dataset*) is collected from a Kinect mounted on a mobile platform (a PR2 robot). The robot was driven (tele-operated) around an office building, while sequences of people performing daily activities in offices, corridors, hallways and cafeteria were acquired. The sequences include various configurations where the camera and targets are moving at the same time, the targets are located at different distances from the camera, the number of targets in the scene are changing over time, and targets are subject to occlusions, illumination condition varies in time, etc. This dataset includes 18 video sequences.

In both Kinect datasets, humans are hand-annotated with bounding boxes around upper bodies in each image. Targets' 3D locations are inferred from the bounding

| | | | |
|---|---|---|---|
| Motion | Targets | $p_s^t$ | 0.8 |
| | | $p_e^t$ | 0.15 |
| | Interaction | $c_r$ | 2.0 |
| | | $s_g$ | 3.0 |
| | | $c_g$ | 12.0 |
| | | $t_g$ | 1.0 |
| | Geometric Features | $p_s^g$ | 0.9 |
| | | $p_e^g$ | 0.2 |
| Sampling | Common | # Samples (N) | 5000 |
| | | # Burn-in | 1000 |
| | | # Thinning | 100 |
| Proposal | Targets | $q_S$ | 0.05 |
| | | $q_L$ | 0.05 |
| | | $q_A$ | 0.025 |
| | | $q_D$ | 0.025 |
| | | $q_U$ | 0.2 |
| | | $q_I$ | 0.1 |
| | Geometric Features | $q_L$ | 0.075 |
| | | $q_S$ | 0.075 |
| | | $q_U$ | 0.1 |
| | Camera | $q_U$ | 0.3 |

Table 2.2: Model parameters used in the experiments.

boxes and depth images (where available). The annotation is provided on four images every second. In addition, ground truth odometry information of the camera's location in 3D space is also provided for evaluation purposes. In the *Kinect mobile dataset*, the odometry is obtained via the robot localization using the ROS system[3], which utilizes multiple sensor inputs as well as a known building map.

### 2.7.1 Implementation Details

The overall system flow is as follows. Given a sensor input in each time frame, a set of weak detection hypotheses $X_t$ of human targets in the scene are generated using the observation cues. The correspondences between the predicted locations of targets, $\hat{Z}_t$, and the weak hypotheses are identified using the Hungarian algorithm (*Kuhn* (1955)).

[3]ROS, The Robot Operating System, http://www.ros.org/

Geometric features (a maximum of 40) are detected using the SURF detector (*Bay et al.* (2008)) and tracked using the KLT tracker (*Tomasi and Kanade* (1991)). For the RGB-D data, we include depth information.

The feature trajectories, sensor inputs, detection hypotheses and previous time posterior distribution are passed to the RJ-MCMC algorithm to estimate the posterior at time $t$. The MAP approximation of the new camera and targets' states is given by the mean of the posterior samples, $\{\Omega_t^{(r)}\}_{r=1}^N$. In order to improve the computational efficiency, we remove the trajectories that generate fewer than 10% of the samples.

Some important model parameters are listed in Table.2.2. For example, we draw 5000 samples in each time stamp to estimate camera parameters, targets' states and geometric features' states. With this set of parameters, the algorithm runs in near real-time (about $1 \sim 200ms$ per frame without particular optimization or parallelization). Notice that the algorithm requires more samples to be drawn as the number of targets increase. We plan to study the impact of the number of targets against the tracking accuracy as a future direction. We omit the detailed parameters for observation likelihood in the Table.2.2 for brevity. Please see the code for the complete list of numerical parameters[4].

To account for the different sensor modalities in each of the test sets, the experiments below were run with two different system setups.

**ETH datasets:** In each ETH sequence, only 2D information is used, therefore we employ the simplified version of the camera projection function (see Section2.4.1). In addition, the people are often quite small, so we cannot use depth, faces, skin detection or 3D-based motion for this data. Instead, we use the Deformable Parts Model (DPM) detector (*Felzenszwalb et al.* (2010)) and color-based meanshift tracker (*Comaniciu and Meer* (2002)) as the observation cues. The detection cues correspond to DPM detections with confidence greater than 0.5. Note that as shown in *Ess et al.* (2009),

[4]The source code and datasets are available at `http://www-personal.umich.edu/~wgchoi/pami_track/`.

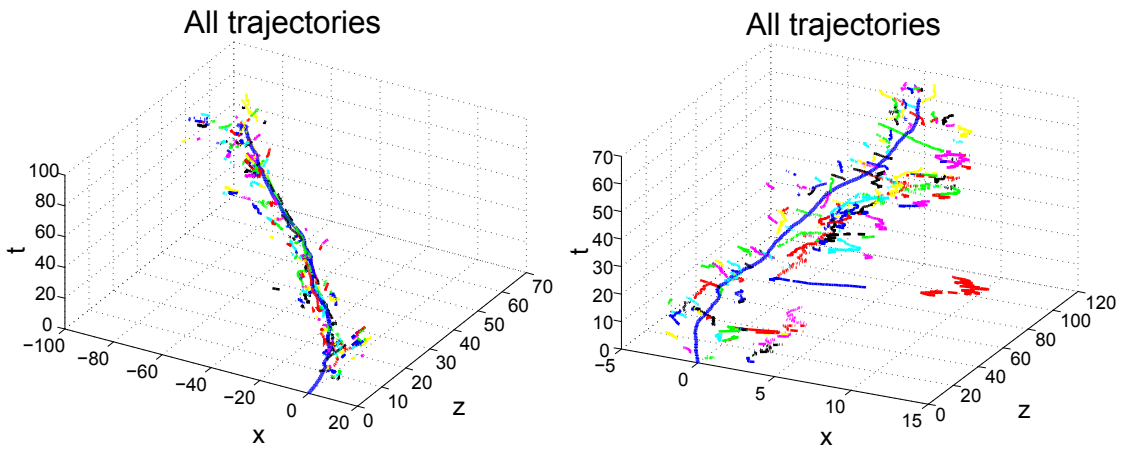



Figure 2.8: The camera trajectories (long dark-blue lines) estimated from the sequences *ETH-Linthescher* (left) and *ETH-Bahnhof* (right). Targets' trajectories (short multi-color lines) are also shown for illustration purposes.

depth may help to further improve the detection rate. All of the system parameters are held constant for all of the sequences.

**Kinect datasets:** All the observation cues described in Section 2.4 are used in these experiments. We incorporate the upper and full body HOG detectors as trained by *Ferrari et al.* (2008) and *Dalal and Triggs* (2005), and as implemented in OpenCV to run on the GPU. Although the DPM detector (*Felzenszwalb et al.* (2010)) is more accurate, the speed of the GPU-based HOG detector is required. A 640x480 pixel image can be processed in 100∼200 milliseconds. We also use the face detector implemented in OpenCV. Skin pixels are identified by thresholding HSV values between (2, 60, 40) and (15, 200, 200). Finally, the octree-based motion detector is discretized to 3cm. The weak detection hypotheses $X_t$ consist of the HOG detections (upper and full-body), face detections, as well as 3D point clusters (*Rusu and Cousins* (2011)).

### 2.7.2 Evaluation on the ETH dataset

We first study the single-frame detection accuracy on the video sequences *ETH-Linthescher* and *ETH-Bahnhof* (Fig.2.7) and compare it to the baseline methods: the DPM detector (*Felzenszwalb et al.* (2010)), the method by *Wojek et al.* (2011)

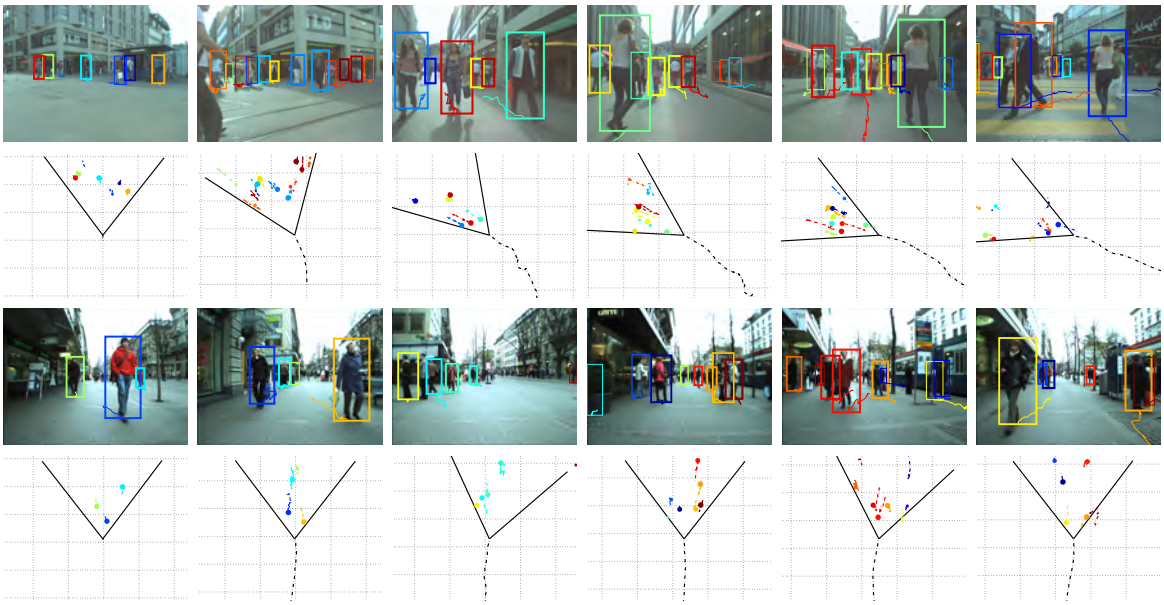


Figure 2.9: Qualitative examples of tracking and camera estimation on the ETH datasets, *ETH-Linthescher* and *ETH-Bahnhof*. Each set of tracking examples is shown in two rows: the target trajectory and detection overlaid on the image (**top**) and a top-down projection onto the ground plane (**bottom**). Each target's trajectory is shown in a distinct color. In the top-down view, the V-shaped line indicates the camera's field of view, and the tail behind the V is the camera center's location over time. Notice the long target paths which indicate stable tracking.

and the system by *Ess et al.* (2009). As a metric, we compute single-frame detection accuracy via the overlap ratios between the ground truth bounding boxes and the tracked bounding boxes. True and false positives are identified among such detections following the PASCAL challenge protocol (*Everingham et al.* (2010)). The confidence of each target is measured as the number of valid samples. Note that we use the extended annotation from *Wojek et al.* (2011) for the evaluation of *ETH-Linthescher* but we use the annotation in *Ess et al.* (2009) for the evaluation of *ETH-Bahnhof*. The extended annotation decreases the minimum person-size from 60 pixels to 48 pixels. As in *Wojek et al.* (2011); *Ess et al.* (2009), we discard detections and annotations that are smaller than 60 pixels in the evaluation.

To show the adaptability of our system, we show experiments using two different

DPM models as learned from the the INRIA (*Dalal and Triggs* (2005)) and the VOC09 (*Everingham et al.* (2010)) datasets. The results in Figure 2.7(left) show that our method (Ours+INRIA and Ours+VOC curves) improves detection accuracy over the two DPM baselines (DPM INRIA and DPM VOC), and obtains better or comparable results than the system in *Ess et al.* (2009). Note that, as observed by Ess *Ess et al.* (2009), tracking algorithms often produce inferior detection results to their baseline detector since the tracker requires multiple frames to initiate tracking and also holds on to targets a few frames after they disappear (shown as thin bounding boxes in Figure 2.9). Nevertheless, our system produces better detections than the baseline detector. *Wojek et al.* (2011) produce better detections than our system, however, they do not perform tracking, nor do they estimate the camera's trajectory.

We also show the results of camera estimation in Figure 2.8 (long dark-blue lines). The $(x, z)$ plane is defined along the camera coordinate system in the first frame of each video sequence, and the third dimension is time. Although no ground truth is available for the camera, we can qualitatively see that in the *ETH-Linthescher* sequence, the camera makes a left turn around the $150^{th}$ frame, which matches what we observe in the video sequence. Afterwards the camera moves approximately straight ahead until the end of the video. The camera motion in the *ETH-Bahnhof* sequence is correctly estimated as going roughly straight through the crowd.

### 2.7.3 Evaluation on the Kinect datasets

Next we demonstrate our method using the two Kinect datasets. As before, we begin by examining the detection accuracy. We compare our system against the DPM full body and upper body detectors as trained in *Ferrari et al.* (2008). Figure 2.10(a) shows the FPPI vs miss-rate curves for the three approaches. In the legend, we also provide the *log-average miss rate* (LAMR) proposed by *Wojek et al.* (2011). As in *Wojek et al.* (2011), the LAMR is computed by drawing equally spaced samples

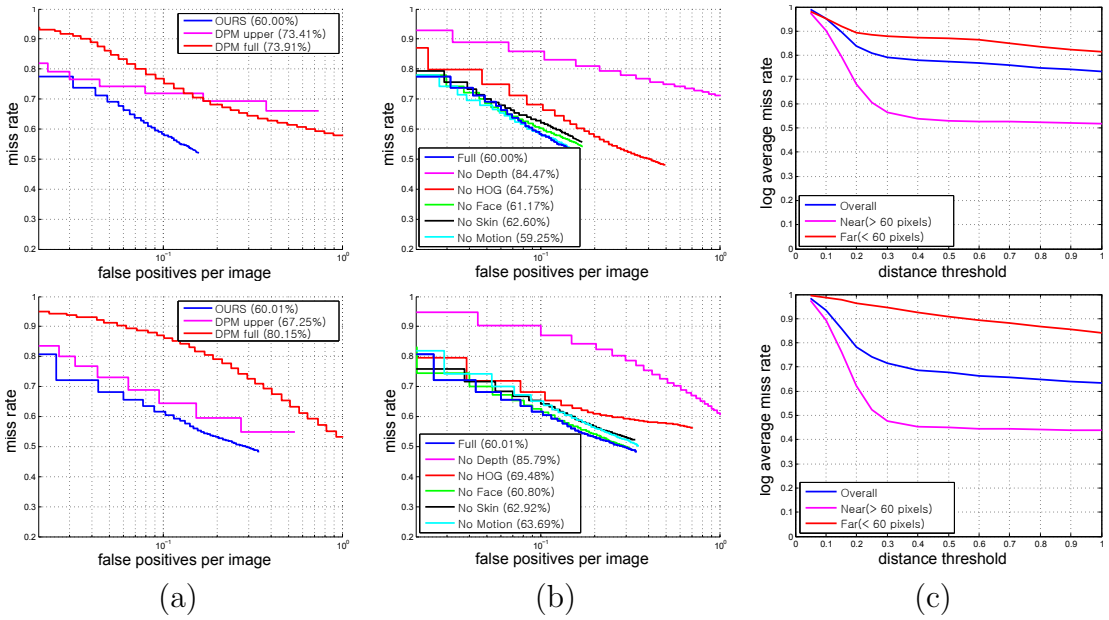



Figure 2.10: Top row: results on the *Kinect office dataset*. Bottom row: results on the *Kinect mobile dataset*. (a) Baseline comparison versus the Deformable Parts Model (DPM) (*Felzenszwalb et al.* (2010); *Ferrari et al.* (2008)). Our system outperforms both the full- and the upper-body DPMs. (b) System analysis where contributions of each observation cue are visualized with different plots. The 'Full' observation includes all components. The other curves show the results obtained by removing specific components (such as the face detector). Notice that the depth mask is the most important cue, followed by the HOG detectors. The other components' contributions are situation-dependent and on average they appear less important. (c) Log Average Miss Rate (LAMR) over different distance thresholds. Detections were considered true positives if they were within 30cm in height and the distance threshold in depth from the ground truth. Results are shown for all the data, and also broken down for two distance ranges: near (detections larger than 60 pixels in height) and far (smaller than 60 pixels).

in log space of the FPPI. We incorporate two evaluation protocols to determine a true positive. The first is based on the bounding box overlap protocol from PASCAL (*Everingham et al.* (2010)). The second is based on a 3D distance threshold for localization.

Our algorithm outperforms both baseline methods significantly; there is 13% improvement in LAMR over both baselines on the *Kinect office dataset*, and 7% over the upper body DPM detector and 20% over the full body detector on the *Kinect*

*mobile dataset*. Notice that we achieve such improvement even though we employ the weak HOG detector for detecting targets. As expected, the full body detector does not work well in the indoor scenario due to frequent occlusions, tight field of view, and unconventional poses such as sitting.

Next, we compare the contribution of each observation cue to our system. In this experiment, we turn off one detection cue at a time and compare the resulting detection accuracies in Figure 2.10(b). Turning off the depth shape detector (the No Depth curve) is the most detrimental to the system. Turning off both of the HOG detectors also results in a clear decrease in performance. Turning off the other observation cues has less obvious impact. This can be explained by the fact that the other cues are situation-dependent, and so their contribution is not evident when averaging over the dataset. For example, the face detector is a very strong and reliable cue when there is a large frontal face in the scene. However, often the person is far from or turned away from the camera, making the face detector useless, or worse, creating noise in the system. A similar argument can be made about motion detection. The fact that our system is able to perform well despite the variability of its individual components is a testament to its robustness. As future work, however, we would like to learn the situations in which to use various detectors, for example using full-body HOG detectors when moving down a hallway but not when in a cluttered room.

Finally, we evaluate our algorithm's localization accuracy. In Figure 2.10(c), we show the LAMR measure over different 3D distance thresholds. Our method is more accurate in detecting people less than approximately 5 meters from the camera than those past 5 meters. This is an expected effect since the Kinect provides virtually no depth information past 5 meters, and in fact the depth information past 3 meters is very noisy.

Overall, experiments show that our algorithm outperforms state-of-the-art detec-

tors. In addition, the fusion of multiple detection cues provides a more reliable final result and is capable of handling the variable performance of each individual detector. Selected tracking examples are shown in Figure 2.12. As shown in these results, the proposed method can reliably detect and track people in challenging indoor scenarios including occlusion between people, people in various poses, truncated body parts, and clutter.

**Camera Estimation**: Finally, we evaluate our system's ability to estimate the camera parameters. We compare our results against a baseline method which is constructed as follows. Given the feature trajectories, we compute the rotation matrix $R_t^{t-1}$ and translation vector $T_t^{t-1}$ of the camera between consecutive frames. Using the depth provided by the RGB-D sensor, we can compute $R_t^{t-1}$ and $T_t^{t-1}$ using the orthogonal Procrustes problem (*Gower and Dijksterhuis* (2004)). To cope with dynamic elements in the scene and add robustness, we add a RANSAC (*Fischler and Bolles* (1981)) step on top of the estimator.

The comparison is presented in Figure 2.11. Since our method localizes the camera online, we measure the difference between parameters in consecutive time stamps. For all pairs of time stamps $t_i$ and $t_j$ with temporal gap $t_g$ ($t_j = t_i + t_g$), we compute the transformation that maps the camera coordinate system of $t_i$ to $t_j$. Such transformations are obtained for both the ground truth and the two estimations. The error between the transformations of ground truth and each estimation is reported for different time intervals ($t_g$). Figure 2.11 shows the mean and standard deviation of the error. The amount of error tends to increase with the time span due to error accumulation in the estimation. We report the estimation accuracy for each variable: translation $(x, y, z)$ and rotation (roll, pitch, yaw). Each row in Figure 2.13 shows the estimated parameters over time for four different sequences. The ground truth is in red, the baseline in green, and our system is in blue.

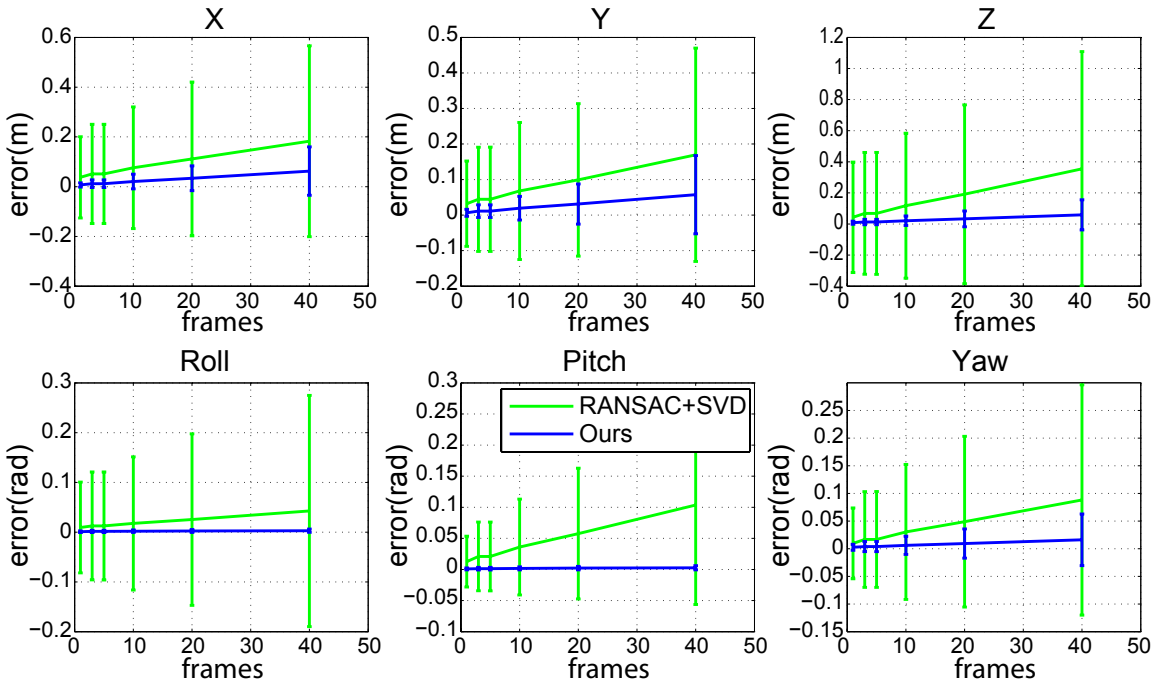



Figure 2.11: Quantitative evaluation of camera localization. The mean and standard deviation of the error in each camera parameter estimation over different time spans.

As demonstrated in these results, our method is capable of robustly estimating camera motion under difficult conditions in which the baseline method fails to localize the camera. These scenes are challenging due to 1) lack of dominant stationary scene elements, 2) lack of a motion model for the camera or targets. Our method is able to cope with such challenges by 1) jointly identifying moving targets and static features in the estimation process, 2) using high level semantics (targets) as well as local features, and 3) incorporating the camera's motion prior. We observe that our method can localize the camera very accurately except for few very hard cases; e.g. the camera was facing a featureless wall around the 1000th frame of the 4th example in Figure 2.13.

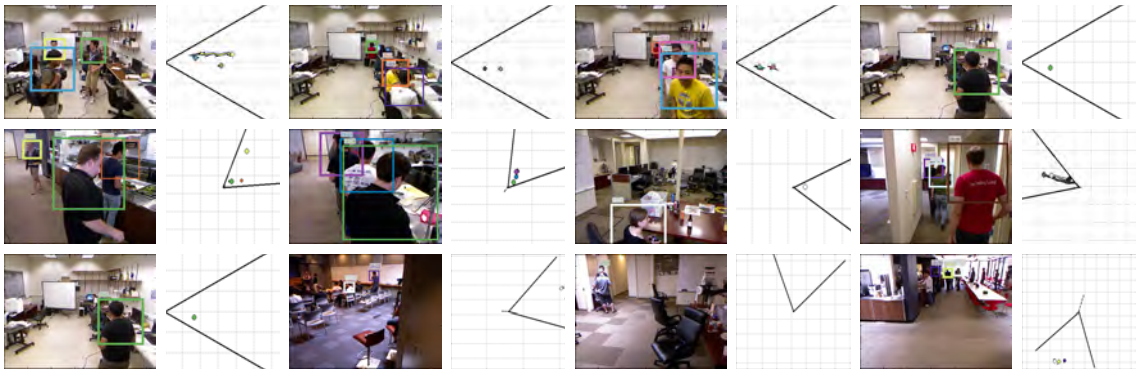


Figure 2.12: Examples of tracking results. First row: results on the *Kinect office dataset*. Second row: results on the *Kinect mobile dataset*. Detections are shown as boxes in images, and dots projected onto the ground plane in the top-down view. Each color is a person. Note that our system detects people in various poses, truncated by the image, and despite the severe occlusions between people that are common in indoor environments. The last row shows examples of challenging scenes where the people appear beyond the Kinect's range or under extreme lighting conditions.

## 2.8 Conclusion

Tracking multiple people, in different environments, performing different tasks and with different relationships will always be a challenging problem. Even humans have a great deal of trouble performing this task; only the best of athletes can predict how their team will move on the field, and the ability requires years of training, a deep knowledge of the team, and the constrained rules of a specific sport. In this chapter, we have laid the groundwork for a general person tracking system and applied it to two specific environments - tracking people from a moving, ground-level camera, and tracking people indoors from a robot platform. We argue that the system is adaptable enough to be applied to other scenarios due to the following characteristics.

**The joint formulation of all variables:** The relationship between the camera, targets' and geometric features' states is combined into a novel probability model, allowing them to influence and improve each other's estimate during inference.

**The combination of multiple observation cues:** By combining multiple de-

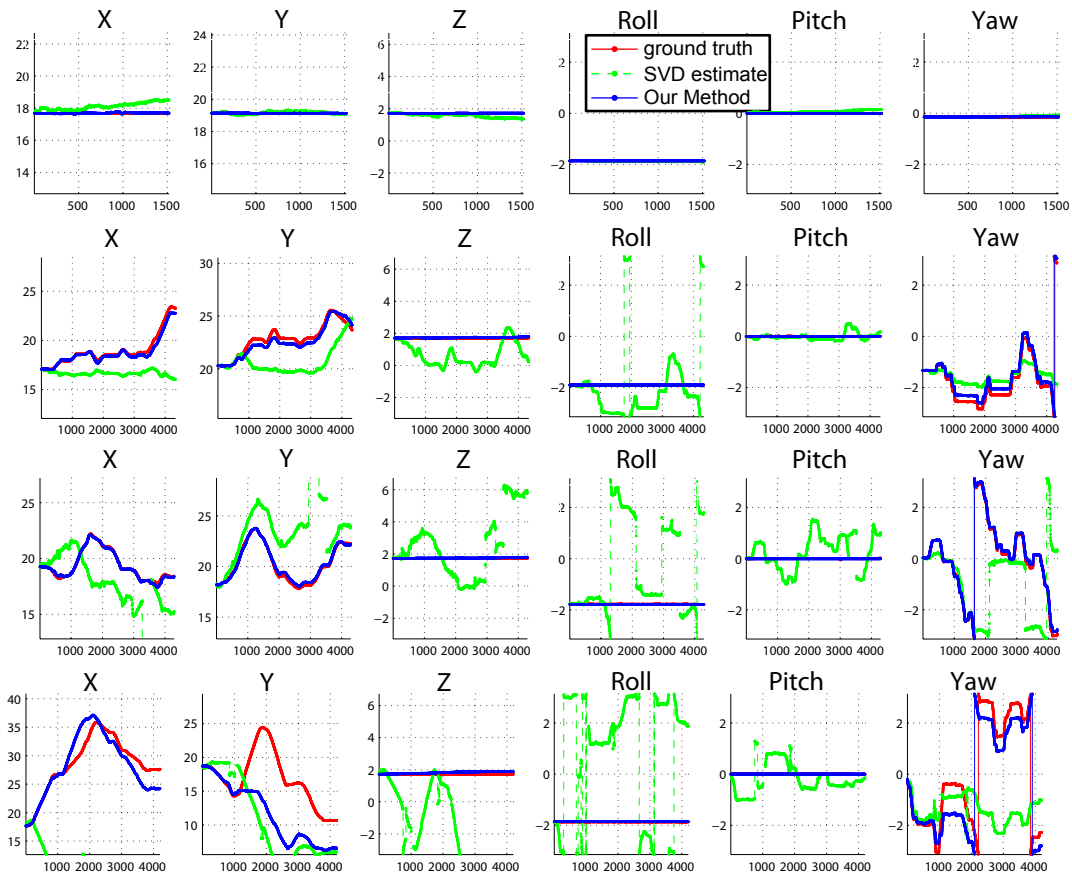



Figure 2.13: Each row represents the 6 estimated camera parameters over time for selected sequences. Our method reliably estimates the camera in most cases, but can fail if there are no features (e.g. camera faces a featureless wall for time frames 1000 $\sim$ 1500 in the last sequence.)

tection cues from different sensor modalities in a principled fashion, our system is adaptable to different sensor configurations and different environments.

**Allowing people to interact:** We do not assume that people move independently, instead we model interaction with two modes: repulsion and group movement. By automatically selecting between interaction modes, the system adapts to different scenarios.

**Automatically detecting people:** Our system automatically detects people, removing the need for manual initialization. Since the detection is probabilistic, the tracker can also recover from missed detections or false positives via motion model

and sampling.

**Automatic detection of static features for camera estimation:** Since it estimates the camera's motion, our system can be applied on sequences acquired from a moving camera, even under the assumption that the odometry of the camera is unknown or poorly specified. The camera estimation is performed automatically using stationary features from the environment.

As we apply this system to additional scenarios in the future, we would like to learn what is the best combination of observation cues for a given sensor suite and environment from a training data. This system can be used as a building block to learn and recognize high level semantics about human activities by providing trajectories of people.

# CHAPTER III

# Recognizing Collective Activities via Crowd Context

In human interactions, activities have an underlying purpose. This purpose can be to accomplish a goal, or to respond to some stimulus. Both of these parameters are governed by the environment of the individuals, which dictates the contextual elements in the scene. Since this environment is shared by all individuals present in the scene, it is often the case that the actions of individuals are interdependent and some coherency between these actions may exist. We call such activities "collective". Examples of collective activities are: Crossing the road, Talking, Waiting, Queuing, Walking, Dancing and Jogging. In this chapter, we seek to recognize such collective activities from videos.

Consider a collective activity "queuing": the definition of the activity itself requires that multiple individuals be present in the scene and waiting their turn in some structure. Over time, the individuals may progress forward in the queue slowly. This queue structure imposes restrictions on what the spatial distribution of individuals over time may look like. Although the queuing individuals are also "waiting", and a few perhaps are also "talking", the predominant group activity remains the one of queuing. We refer to such dominant and coherent behavior over the temporal

This chapter is based on the publications (*Choi et al.* (2009, 2011b); *Choi and Savarese* (2013))

and spatial dimension as crowd context. We argue that crowd context is a critical ingredient for characterizing collective activities.

We introduce the concept of *crowd context* where the spatio-temporal relationship among people is encoded by a Spatio-Temporal-Local (STL) descriptor. Though experimental results tested on the collective activity dataset suggest that the descriptor is already effective in recognizing complex collective activities, such a descriptor has a number of limitations including the fact that the structure of the bins of the STL descriptor must be predefined beforehand. In particular, by assuming that the spatial support has fixed size, the STL descriptor does not have the ability to adaptively filter out background activities or activities that differ from the dominant one. We further extend the concept of crowd context where the crowd context is learned by adaptively binning the spatio-temporal volume as well as the attribute (*e.g.*, pose and velocity of individuals) space using a novel random forest (RF) classification scheme. We call our scheme a Randomized Spatio-Temporal Volume (RSTV) classifier. In our framework, the feature that the trees in a RF operate on, is calculated over a random spatio-temporal volume. Hence, the proposed random forest picks the most discriminating spatio-temporal volume over which to calculate the feature, and then further continues to pick the most discriminating separating plane in order to perform classification as usual in a random forest (*Breiman and Cutler* (2004)). Our adaptive binning strategy: 1) establishes robustness to clutter, 2) is able to incorporate other cues/evidence gracefully for classification, and 3) exhibits parameter free learning under a principled probabilistic framework. We use the Random Forest classifier to associate each individual with a collective activity label, performing local classification. We also propose a subsequent step based on a 3D spatio-temporal Markov Random Field that is leveraged to exploit the temporal and spatial consistency of activities to perform global classification.

## 3.1 Background

A large literature on activity classification has mostly focused on understanding the behavior of humans in isolation (*atomic activities*). *Song et al.* (2003) and *Fanti et al.* (2005) model actions using a constellation of parts and relate the spatial-temporal dependencies of such parts using a probabilistic formulation. *Laptev and Lindeberg* (2003) and *Dollar et al.* (2005) propose a compact characterization of an activity in terms of a sparse set of local spatial-temporal interest points. *Savarese et al.* (2008) introduce a framework for incorporating 2D spatial-temporal short and long term dependencies into a bag-of-words representation. *Niebles et al.* (2008) introduce an unsupervised learning method for modeling activities that leverage the construction of latent intermediate visual concepts. Other interesting formulations can be found in the works by *Yu et al.* (2010); *Kim et al.* (2007); *Wong et al.* (2007); *Liu et al.* (2008, 2011a,b); *Lu and Little* (2006); *Lv and Nevatia* (2007); *Marszalek et al.* (2009) and are nicely summarized in the survey by *Turaga et al.* (2008). Progress on atomic activity recognition is coupled with the effort of collecting datasets of human activities that appear in images or videos. Early notable examples are the *KTH* dataset of *Laptev and Lindeberg* (2003) and the *Wisemann* dataset of *Gorelick et al.* (2007). More recent collections are proposed by *Liu et al.* (2009) (a data set of videos from a public video repository (YouTube)) and *Laptev et al.* (2008) (a collection of video sequences from Hollywood movies) which provide a test-bed that is closer to real world application scenarios. Recently, *Niebles et al.* (2010) propose to model structured atomic activities such as those that appear in sport events (e.g., *tennis-serve*, *triple-jump*, etc) and provide a large dataset for enabling quantitative evaluation.

While successful, however, most of these methods are targeted to atomic activities. Research by *Ryoo and Aggarwal* (2009), *Yao et al.* (2010) and *Patron et al.* (2010) goes beyond single-person activity understanding and propose methods for modeling interactions between pairs of individuals. The extension to activities that involve

more than two individuals has been investigated in a number of works including *Ryoo and Aggarwal* (2010); *Lan et al.* (2010a); *Amer and Todorovic* (2011). In *Ryoo and Aggarwal* (2010) complex group activities are analyzed using a stochastic context free grammar model with a number of predefined activity predicates. *Lan et al.* (2010a,b) propose to encode 2D interactions among individuals using the contextual information that originates from higher level activity semantics. We focus on activities that are characterized by a larger number of individuals (e.g., the collective activities) and propose to capture the collective behavior using a descriptor called *Crowd Context*. We also propose one of the first data sets that include challenging videos of collective activities. Moreover, a number of works (*Intille and Bobick* (2001); *Li et al.* (2009); *Swears and Hoogs* (2011)) focus on group activities that appears in sport events such as a football game. Specifically, *Intille and Bobick* (2001) model trajectories of individuals with a Bayesian network, *Li et al.* (2009) introduce a discriminative temporal interaction manifold for modeling activities *Li et al.* (2009) and *Swears and Hoogs* (2011) propose a non-stationary kernel hidden Markov model to capture temporal dependencies. Notice that most of these methods require different degrees of manual annotations in identifying human trajectories in time and space. Finally, at the opposite side of the spectrum, research by *Ramin Mehran and Shah* (2009); *Hakeem and Shah* (07); *Zhou et al.* (2012) seeks to study the semantic properties of large crowds of individuals. These methods, however, go beyond the scope of this chapter in that the focus is on modeling large crowds as a whole, without considering the individual behavior of the actors.

## 3.2 Crowd Context for Collective Activity Recognition

In this section, we introduce the definition of the crowd context and describe its mathematical formulation given a set of spatio-temporal trajectories. The concept of crowd context is defined as *a coherent behavior of individuals in time and space* that

are performing a certain collective activity.

The crowd context is captured by introducing a new descriptor called Spatio-Temporal-Local (STL) descriptor that encodes the spatial-temporal dependencies of individuals in a neighborhood of the video sequence. The STL descriptor is in essence a fixed-dimensional vector (Fig.3.1) and is associated to each person. For each time stamp, the STL descriptors are used to classify the collective activity using a standard Support Vector Machine (SVM) (*Chang and Lin* (2001)) classifier. Temporal smoothness is enforced by applying a markov chain model across each time stamp. Though the method shows promising results, such rigid descriptors require the parameters that control the structure of the descriptor to be manually specified, which can be extremely difficult in presence of large intra-class variability. We address such limitation in the later section where a new scheme called Randomized Spatio Temporal Volume (RSTV) is used to automatically learn the best structure of the descriptor. In following sections, we discuss the rigid STL descriptor first and the extended RSTV later.

### 3.2.1 Rigid STL Descriptor

In this section, we describe how to extract an STL descriptor for each individual (track) in each time stamp given a set of trajectories $\mathbb{T} = \{T_1, T_2, ..., T_N\}$, where $T_i = \{l_i, p_i, t_i\}$ is an individual track and $l_i = (x_i, y_i)$, $p_i$ and $t_i$ are sequences of x, y location, pose and time index, respectively. Note that the pose captures the orientation of an individual in this framework (*e.g.* left, front, right, and back). The location of individual target $l_i$ is obtained by accumulating the estimated state $Z_t^i$ acquired by any multi-target tracking method (such as the one discussed in the chapter II) and pose $p_i$ is acquired by using SVM classifier equipped with HoG descriptor.

Given a person $i$ in certain time stamp $t$ (the *anchor*), they determine the locations $l_j^i$ and poses $p_j^i$ of other individuals in the anchor's coordinate system, where the

anchor's coordinate system has the origin at the anchor's $(x, y)$ location and is oriented along the pose direction of the anchor (see Fig.3.1 top).

The space around each anchor $i$ at time $t$ is divided into multiple bins following a log-polar space partition similar to the shape context descriptor (*Belongie et al.* (02)). Moreover, for each spatial bin, $P$ "pose" bins are considered where $P$ is the number of poses that are used to describe a person orientation. Finally, the temporal axis is also decomposed in temporal bins around time stamp $t$. This spatial, temporal and pose sensitive structure is used to capture the distribution of individuals around the anchor $i$ at time $t$ and construct the STL descriptor. For each anchor $i$ and time stamp $t$, an STL descriptor is obtained by counting the number of individuals that fall in each bin of the structure described above. Thus, the STL descriptor implicitly embeds the *flow* of people around the anchor over a number of timestamps. After accumulating the information, the descriptor is normalized by the total number of people that fall in the spatio-temporal extension of the descriptor.

There are a number of important characteristics of the STL descriptor. First, the descriptor is rotation and translation invariant. Since the relative location and pose of individuals are defined in the anchor's coordinate system, the descriptor yields a consistent representation regardless of the orientation and location of the anchor in the world. Moreover, the dimensionality of the descriptor is fixed regardless of the number of individuals that appear in the video sequence. This property is desirable in that it allows to represent an activity using a data structure that is not a function of the specific instantiation of a collective activity. Finally, by discretizing space and time into bins, the STL descriptor enables a classification scheme for collective activities that is robust to variations in the spatio-temporal location of individuals for each class of activity (intra-class variation).

Given a set of STL descriptors (each person in the video is associated to a STL descriptor) along with the associated collective activity labels, one can solve the

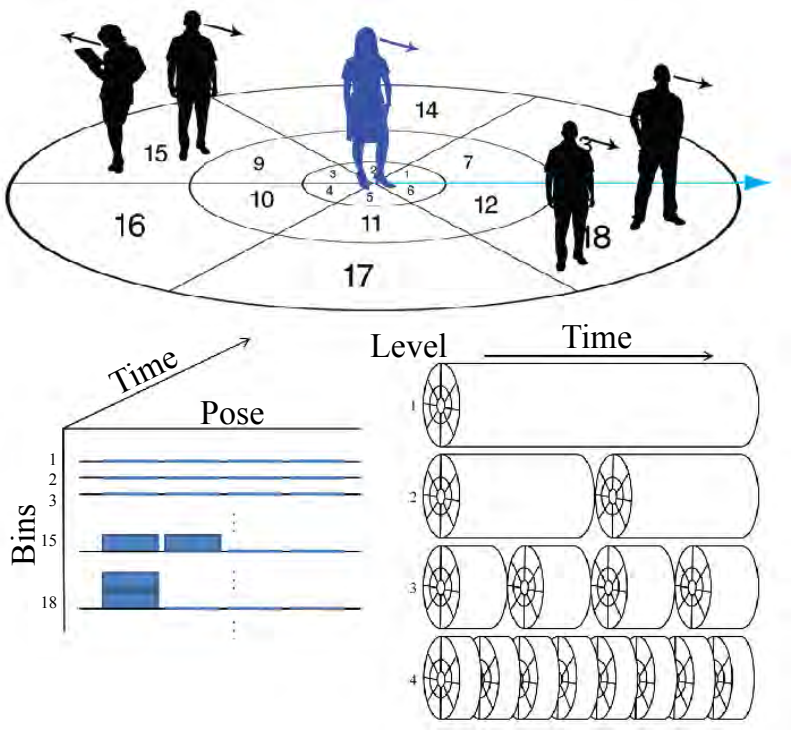



Figure 3.1: Spatio-Temporal Local Descriptor. (a) Space around anchor person (blue) is divided into multiple bins. The pose of the anchor person (blue arrow) locks the "orientation" of the descriptor which induces the location of the reference bin "1". (b) Example of STL descriptor - the descriptor is a histogram capturing people and pose distribution in space and time around the anchor person. (c) Classification of STL descriptor is achieved by decomposing the histogram in different levels along the temporal axis.

collective activity classification problem by using a classification method such as SVM (*Chang and Lin* (2001)). In order to capture various levels of temporal granularity, we adopt SVM classifier equipped with a temporal pyramid intersection kernel (see Fig.3.1 bottom right). The temporal axis is divided into 4 hierarchical levels of temporal windows and intersection kernel is defined per each level. The finest temporal window allows to capture the detailed motion of individuals around the anchors; the highest level allows to encode the overall distribution of people around the anchor over the observed period.

### 3.2.2 Learning the Crowd Context

Even though the STL descriptor has been successfully employed for collective activity classification, it is limited in that the structure of the bins of the STL descriptor is predefined beforehand and parameters such as the minimum distance from the anchor or the maximum support of the descriptor are defined once for all. In particular, by assuming that the spatial support has fixed size, the STL descriptor does not have the ability to adaptively filter out background activities or activities that differ from the dominant one.

In order to avoid above mentioned limitations, we propose a novel scheme, called Randomize Spatio-Temporal Volume (RSTV). The RSTV approach is based on the same intuition as STL that crowd context can be captured by counting the number of people with a certain pose and velocity in fixed regions of the scene, relative to an anchor person. However, RSTV extends this intuition and considers variable spatial regions of the scene with a variable temporal support. The full feature space contains the evidence extracted from the entire videos: the location of each individual in anchor's coordinates as well as the velocity & pose of each individual per video frame. This can be interpreted as a soft binning scheme where the size and locations of bins are estimated by a random forest so as to obtain the most discriminative regions in the feature space. Over these regions, the density of individuals is inspected, which can be used for classification. Fig.3.2 compares the rigid STL binning scheme and the flexible RSTV. RSTV is a generalization of the STL in that the rigid binning restriction imposed in the STL is removed. Instead, portions of the continuous spatio-temporal volume are sampled at random and the discriminative regions for classification of a certain activity are retained. RSTV provides increasing discrimination power due to increased flexibility.

There are several benefits of the RSTV framework over rigid STL descriptor. 1) The RSTV automatically determines the discriminative features in the feature space

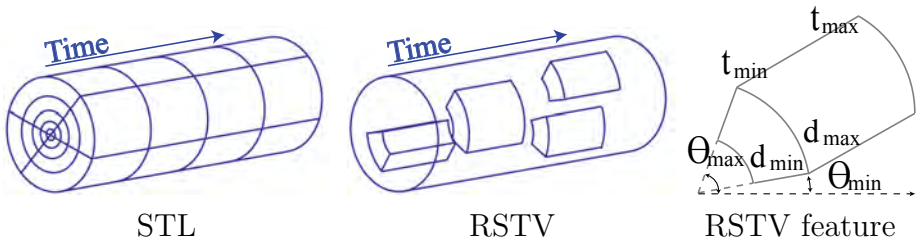


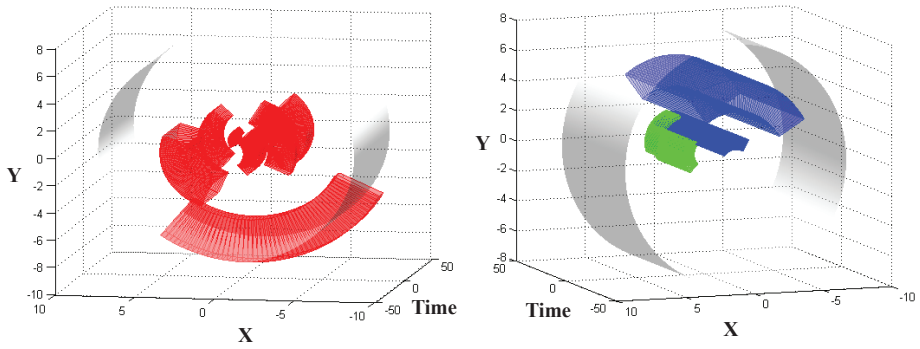

Figure 3.2: STL counts the number of people in each spatio-temporal and pose bins that are divided by a hand defined parameters (*left*). On the other hand, the RSTV learns what spatial bins are useful (shown as a trapezoid-like volume) in order to discriminate different collective activities and discards the regions (shown as empty regions) that are not helpful for such discrimination task (*middle*). A random spatio-temporal volume (feature) is specified by a number of parameters (*right*). Pose and velocity are omitted from the illustration.

that are useful for classification. Indeed, while STL proposes a rigid and arbitrary decomposition of the feature space, in RSTV the binning space is partitioned so as to maximize discrimination power. 2) Unlike STL, there are no parameters that are to be learned or selected empirically (e.g. support distance, number of bins). 3) It enables robustness to clutter. Indeed, unlike STL, the RSTV does not operate given fixed parameters such as radial support and number of spatial bins, but explores the possible space of parameters; thus the density feature, using which classification is performed, is only calculated over regions relevant to each different activity. Hence the classification evidence is pertinent to each activity and avoid clutter that possibly arises from hard-coded framework parameters that may be tuned to achieve optimal classification of a few activities, but not all. Notice that STL concept is similar to the Shape Context (*Belongie et al.* (02)) descriptor, which is known to be susceptible to clutter due to non discriminative inclusion of all points within the radial support.

**Learning RSTV with Random Forest :** The Random Forest classifier is used to learn the structure of RSTV given training data. A Random forest (*Breiman and Cutler* (2004)) is an ensemble of many singular classifiers known as decision trees

which is trained from a portion of the training data. The training set is subdivided into multiple *bags* by random sampling with replacement (*bagging*) in order to reduce the effect of over-fitting. Given each set, one random decision tree is trained following successively drawing and selection of a random feature that best discriminates the given training set (*Breiman and Cutler* (2004)).

The RSTV is trained based on the random forest classifier given a set of training data and associated activity labels $(x_i, y_i)$ where each data point is defined for each person and time stamp. In following description, it is assumed that the trajectories and poses of all people are already transformed into the anchor's coordinate system to form data point $x_i$ and associated activity label $y_i$. Given a random bag, a random decision tree is learned by recursively discovering the most discriminative features. The algorithm firstly randomizes over different volumes of the feature space and secondly randomizes over different decision thresholds given the feature subspace. The feature is defined as the number of people lying in a spatio-temporal volume that is specified by location $(l^k)$, velocity $(v^k)$, pose $(p^k)$ and time $(t)$ defined in the anchor's $(k)$ coordinate system. A unique spatio-temporal volume is specified by a number of parameters : 1) minimum and maximum distance $d_{min}, d_{max}$, 2) minimum and maximum angle in the space $\theta_{min}$, $\theta_{min}$, 3) relative orientation/pose $p$, 4) temporal window $t_{min}$, $t_{max}$ and 5) minimum and maximum velocity $v_{min}, v_{max}$ (Fig.3.2 right). In each node, a number $M$ of such hyper-volume $r_n$ and a scalar decision threshold $d_n$ is drawn randomly multiple times. Given the feature pair $(r_n, d_n)$, the training data is partitioned into two subsets $I_r$ and $I_l$ by testing $f(x; r_n) > d_n$, where $f(x; r_n)$ is a function that counts the number of people lying in the hyper volume $r_n$. Among the set of candidate features, the one that best discriminates the training data into two partitions is selected by examining the information gain (Eq.3.1).




(a) Waiting (b) Talking

Figure 3.3: Example of learned RSTV regions. **(a)** & **(b)** illustrate a set of RSTV regions learned automatically by a single tree. Each colour indicates different pose of neighbouring individuals (up - red, down - blue and right - green). Each RSTV is oriented such that the anchor is facing in the upward $z$ direction. Hence **(a)** indicates that while waiting, an anchor is surrounded on the left and right by people facing the same direction. RSTV in **(b)** illustrates that during talking the anchor and neighbour face each other and are in very close proximity. Note that each RSTV needs only capture some coherent portion of evidence since there exist many trees in the RF. $x$ and $z$ have units of meters while time is measured in frames.

$$\Delta E = -\frac{|I_l|}{|I|}E(I_l) - \frac{|I_r|}{|I|}E(I_r)$$
$$E(I) = -\sum_{i=1}^{C} p_i \; log_2(p_i) \tag{3.1}$$

$I_l$ and $I_r$ are the partition of set $I$ divided by given feature, $C$ is the number of activity classes, $p_i$ is the proportion of collective activity class $i$ in set $I$, and $|I|$ is the size of the set $I$. Typical examples of learned RSTV structure is shown in Fig.3.3. The detailed algorithm for learning RSTV is presented in Alg.1 and Alg.2.

**Algorithm 1** RSTV learning

**Require:** $I = \{(x_i, y_i)\}$
  Randomly draw a bag $I_t$ for each tree
  **for all** random decision tree **do**
    At the root node, $root \leftarrow NodeLearn(I_t)$
  **end for**

Given the learned RSTV forests, one can classify a novel testing example $x$ by passing down the example along each tree and taking the class that maximizes marginal posterior probability $P(y|x) = \sum_{tree} P_{tree}(y|x)$ over all trees. The posterior probability of a tree is defined as the corresponding $p_y$ in the leaf node that the testing example reached in the decision tree.

**Algorithm 2** Recursive Node Learning (NodeLearn)

**Require:** $I_n$
  **if** $|I_n| < N_{min}$ **then**
    Compute distribution of classes $p_i$ over all $C$
    $node.isleaf \leftarrow TRUE$
    $node.p \leftarrow p_i$
    **return** *node*
  **end if**
  $\Delta E_{max} \leftarrow -INF$
  **for** $m = 0$ **to** $M$ **do**
    Randomly draw a feature pair $(r_n^m, d_n^m)$
    Compute information gain $\Delta E_m$
    **if** $\Delta E_{max} < \Delta E_m$ **then**
      $\Delta E_{max} \leftarrow \Delta E_m$
      $(r_n, d_n) \leftarrow (r_n^m, d_n^m)$
    **end if**
  **end for**
  Partition $I_n$ into $(I_l, I_r)$ using $(r_n, d_n)$
  $node.isleaf \leftarrow FALSE$
  $node.left \leftarrow NodeLearn(I_l)$
  $node.right \leftarrow NodeLearn(I_r)$
  $node.feature \leftarrow (r_n, d_n)$
  **return** *node*

## 3.3 Globally Consistent Classification with Markov Random Field

The STL and RSTV models allow to classify each person in the video individually and associate a collective activity label to it. If the scene contains only one or few collective activities, however, one can impose some level of spatial or temporal regularization across labelling assignments. Such regularization helps mitigate the

classification errors due to the inherent noise in constructing the STL/RSTV descriptors as well as the intrinsic ambiguities in discriminating collective activities. This regularization is modeled using a Markov Random Field (MRF) over both space and time.

An MRF is a general model that can encode the correlation between many random variables in a coherent fashion. Such model is frequently adopted in the image segmentation problem in order to provide consistency between spatially adjacent pixels (*Kohli and Torr* (2010)). We propose to use an MRF to capture the local and spatial coherency of labelling assignments. MRF favor to assign the same activity label to nearby people in a single time stamp and to a person over adjacent timestamps. The intuition is that 1) nearby people tend to participate in the same activity and 2) a person tends to perform the same activity in nearby time stamps. Such model can be formulated as follows

Let $x_t^i$ and $y_t^i$ denote the data and collective activity labels associated with an individual person $i$ at a certain time stamp $t$. Then the posterior probability over all activity labels $y$ given all input $x$ can be represented as:

$$P(y|x,l) \propto \prod_t \prod_i P(y_t^i|x_t^i) \prod_t \prod_{(i,j)\in E_s} \Phi_S(y_t^i, y_t^j; l_t^i, l_t^j)$$
$$\prod_i \prod_t \Phi_T(y_{t-1}^i, y_t^i) \tag{3.2}$$

where $l_t^i$ is the location of person $i$ in $t$, $E_s$ is the set of edges between people (Fig.3.4), $P(y_t^i|x_t^i)$ is the unary probability estimate from Random Forest for a person $i$ in time $t$, $\Phi_S(y_t^i, y_t^j; l_t^i, l_t^j)$ is the spatial pair-wise potential, and $\Phi_T(y_{t-1}^i, y_t^i)$ is the temporal pairwise potential. The temporal edges are established between temporally adjacent nodes of the same person. The two nodes in the same time-stamp are connected if they are close to each other ($< 2$ meter) in order to enforce similar labelling between the two. The maximum-a-posteri (MAP) solution of the MRF can be obtained by

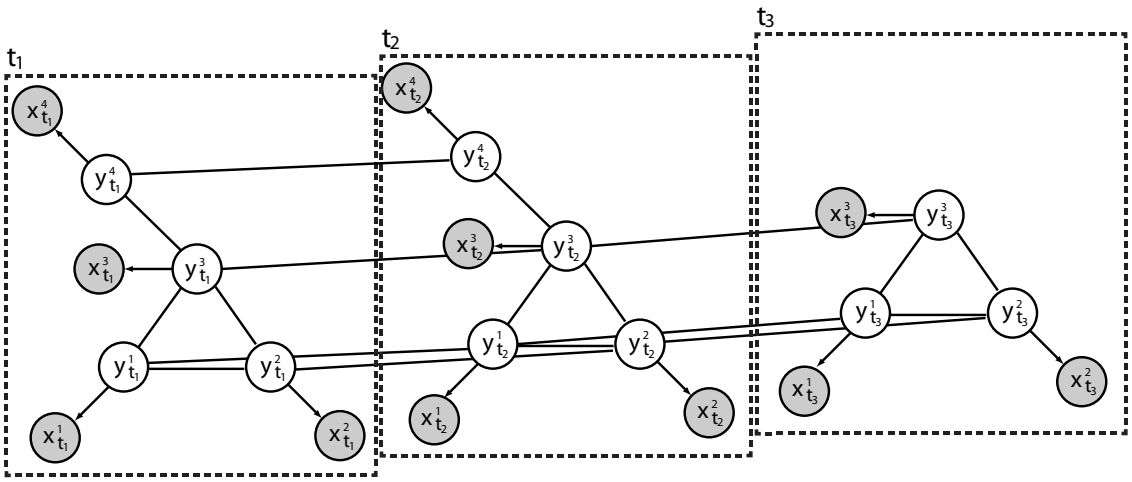



Figure 3.4: Graphical representation for the proposed MRF over collective activity variables $y$. $y^j_{t_i}$ models the activity of a person in one time slice (hidden variable), $x^j_{t_i}$ represents the trajectories associated to an anchor person. If two people are close enough ($\leq$ 2 meter away), the spatial edges are inserted to inject spatial coherency. For every person, temporal edges are constructed between nearby nodes.

a Gibbs sampling procedure (*Bishop* (2006)) given the parameters for the pairwise potentials. We obtain the temporal pairwise potentials by counting and normalize the co-occurrence of a pair of collective activity. The spatial potentials are estimated in a non-parametric way by collecting location difference oriented with respect to each person's pose for all activity pairs.

## 3.4 Experimental Results

In this section, we present an overview of the classification results obtained using the STL and RSTV crowd context descriptors. Both methods are evaluated using a dataset we collected (Collective Activity Dataset available at `http://www.eecs.umich.edu/vision/activity-dataset.html`). Though there exist many different types of datasets for human activity recognition such as CAVIAR, IXMAS, KTH or UIUC, none of them are suitable for the proposed descriptors in that they focus on

activities performed by a single or very few actors.

**Dataset :** We collected two versions of collective activity datasets. The first version of the dataset is composed of 5 different collective activity categories, *Crossing, Standing, Queuing, Walking* and *Talking*. It includes 44 short video clips each of which is recorded from a real world scene with a variable number of people. The second version of the dataset includes 6 different collective activity categories, *Crossing, Standing, Queuing, Talking, Dancing* and *Jogging*. Similarly to the first version, the second version of the dataset has 74 short video clips with a variable number of people in the scene (see Fig.3.7 for examples).

In both sequences, the videos were taken from a hand-held camera with a unpredictable camera motion incurred by jittering of the hand. Thus, our tracking method (described in the chapter II) is used in order to obtain the 3D trajectories of people in the videos.

**Activity Classification Results:** Table.3.1 shows the comparison among several state-of-the-art and baseline methods for collective activity classification evaluated on the two collective activity datasets. Each row represents the overall classification accuracy and the columns represent different versions of the datasets. The first method is the Action Context (AC) proposed by *Lan et al.* (2010a). AC is another type of contextual descriptor that accumulates the activity classifier confidence of both the anchor person and surrounding people. The second method is the STL descriptor equipped with SVM classifier. The third method augments the second method by adding a markov chain over an individual. We use a markov chain for each individual person in order to utilize temporal smoothness of the collective activity, i.e. a person doing *Crossing* tends to continue doing *Crossing* in next timestamps. The fourth method is the STL descriptor equipped with the random forest classifier. As

| Dataset | 5 Activities | 6 Activities |
|---|---|---|
| AC *Lan et al.* (2010a) | 68.2% | - |
| STL | 64.3% | - |
| STL+MC | 65.9% | - |
| STL+RF | 64.4% | - |
| RSTV | 67.2% | 71.7% |
| RSTV+MRF | **70.9%** | **82.0%** |

Table 3.1: Average classification results of various state-of-the-art *Lan et al.* (2010a) and baseline methods on the dataset with 5 activities (left column) and 6 activities (right column). See text for details.

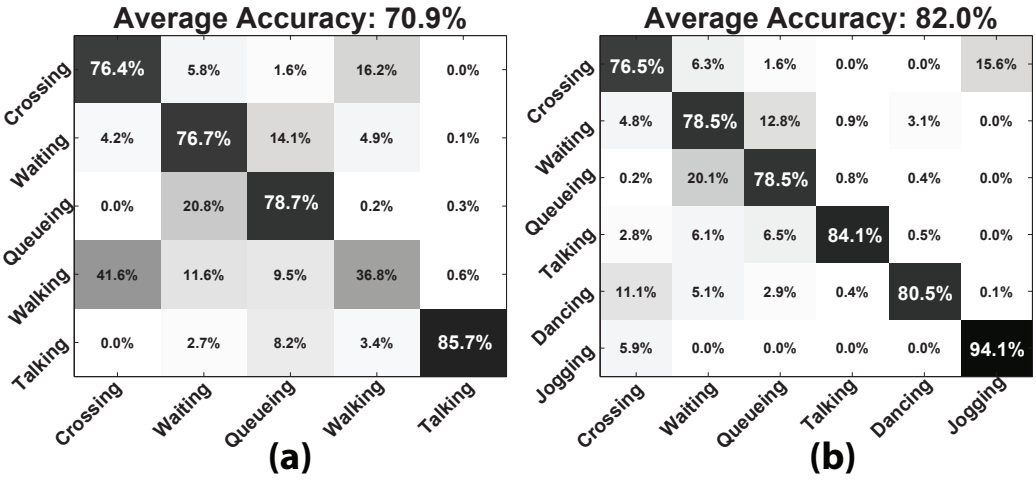



Figure 3.5: The confusion tables using RSTV with MRF regularization on the dataset with 5 activities (a) and 6 activities (b).

noted in the comparison between the second and fourth methods, a mere replacement of the SVM classifier with a Random Forest does not yield an improvement in the collective activity classification. The fifth method is the RSTV and the last is RSTV equipped with MRF regularization. As shown in the table, the RSTV with MRF method gives the most robustness in collective activity classification thanks to the flexibility in learning the contextual information. All results presented are obtained using a leave-one-video-out training and testing scheme. In the future, we plan to study the statistical significance of the proposed algorithms using multiple datasets as suggested by *Demšar* (2006).

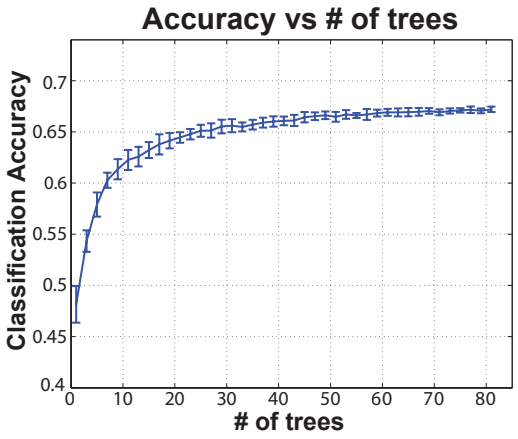



Figure 3.6: Classification accuracy by RSTV using different number of trees. As the number of trees increases, the classification accuracy also improves and converges at around 60 trees. The 5-category dataset is used in this experiment. Vertical bars measure the standard deviation around the average classification accuracy.

Fig.3.5 presents the confusion table for the collective activity dataset using the RSTV with MRF regularization. In the 5 category dataset experiment, the most confusion in classification occurs in discriminating the *Crossing* and *Walking* classes. This seems reasonable since the two classes share similar spatio-temporal properties. In the augmented 6 category experiment, the method produces more stable classification results since each collective activity category has distinctive spatio-temporal characteristics which can be more effectively captured by the crowd context descriptors.

Since each tree in RSTV forest is trained using the *bagging* procedure, each tree captures different spatio-temporal characteristics for each collective activity class. Thus having larger number of trees would provide more robust classification results in general. Such trend is shown in Fig.3.6. When only few trees are used in the experiment, classification results by RSTV are rather unstable. As the number of trees increases, the classifier becomes more robust and converges to the best accuracy at 60 trees.


Figure 3.7: Example results on the 6-category dataset using RSTV with MRF. Top 3 rows show examples of good classification and bottom row shows examples of false classification. The labels X (magenta), S (blue), Q (cyan), T (orange), D (red), J (green) and NA (white) indicate *crossing*, *waiting*, *queuing*, *talking*, *dancing*, *jogging* and not assigned, respectively. When there is insufficient evidence to perform classification, the NA label is displayed. The misclassified results indicate that miss classifications mostly occur between classes with similar structure. This figure is best viewed in color.

## 3.5 Conclusion

In this chapter, we have reviewed a recent formulation for classifying collective activities that takes advantage of the concept of *crowd context* and introduced two descriptors (STL and RSTV) to model the crowd context. Experimental evaluation indicates that the crowd context is a powerful and robust source of information to discriminate different types of collective activities.

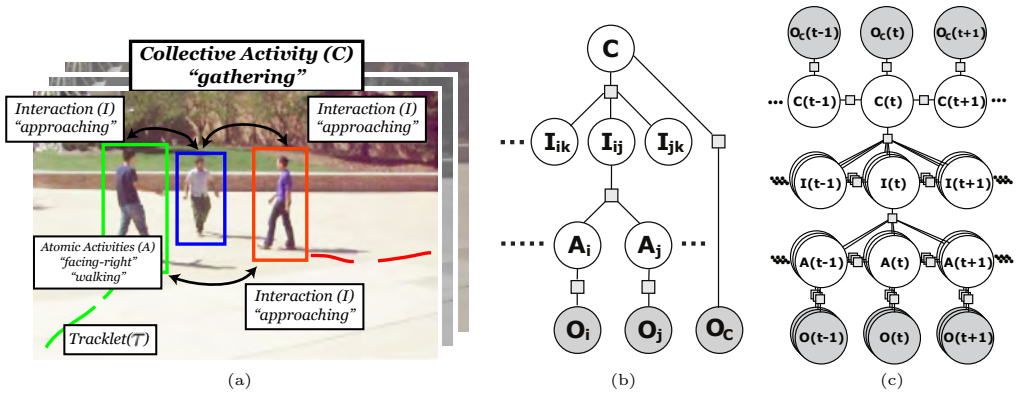
# CHAPTER IV

# Unified Model for Tracking Multiple Targets and Recognizing Activities at Multiple Levels

There are many degrees of granularity with which we can understand the behavior of people in video. We can detect and track the trajectory of a person (as discussed in the chapter II), we can observe a person's pose and discover what *atomic activity* (*e.g.*, *walking*) they are performing, we can determine an *interaction activity* (*e.g.*, *approaching*) between two people, and we can identify the *collective activity* (*e.g.*, *gathering*) of a group of people (as discussed in the chapter III). These different levels of activity are clearly not independent: if everybody in a scene is walking, and all possible pairs of people are approaching each other, it is very likely that they are engaged in a gathering activity. Likewise, a person who is gathering with other people is probably walking toward a central point of convergence, and this knowledge places useful constraints on our estimation of their spatio-temporal trajectory.

Regardless of the level of detail required for a particular application, a powerful activity recognition system will need to exploit the dependencies between different levels of activity. Such a system should reliably and accurately: (i) identify stable and coherent trajectories of individuals; (ii) estimate attributes, such as poses, and infer atomic activities; (iii) discover the interactions between individuals; (iv) recog-

This chapter is based on the publication (*Choi and Savarese* (2012)).




(a) (b) (c)

Figure 4.1: In this chapter, we aim at jointly and robustly tracking multiple targets and recognizing the activities that such targets are performing. **(a)**: The collective activity *"gathering"* is characterized as a collection of interactions (such as *"approaching"*) between individuals. Each interaction is described by pairs of atomic activities (e.g. *"facing-right"* and "*facing-left"*). Each atomic activity is associated with a spatial-temporal trajectory (tracklet $\tau$). We advocate that high level activity understanding helps obtain more stable target trajectories. Likewise, robust trajectories enable more accurate activity understanding. **(b)**: The hierarchical relationship between atomic activities ($A$), interactions ($I$), and collective activity ($C$) in one time stamp is shown as a factor graph. Squares and circles represent the potential functions and variables, respectively. Observations are the tracklets associated with each individual along with their appearance properties $O_i$ as well as crowd context descriptor $O_c$ (Sec.4.2.1). **(c)**: A collective activity at each time stamp is represented as a collection of interactions within a temporal window. Interaction is correlated with a pair of atomic activities within specified temporal window (Sec.4.2.2). Non-shaded nodes are associated with variables that need to be estimated and shaded nodes are associated with observations.

nize any collective activities present in the scene. Even if the goal is only to track individuals, this tracking can benefit from the scene's context. Even if the goal is only to characterize the behavior of a group of people, attention to pairwise interactions can help.

Much of the existing literature on activity recognition and tracking (*Scovanner and Tappen* (2009); *Pellegrini et al.* (2009); *Leal-Taixe et al.* (2011); *Choi and Savarese* (2010); *Khan et al.* (2005); *Yamaguchi et al.* (2011); *Intille and Bobick* (2001); *Li et al.* (2009); *Lan et al.* (2010b)) avoids the complexity of this context-rich approach

by seeking to solve the problems in isolation. We instead argue that tracking, track association, and the recognition of atomic activities, interactions, and group activities must be performed completely and coherently. In this chapter, we introduce a model that is both principled and solvable and that is the first to successfully bridge the gap between tracking and group activity recognition (Fig.4.1).

## 4.1 Background

As discussed in the chapter II, target tracking is one of the oldest problems in computer vision, but it is far from solved. In difficult scenes, tracks are not complete, but are fragmented into tracklets. It is the task of the tracker to associate tracklets in order to assemble complete tracks. Tracks are often fragmented due to occlusions. Recent algorithms address this through the use of detection responses (*Wu and Nevatia* (2007); *Ess et al.* (2008)), and pairwise interaction models (*Scovanner and Tappen* (2009); *Pellegrini et al.* (2009); *Leal-Taixe et al.* (2011); *Choi and Savarese* (2010); *Khan et al.* (2005); *Yamaguchi et al.* (2011)). The interaction models, however, are limited to a few hand-designed interactions, such as attraction and repulsion. Methods such as *Rodriguez et al.* (2009) leverage the consistency of the flow of crowds with models from physics, but do not attempt to associate tracklets or understand the actions of individuals. *Zhang et al.* (2008); *Pirsiavash et al.* (2011) formulate the problem of multi-target tracking into a min-cost flow network based on linear/dynamic programming. As also discussed in the chapter II, both model interactions between people, they still rely on heuristics to guide the association process via higher level semantics.

A number of methods have recently been proposed for action recognition by extracting sparse features by *Dollar et al.* (2005), correlated features by *Savarese et al.* (2008), discovering hidden topic models by *Niebles et al.* (2008), or feature mining by *Liu et al.* (2009). These works consider only a single person, and do not benefit

from the contextual information available from recognizing interactions and activities. *Ryoo and Aggarwal* (2009) models the pairwise interactions between people, but the model is limited to local motion features. Several works address the recognition of planned group activities in football videos by modelling the trajectories of people with Bayesian networks (*Intille and Bobick* (2001)), temporal manifold structures (*Li et al.* (2009)), and non-stationary kernel hidden Markov models (*Swears and Hoogs* (2011)). All these approaches, however, assume that the trajectories are available (known). As discussed in the chapter III, *Ni et al.* (2009) recognizes group activities by considering local causality information from each track, each pair of tracks, and groups of tracks. *Ryoo and Aggarwal* (2010) models a group activity as a stochastic collection of individual activities. None of these works exploit the contextual information provided by collective activities to help identify targets or classify atomic activities. *Lan et al.* (2010b) uses a hierarchical model to jointly classify the collective activities of all people in a scene, but they are restricted to modelling contextual information in a single frame, without seeking to solve the track identification problem. Finally, *Ramin Mehran and Shah* (2009) recognizes the overall behavior of large crowds using a social force model, but does not seek to specify the behaviour of each individual.

The contributions of this chapter are four-fold: we propose (i) a model that merges for the first time the problems of collective activity recognition and multiple target tracking into a single coherent framework; (ii) a novel path selection algorithm that leverages target interactions for guiding the process of associating targets; (iii) a new hierarchical graphical model that encodes the correlation between activities at different levels of granularity; (iv) quantitative evaluation on a number of challenging datasets, showing superiority to the state-of-the-art.

## 4.2 Modelling Collective Activity

Our model accomplishes collective activity classification by simultaneously estimating the activity of a group of people (*collective activity* $C$), the pairwise relationships between individuals (*interactions activities* $I$), and the specific activities of each individual (*atomic activities* $A$) given a set of observations $O$ (see Fig.4.1). A collective activity describes the overall behavior of a group of more than two people, such as *gathering*, *talking*, and *queuing*. Interaction activities model pairwise relationships between two people which can include *approaching*, *facing-each-other* and *walking-in-opposite-directions*. The atomic activity collects semantic attributes of a tracklet, such as poses (*facing-front*, *facing-left*) or actions (*walking*, *standing*). Feature observations $O = (O_1, O_2, ...O_N)$ operate at a low level, using tracklet-based features to inform the estimation of atomic activities. Collective activity estimation is helped by observations $O_C$, which use features such as spatio-temporal local descriptors *Choi et al.* (2009, 2011b) to encode the flow of people around individuals. At this time, we assume that we are given a set of tracklets $\tau_1, ..., \tau_N$ that denote all targets' spatial location in 2D or 3D. These tracklets can be estimated using methods discussed in the chapter II. Tracklet associations are denoted by $T = (T_1, T_2, ..., T_M)$ and indicate the association of tracklets. We address the estimation of $T$ in Sec.4.3.

The information extracted from tracklet-based observations $O$ enables the recognition of atomic activities $A$, which assist the recognition of interaction activities $I$, which are used in the estimation of collective activities $C$. Concurrently, observations $O_c$ provide evidence for recognizing $C$ (see also the chapter III), which are used as contextual clues for identifying $I$, which provide context for estimating $A$. The bi-directional propagation of information makes it possible to classify $C$, $A$, and $I$ robustly, which in turn provides strong constraints for improving tracklet association $T$. Given a video input, the hierarchical structure of our model is constructed dynamically. An atomic activity $A_i$ is assigned to each tracklet $\tau_i$ (and observation $O_i$),

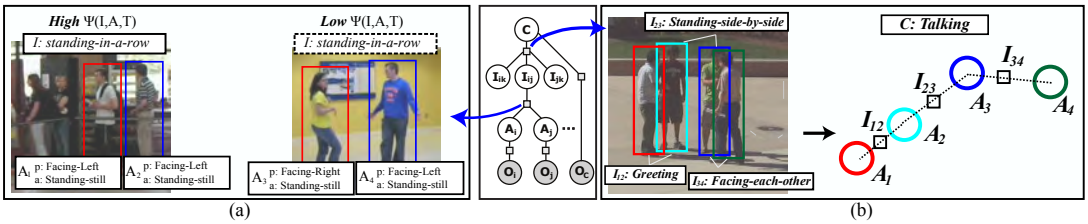



Figure 4.2: **(a)**: Each interaction is represented by a number of atomic activities that are characterized by an action and pose label. For example, with interaction $I$ = *standing-in-a-row*, it is likely to observe two people with both $p$ = *facing-left* and $a$ = *standing-still*, whereas it is less likely that one person has $p$ = *facing-left* and the other $p$ = *facing-right*. **(b)**: Collective activity $C$ is represented as a collection of interactions $I$. For example, with $C$ = *talking* collective activity, it is likely to observe the interaction $I_{34}$ = *facing-each-other*, and $I_{23}$ = *standing-side-by-side*. The consistency of $C, I_{12}, I_{23}, I_{34}$ generates a high value for $\Psi(C, I)$.

an interaction variable $I_{ij}$ is assigned to every pair of atomic activities that exist at the same time, and all interaction variables within a temporal window are associated with a collective activity $C$.

### 4.2.1 The model

The graphical model of our framework is shown in Fig.4.1. Let $O = (O_1, O_2, ...O_N)$ be the $N$ observations (visual features within each tracklet) extracted from video $V$, where observation $O_i$ captures appearance features $s_i(t)$, such as histograms of oriented gradients (HoG by *Dalal and Triggs* (2005)), and spatio-temporal features $u_i(t)$, such as a bag of video words (BoV by *Dollar et al.* (2005)). $t$ corresponds to a specific time stamp within the set of frames $\mathcal{T}_V = (t_1, t_2, ..., t_Z)$ of video $V$, where $Z$ is the total number of frames in $V$. Each observation $O_i$ can be seen as a realization of the underlying atomic activity $A_i$ of an individual. Let $A = (A_1, A_2, ..., A_N)$. $A_i$ includes pose labels $p_i(t) \in \mathcal{P}$, and action class labels $a_i(t) \in \mathcal{A}$ at time $t \in \mathcal{T}_V$. $\mathcal{P}$ and $\mathcal{A}$ denote the set of all possible pose (e.g, *facing-front*) and action (e.g, *walking*) labels, respectively. $I = (I_{12}, I_{13}, ..., I_{N-1N})$ denotes the interactions between all possible (coexisting) pairs of $A_i$ and $A_j$, where each $I_{ij} = (I_{ij}(t_1), ...I_{ij}(t_Z))$ and $I_{ij}(t) \in \mathcal{I}$

is the set of interaction labels such as *approaching*, *facing-each-other* and *standing-in-a-row*. Similarly, $C = (C(t_1), ..., C(t_Z))$ and $C(t_i) \in \mathcal{C}$ indicates the collective activity labels of the video $V$, where $\mathcal{C}$ is the set of collective activity labels, such as *gathering*, *queueing*, and *talking*. In this chapter, we assume there exists only one collective activity at a certain time frame. Extensions to modelling multiple collective activities will be addressed in the future. $T$ describes the target (tracklet) associations in the scene as explained in Sec.4.2.

We formulate the classification problem in an energy maximization framework (*LeCun et al.* (2006)), with overall energy function $\Psi(C, I, A, O, T)$. The energy function is modelled as the linear product of model weights $w$ and the feature vector $\psi$ :

$$\Psi(C, I, A, O, T) = w^T \psi(C, I, A, O, T) \tag{4.1}$$

$\psi(C, I, A, O, T)$ is a vector composed of $\psi_1(\cdot), \psi_2(\cdot), ..., \psi_m(\cdot)$ where each feature element encodes local relationships between variables and $w$, which is learned discriminatively, is the set of model parameters. High energy potentials are associated with configurations of $A$ and $I$ that tend to co-occur in training videos with the same collective activity $C$. For instance, the *talking* collective activity tends to be characterized by interaction activities such as *greeting*, *facing-each-other* and *standing-side-by-side*, as shown in Fig.4.2.

### 4.2.2 Model characteristics

The central idea of our model is that the atomic activities of individuals are highly correlated with the overall collective activity, through the interactions between people. This hierarchy is illustrated in Fig.4.1. Assuming the conditional independence implied in our undirected graphical model, the overall energy function can be decomposed as a summation of seven local potentials: $\Psi(C, I)$, $\Psi(C, O)$, $\Psi(I, A, T)$,

$\Psi(A,O)$, $\Psi(C)$, $\Psi(I)$, and $\Psi(A)$. The overall energy function can easily be represented as in Eq.4.1 by rearranging the potentials and concatenating the feature elements to construct the feature vector $\psi$. Each local potential corresponds to a node (in the case of unitary terms), an edge (in the case of pairwise terms), or a high order potential seen on the graph in Fig.4.1.(c): 1) $\Psi(C,I)$ encodes the correlation between collective activities and interactions (Fig.4.2.(b)). 2) $\Psi(I,A,T)$ models the correlation between interactions and atomic activities (Fig.4.2.(a)). 3) $\Psi(C)$, $\Psi(I)$ and $\Psi(A)$ encode the temporal smoothness prior in each of the variables. 4) $\Psi(C,O)$ and $\Psi(A,O)$ model the compatibility of the observations with the collective activity and atomic activities, respectively.

**Collective - Interaction** $\Psi(C,I)$: The function is formulated as a linear multi-class model (*Weston and Watkins* (1998)):

$$\Psi(C,I) = \sum_{t\in\mathcal{T}_V} \sum_{a\in\mathcal{C}} w_{ci}^{a} \cdot h(I,t;\triangle t_C)\mathbb{I}(a,C(t)) \tag{4.2}$$

where $w_i$ is the vector of model weights for each class of collective activity, $h(I,t;\triangle t_C)$ is an $\mathcal{I}$ dimensional histogram function of interaction labels around time $t$ (within a temporal window $\pm\triangle t_C$), and $\mathbb{I}(\cdot,\cdot)$ is an indicator function, that returns 1 if the two inputs are the same and 0 otherwise.

**Collective Activity Transition** $\Psi(C)$: This potential models the temporal smoothness of collective activities across adjacent frames. That is,

$$\Psi(C) = \sum_{t\in\mathcal{T}_V} \sum_{a\in\mathcal{C}} \sum_{b\in\mathcal{C}} w_{c}^{ab}\ \mathbb{I}(a,C(t))\ \mathbb{I}(b,C(t+1)) \tag{4.3}$$

**Interaction Transition** $\Psi(I) = \sum_{i,j}\Psi(I_{ij})$: This potential models the temporal smoothness of interactions across adjacent frames. That is,

$$\Psi(I_{ij}) = \sum_{t\in\mathcal{T}_V} \sum_{a\in\mathcal{I}} \sum_{b\in\mathcal{I}} w_{i}^{ab}\ \mathbb{I}(a,I_{ij}(t))\ \mathbb{I}(b,I_{ij}(t+1)) \tag{4.4}$$

**Interaction - Atomic** $\Psi(I, A, T) = \sum_{i,j} \Psi(A_i, A_j, I_{ij}, T)$: This encodes the correlation between the interaction $I_{ij}$ and the relative motion between two atomic motions $A_i$ and $A_j$ given all target associations $T$ (more precisely the trajectories of $T_k$ and $T_l$ to which $\tau_i$ and $\tau_j$ belong, respectively). The relative motion is encoded by the feature vector $\psi$ and the potential $\Psi(A_i, A_j, I_{ij}, T)$ is modelled as:

$$\Psi(A_i, A_j, I_{ij}, T) = \sum_{t \in \mathcal{T}_V} \sum_{a \in \mathcal{I}} w_{ai}^a \cdot \psi(A_i, A_j, T, t; \triangle t_I)\ \mathbb{I}(a, I_{ij}) \tag{4.5}$$

where $\psi(A_i, A_j, T, t; \triangle t_I)$ is a vector representing the relative motion between two targets within a temporal window $(t - \triangle t_I,\ t + \triangle t_I)$ and $w_{ai}^a$ is the model parameter for each class of interaction. The feature vector is designed to encode the relationships between the locations, poses, and actions of two people. See Appendix.A for details. Note that since this potential incorporates information about the location of each target, it is closely related to the problem of target association. The same potential is used in both the activity classification and the multi-target tracking components of our framework.

**Atomic Prior** $\Psi(A)$: Assuming independence between pose and action, the function is modelled as a linear sum of pose transition $\Psi_p(A)$ and action transition $\Psi_a(A)$. This potential function is composed of two functions that encode the smoothness of pose and action. Each of them is parameterized as the co-occurrence frequency of the pair of variables similar to $\Psi(I_{ij})$.

**Observations** $\Psi(A, O) = \sum_i \Psi(A_i, O_i)$ and $\Psi(C, O)$: these model the compatibility of atomic $(A)$ and collective $(C)$ activity with observations $(O)$. Details of the features are explained in Sec.4.6.

## 4.3 Multiple Target Tracking

Our multi-target tracking formulation follows the philosophy of *Singh et al.* (2008), where tracks are obtained by associating corresponding tracklets. Unlike other methods, we leverage the contextual information provided by interaction activities to make target association more robust. Here, we assume that a set of initial tracklets, atomic activities, and interaction activities are given. We will discuss the joint estimation of these labels in Sec.4.4.

As shown in Fig.4.3, tracklet association can be formulated as a min-cost network problem (*Zhang et al.* (2008)), where the edge between a pair of nodes represents a tracklet, and the black directed edges represent possible links to match two tracklets. We refer the reader to *Zhang et al.* (2008); *Pirsiavash et al.* (2011) for the details of network-flow formulations.

Given a set of tracklets $\tau_1, \tau_2, ..., \tau_N$ where $\tau_i = \{x_{\tau_i}(t_0^i), ..., x_{\tau_i}(t_e^i)\}$ and $x(t)$ is a position at $t$, the tracklet association problem can be stated as that of finding an unknown number $M$ of associations $T_1, T_2, ..., T_M$, where each $T_i$ contains one or more indices of tracklets. For example, one association may consist of tracklets 1 and 3: $T_1 = \{1, 3\}$. To accomplish this, we find a set of possible paths between two non-overlapping tracklets $\tau_i$ and $\tau_j$. These correspond to match hypotheses $p_{ij}^k = \{x_{p_{ij}^k}(t_e^i + 1), ..., x_{p_{ij}^k}(t_0^j - 1)\}$ where the timestamps are in the temporal gap between $\tau_i$ and $\tau_j$. The association $T_i$ can be redefined by augmenting the associated pair of tracklets $\tau_i$ and $\tau_j$ with the match hypothesis $p_{ij}$. For example, $T_1 = \{1, 3, 1\text{-}2\text{-}3\}$ indicates that tracklet 1 and 3 form one track and the second match hypothesis (the solid edge between $\tau_1$ and $\tau_3$ in Fig. 4.3) connects them. Given human detections, we can generate match hypotheses using the K-shortest path algorithm (*Yen*). Please see Appendix.A for details.

Each match hypothesis has an associated cost value $c_{ij}^k$ that represents the validity of the match. This cost is derived from detection responses, motion cues, and color

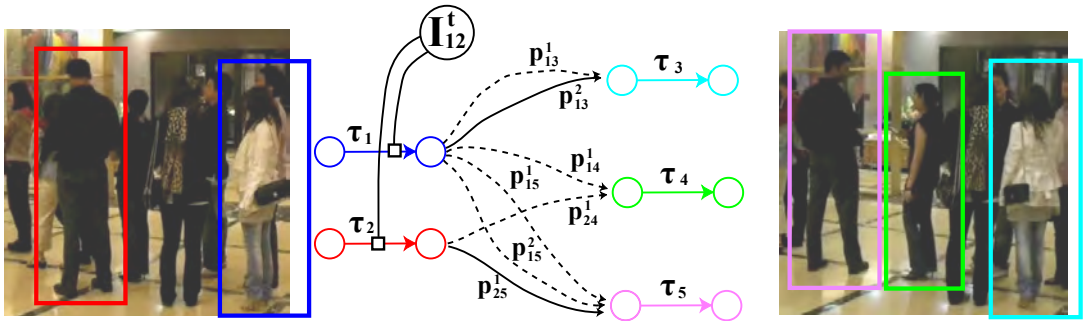



Figure 4.3: The tracklet association problem is formulated as a min-cost flow network (*Zhang et al.* (2008); *Pirsiavash et al.* (2011)). The network graph is composed of two components: tracklets $\tau$ and path proposals $p$. In addition to these two, we incorporate interaction potential to add robustness in tracklet association. In this example, the interaction "standing-in-a-row" helps reinforce the association between tracklets $\tau_1$ and $\tau_3$ and penalizes the association between $\tau_1$ and $\tau_4$.

similarity. By limiting the number of hypotheses to a relatively small value of $K$, we prune out a majority of the exponentially many hypotheses that could be generated by raw detections. If we define the cost of entering and exiting a tracklet as $c_{en}$ and $c_{ex}$ respectively, the tracklet association problem can be written as :

$$\hat{f} = \underset{f}{\operatorname{argmin}}\, c^T f = \underset{f}{\operatorname{argmin}} \sum_i c_{en} f_{en,i} + \sum_i c_{ex} f_{i,ex} + \sum_{i,j} \sum_k c_{ij}^k f_{ij}^k$$

$$s.t.\ f_{en,i}, f_{i,ex}, f_{ij}^k \in \{0,1\},\ f_{en,i} + \sum_j \sum_k f_{ji}^k = f_{i,ex} + \sum_j \sum_k f_{ij}^k = 1$$

where $f$ represent the flow variables, the first set of constraints is a set of binary constraints and the second one captures the inflow-outflow constraints (we assume all the tracklets are true). Later in this chapter, we will refer to $\mathbb{S}$ as the feasible set for $f$ that satisfies the above constraints. Once the flow variable $f$ is specified, it is trivial to obtain the tracklet association $T$ through a mapping function $T(f)$. The above problem can be efficiently solved by binary integer programming, since it involves only a few variables, with complexity $O(KN)$ where $N$ (the number of tracklets) is typically a few hundred, and there are $2N$ equality constraints. Note

that the number of nodes in *Zhang et al.* (2008); *Pirsiavash et al.* (2011) is usually in the order of tens or hundreds of thousands.

One of the novelties of our framework lies in the contextual information that comes from the interaction activity nodes. For the moment, assume that the interactions $I_{12}^t$ between $A_1$ and $A_2$ are known. Then, selecting a match hypothesis $f_{ij}^k$ should be related with the likelihood of observing the interaction $I_{12}^t$. For instance, the *red* and *blue* targets in Fig.4.3 are engaged in the *standing-in-a-row* interaction activity. If we select the match hypothesis that links *red* with *pink* and *blue* with *sky-blue* (shown with solid edges), then the interaction will be compatible with the links, since the distance between *red* and *blue* is similar to that between *pink*/*sky-blue*. However, if we select the match hypothesis that links *red* with *green*, this will be less compatible with the *standing-in-a-row* interaction activity, because the *green*/*pink* distance is less than the *red*/*blue* distance, and people do not tend to move toward each other when they are in a queue. The potential $\Psi(I, A, T)$ (Sec.4.2.2) is used to enforce this consistency between interactions and tracklet associations.

## 4.4 Unifying activity classification and tracklet association

The previous two sections present collective activity classification and multi-target tracking as independent problems. In this section, we show how they can be modelled in a unified framework. Let $\hat{y}$ denote the desired solution of our unified problem. The optimization can be written as:

$$\hat{y} = \underset{f,C,I,A}{\operatorname{argmax}} \underbrace{\Psi(C, I, A, O, T(f))}_{Sec.4.2} - \underbrace{c^T f}_{Sec.4.3}, \ s.t.\ f \in \mathbb{S} \tag{4.6}$$

where $f$ is the binary flow variables, $\mathbb{S}$ is the feasible set of $f$, and $C, I, A$ are activity variables. As noted in the previous section, the interaction potential $\Psi(A, I, T)$ involves the variables related to both activity classification ($A$, $I$) and tracklet as-

sociation ($T$). Thus, changing the configuration of interaction and atomic variables affects not only the energy of the classification problem, but also the energy of the association problem. In other words, our model is capable of propagating the information obtained from collective activity classification to target association and from target association to collective activity classification through $\Psi(A, I, T)$.

### 4.4.1 Inference

Since the interaction labels $I$ and the atomic activity labels $A$ guide the flow of information between target association and activity classification, we leverage the structure of our model to efficiently solve this complicated joint inference problem. The optimization problem Eq.4.6 is divided into two sub problems and solved iteratively:

$$\{\hat{C}, \hat{I}, \hat{A}\} = \underset{C,I,A}{\operatorname{argmax}} \Psi(C, I, A, O, T(\hat{f})) \tag{4.7}$$

$$\hat{f} = \underset{f}{\operatorname{argmin}}\, c^T f - \Psi(\hat{I}, \hat{A}, T(f)),\ s.t.\ f \in \mathbb{S} \tag{4.8}$$

Given $\hat{f}$ (and thus $\hat{T}$) the hierarchical classification problem is solved by applying iterative Belief Propagation. Fixing the activity labels $A$ and $I$, we solve the target association problem by applying the Branch-and-Bound algorithm with a tight linear

**Algorithm 3** Iterative Belief Propagation

**Require:** Given association $\hat{T}$ and observation $O$.
  Initialize $C^0, I^0, A^0$
  **while** Convergence, k++ **do**
    $C^k \Leftarrow \text{argmax}_C\, \Psi(C, I^{k-1}, A^{k-1}, O, \hat{T})$
    **for all** $\forall i \in A$ **do**
      $A_i^k \Leftarrow \text{argmax}_A\, \Psi(C^k, I^{k-1}, A, A_{\setminus i}^{k-1}, O, \hat{T})$
    **end for**
    **for all** $\forall i \in I$ **do**
      $I_i^k \Leftarrow \text{argmax}_I\, \Psi(C^k, I, I_{\setminus i}^{k-1}, A^k, O, \hat{T})$
    **end for**
  **end while**

lower bound (see below for more details).

**Iterative Belief Propagation.** Due to the high order potentials in our model (such as the Collective-Interaction potential), the exact inference of the all variables is intractable. Thus, we propose an approximate inference algorithm that takes advantage of the structure of our model. Since each type of variable forms a simple chain in the temporal direction (see Fig.4.1), it is possible to obtain the optimal solution given all the other variables by using belief propagation (*Felzenszwalb and Huttenlocher* (2006)). The iterative belief propagation algorithm is grounded in this intuition, and is shown in detail in Alg.4.

**Target Association Algorithm.** We solve the association problem by using the Branch-and-Bound method. Unlike the original min-cost flow network problem, the interaction terms introduce a quadratic relationship between flow variables. Note that we need to choose at most two flow variables to specify one interaction feature. For instance, if there exist two different tails of tracklets at the same time stamp, we need to specify two of the flows out of seven flows to compute the interaction potential as shown in Fig.4.3. This leads to a non-convex binary quadratic programming problem which is hard to solve exactly (the Hessian $H$ is not a positive semi-definite matrix).

$$\underset{f}{\operatorname{argmin}} \frac{1}{2} f^T H f + c^T f, \ s.t. \ f \in \mathbb{S} \tag{4.9}$$

To tackle this issue, we use a Branch-and-Bound (BB) algorithm with a novel tight lower bound function given by $h^T f \leq \frac{1}{2} f^T H f, \ \forall f \in \mathbb{S}$. See Appendix.A for details about variable selection, lower and upper bounds, and definitions of the BB algorithm.

## 4.5 Model Learning

Given the training videos, the model is learned in a two-stage process: i) learning the observation potentials $\Psi(A, O)$ and $\Psi(C, O)$. This is done by learning each ob-

servation potential $\Psi(\cdot)$ independently using multiclass SVM (*Weston and Watkins* (1998)). ii) learning the model weights $w$ for the full model in a max-margin framework as follows. Given a set of $N$ training videos $(x^n, y^n)$, $n = 1, ..., N$, where $x^n$ is the observations from each video and $y^n$ is a set of labels, we train the global weight $w$ in a max-margin framework. Specifically, we employ the cutting plane training algorithm described in *Joachims et al.* (2009) to solve this optimization problem. We incorporate the inference algorithm described in Sec.4.4.1 to obtain the most violated constraint in each iteration (*Joachims et al.* (2009)). To improve computational efficiency, we train the model weights related to activity potentials first, and train the model weights related to tracklet association using the learnt activity models.

## 4.6 Experimental Validation

**Implementation details.** Our algorithm assumes that the inputs $O$ are available. These inputs are composed of collective activity features, tracklets, appearance feature, and spatio-temporal features as discussed in Sec.4.2.1. Given a video, we obtain tracklets using a proper tracking method (see text below for details). Once tracklets $O$ are obtained, we compute two visual features (the histogram of oriented gradients (HoG) decriptors by *Dalal and Triggs* (2005) and the bag of video words (BoV) histogram by *Dollar et al.* (2005)) in order to classify poses and actions, respectively. The HoG is extracted from an image region within the bounding box of the tracklets and the BoV is constructed by computing the histogram of video-words within the spatio-temporal volume of each tracklet. To obtain the video-words, we apply PCA (with 200 dimensions) and the k-means algorithm (100 codewords) on the cuboids obtained by *Dollar et al.* (2005). Finally, the collective activity features are computed using the STL descriptor (chapter III) on tracklets and pose classification estimates. We adopt the same parameters used in the chapter III for STL construction (8 meters for maximum radius and 60 frames for the temporal support).

Since we are interested in labelling one collective activity per one time slice (i.e. a set of adjacent time frames), we take the average of all collected STL in the same time slice to generate an observation for $C$. In addition, we append the mean of the HoG descriptors obtained from all people in the scene to encode the shape of people in a certain activity. Instead of directly using raw features from HoG, BoV, and STL, we train multiclass SVM classifiers (*Joachims et al.* (2009)) for each of the observations to keep the size of parameters within a reasonable bound. In the end, each of the observation features is represented as a $|\mathcal{P}|$, $|\mathcal{A}|$, and $|\mathcal{C}|$ dimensional features, where each dimension of the features is the classification score given by the SVM classifier. In the experiments, we use the SVM response for $C$ as a baseline method (Tab.4.1 and Fig.4.4).

Given tracklets and associated pose/action features $O$, a temporal sequence of atomic activity variables $A_i$ is assigned to each tracklet $\tau_i$. For each pair of coexisting $A_i$ and $A_j$, $I_{ij}$ describes the interaction between the two. Since $I$ is defined over a certain temporal support ($\triangle t_I$), we sub-sample every 10th frames to assign an interaction variable. Finally, one $C$ variable is assigned in every 20 frames with a temporal support $\triangle t_C$. We present experimental results using different choices of $\triangle t_I$ and $\triangle t_C$, (Tab.4.2). Given tracklets and observations ($O$ and $O_C$), the classification and target association take about a minute per video in our experiments.

| | Collective Activity Dataset | | | | New Dataset | | | |
|---|---|---|---|---|---|---|---|---|
| Method | Ovral ($C$) | Mean ($C$) | Ovral ($I$) | Mean ($I$) | Ovral ($C$) | Mean ($C$) | Ovral ($I$) | Mean ($I$) |
| without $O_C$ | 38.7 | 37.1 | 40.5 | 37.3 | 59.2 | 57.4 | 49.4 | 41.1 |
| no edges between $C$ and $I$ | 67.7 | 68.2 | 42.8 | 37.7 | 67.8 | 54.6 | 42.4 | 32.8 |
| no temporal chain | 66.9 | 66.3 | 42.6 | 33.7 | 71.1 | 68.9 | 41.9 | 46.1 |
| no temporal chain between $C$ | 74.1 | 75.0 | 54.2 | 48.6 | 77.0 | 76.1 | **55.9** | **48.6** |
| full model ($\triangle t_C = 20, \triangle t_I = 25$) | **79.0** | **79.6** | **56.2** | **50.8** | **83.0** | **79.2** | 53.3 | 43.7 |
| baseline | 72.5 | 73.3 | - | - | 77.4 | 74.3 | - | - |

Table 4.1: Comparison of collective and interaction activity classification for different versions of our model using the collective activity dataset (left column) and the newly proposed dataset (right column). The models we compare here are: i) *Graph without* $O_C$. We remove observations (STL) for the collective activity. ii) *Graph with no edges between* $C$ *and* $I$. We cut the connections between variables $C$ and $I$ and produce separate chain structures for each set of variables. iii) *Graph with no temporal edges.* We cut all the temporal edges between variables in the graphical structure and leave only hierarchical relationships. iv) *Graph with no temporal chain between* $C$ *variables.* v) Our full model shown in Fig.4.1.(d) and vi) baseline method. The baseline method is obtained by taking the max response from the collective activity observation ($O_C$).

| | Collective Activity Dataset | | | | New Dataset | | | |
|---|---|---|---|---|---|---|---|---|
| Method | Ovral ($C$) | Mean ($C$) | Ovral ($I$) | Mean ($I$) | Ovral ($C$) | Mean ($C$) | Ovral ($I$) | Mean ($I$) |
| $\triangle t_C = 30, \triangle t_I = 25$ | 79.1 | 79.9 | 56.1 | 50.8 | 80.8 | 77.0 | **54.3** | **46.3** |
| $\triangle t_C = 20, \triangle t_I = 25$ | 79.0 | 79.6 | **56.2** | **50.8** | **83.0** | **79.2** | 53.3 | 43.7 |
| $\triangle t_C = 10, \triangle t_I = 25$ | 77.4 | 78.2 | 56.1 | 50.7 | 81.5 | 77.6 | 52.9 | 41.8 |
| $\triangle t_C = 30, \triangle t_I = 15$ | 76.1 | 76.7 | 52.8 | 40.7 | 80.7 | 71.8 | 48.6 | 34.8 |
| $\triangle t_C = 30, \triangle t_I = 5$ | **79.4** | **80.2** | 45.5 | 36.6 | 77.0 | 67.3 | 37.7 | 25.7 |

Table 4.2: Comparison of classification results using different lengths of temporal support $\triangle t_C$ and $\triangle t_I$ for collective and interaction activities, respectively. Notice that in general larger support provides more stable results.

**Datasets and experimental setup.** We present experimental results on the collective activity dataset introduced in the chapter III and a newly proposed dataset. The first dataset is composed of 44 video clips with annotations for 5 collective activities (*crossing*, *waiting*, *queuing*, *walking*, and *talking*) and 8 poses (*right*, *right-front*, ..., *right-back*). In addition to these labels, we annotate the target correspondence, action labels and interaction labels for all sequences. We define the 8 types of interactions as *approaching* (AP), *leaving* (LV), *passing-by* (PB), *facing-each-other* (FE), *walking-side-by-side* (WS), *standing-in-a-row* (SR), *standing-side-by-side* (SS) and *no-interaction* (NA). The categories of atomic actions are defined as: *standing* and *walking*. Due to a lack of standard experimental protocol on this dataset, we adopt two experimental scenarios. First, we divide the whole set into 4 subsets without overlap of videos and perform 4-fold training and testing. Second, we divide the set into separate training and testing sets as suggested by *Lan et al.* (2010b). Since the first setup provides more data to be analysed, we run the main analysis with the setup and use the second for comparison against *Lan et al.* (2010b).

The second dataset is composed of 32 video clips with 6 collective activities: *gathering*, *talking*, *dismissal*, *walking together*, *chasing*, *queueing*. For this dataset, we define 9 interaction labels: *approaching* (AP), *walking-in-opposite-direction* (WO), *facing-each-other* (FE), *standing-in-a-row* (SR), *walking-side-by-side* (WS), *walking-one-after-the-other* (WR), *running-side-by-side* (RS), *runn ing-one-after-the-other* (RR), and *no-interaction* (NA). The atomic actions are labelled as *walking*, *standing still*, and *running*. We define 8 poses similarly to the first dataset. We divide the whole set into 3 subsets and run 3-fold training and testing. For this dataset, we obtain the tracklets using the method by *Pirsiavash et al.* (2011) and create back projected 3D trajectories using the simplified camera model (*Hoiem et al.* (2008)).

**Results and Analysis.**

We analyze the behavior of the proposed model by disabling the connectivity be-

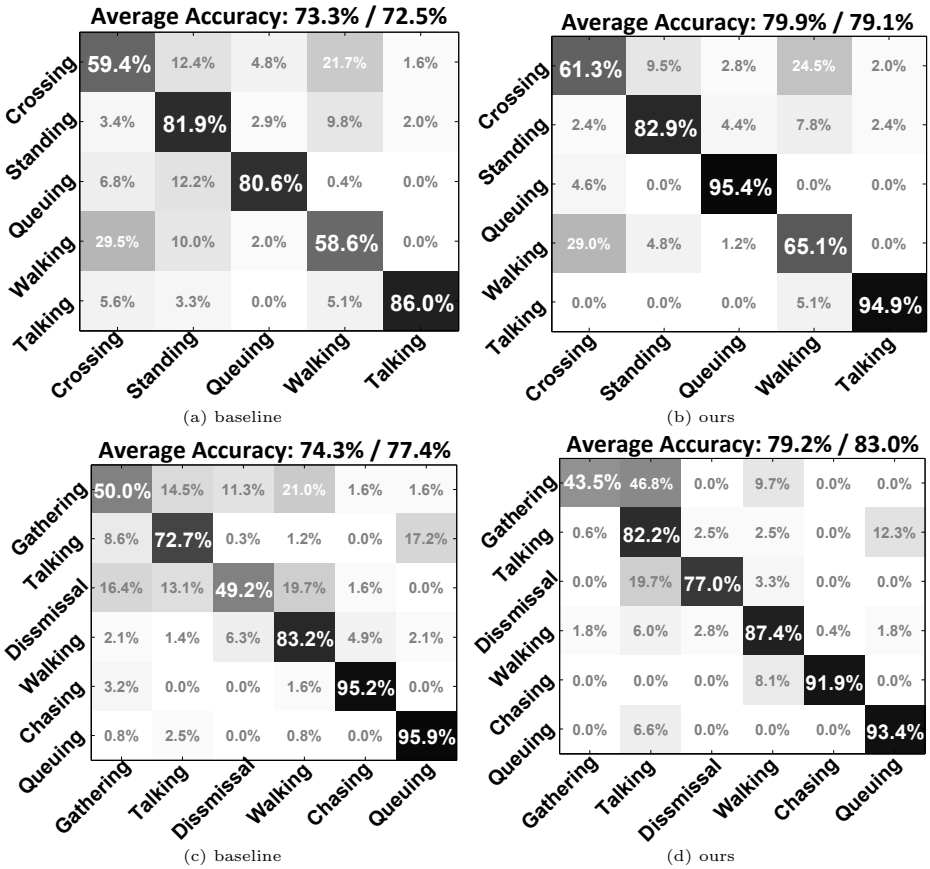



Figure 4.4: (a) and (b) shows the confusion table for collective activity using baseline method (SVM response for $C$) and proposed method on the collective activity dataset, respectively. (c) and (d) compare the two methods on newly proposed dataset. In both cases, our full model improves the accuracy significantly over the baseline method. The numbers on top of each table show *mean-per-class* and *overall* accuracies.

tween various variables of the graphical structure (see Tab.4.1 and Fig.4.4 for details). We study the classification accuracy of collective activities $C$ and interaction activities $I$. As seen in the Tab.4.1, the best classification results are obtained by our full model. Since the dataset is unbalanced, we present both overall accuracy and mean-per-class accuracy, denoted as Ovral and Mean in Tab.4.1 and Tab.4.2.

Next, we analyse the model by varying the parameter values that define the temporal supports of collective and interaction activities ($\triangle t_C$ and $\triangle t_I$). We run different experiments by fixing one of the temporal supports to a reference value and change the other. As any of the temporal supports becomes larger, the collective and in-

teraction activity variables are connected with a larger number of interactions and atomic activity variables, respectively, which provides richer coupling between variables across labels of the hierarchy and, in turn, enables more robust classification results (Tab.4.2). Notice that, however, by increasing connectivity, the graphical structure becomes more complex and thus inference becomes less manageable.

Since previous works adopt different ways of calculating the accuracy of the collective activity classification, a direct comparison of the results may not be appropriate. STL and RSTV (described in the chapter III) adopt leave-one-video-out training/testing and evaluate per-person collective activity classification. *Lan et al.* (2010b) train their model on three fourths of the dataset, test on the remaining fourth and evaluate per-scene collective activity classification. To compare against the results obtained in the chapter III, we assign the per-scene collective activity labels that we obtain with four-fold experiments to each individual. We obtain an accuracy of 74.4% which is superior than 65.9% and 70.9% reported in the chapter III. In addition, we run the experiments on the same training/testing split of the dataset suggested by *Lan et al.* (2010b) and achieve competitive accuracy (80.4% overall and 75.7% mean-per-class compared to 79.1% overall and 77.5% mean-per-class, respectively, reported in *Lan et al.* (2010b)). Anecdotal results are shown in the Fig.B.2.

Tab.4.3 summarizes the tracklet association accuracy of our method. The association accuracy is measured using the Match Error Correction Rate (MECR) $\frac{\#\ error\ in\ tracklet\ -\ \#\ error\ in\ result}{\#\ error\ in\ tracklet}$. In this experiment, we test three different algorithms for tracklet matching : pure match, linear model, and full quadratic model. *Match* represents the max-flow method without interaction potential (only appearance, motion and detection scores are used). *Linear* model represents our model where the quadratic relationship is ignored and only the linear part of the interaction potentials is considered (e.g. those interactions that are involved in selecting only one path). The *Quadratic* model represents our full Branch-and-Bound method for target

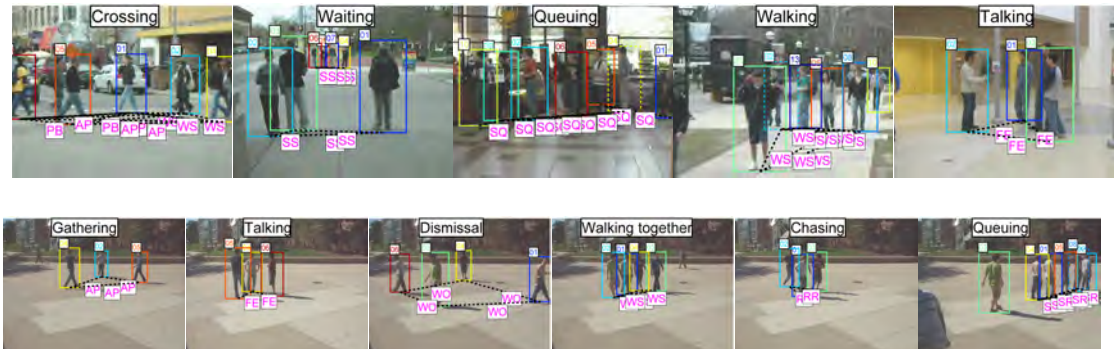



Figure 4.5: Anecdotal results on different types of collective activities. In each image, we show the collective activity estimated by our method. Interactions between people are denoted by the dotted line that connects each pair of people. To make the visualization more clear, we only show interactions that are not labeled as NA (*no interaction*). Anecdotal results on the collective activity dataset and the newly proposed dataset are shown on the top and bottom rows, respectively. Our method automatically discovers the interactions occurring within each collective activity; e.g. *walking-side-by-side* (denoted as WS) occurs with *crossing* or *walking*, whereas *standing-side-by-side* (SS) occurs with *waiting*. See text for the definition of other acronyms.

association. The estimated activity labels are assigned to each variable for the two methods. We also show the accuracy of association when ground truth (GT) activity labels are provided, in the fourth and fifth columns of the table. The last column shows the number of association errors in the initial input tracklets. In these experiments, we adopt the same four fold training/testing and three fold training/testing for the collective activity dataset and newly proposed dataset, respectively. Note that, in the collective activity dataset, there exist 1821 tracklets with 1556 match errors in total. In the new dataset, which includes much less crowded sequences than the former, there exist 474 tracklets with 604 errors in total. As the Tab.4.3 shows, we achieve significant improvement over baseline method (*Match*) using the collective activity dataset as it is more challenging and involves a large number of people (more information from interactions). On the other hand, we observe a smaller improvement in matching targets in the second dataset, since it involves few people (typically $2 \sim 3$)

and is less challenging (note that the baseline (*Match*) already achieves 81% correct match). Experimental results obtained with ground truth activity labels (*Linear GT* and *Quad. GT*) suggest that better activity recognition would yield more accurate tracklet association. Anecdotal results are shown in Fig.4.6.


(a) (b)

Figure 4.6: The discovered interaction *standing-side-by-side* (denoted as SS) helps to keep the identity of tracked individuals after an occlusion. Notice the complexity of the association problem in this example. Due to the proximity of the targets and similarity in color, the *Match* method **(b)** fails to keep the identity of targets. However, our method **(a)** finds the correct match despite the challenges. The input tracklets are shown as a solid box and associated paths are shown in dotted box.



| | *Match* | *Linear* | *Quadratic* | *Linear GT* | *Quad. GT* | *Tracklet* |
|---|---|---|---|---|---|---|
| Collective Activity Dataset | 1109/28.73% | 974/37.40% | 894/42.54% | 870/44.09% | 736/52.70% | 1556/0% |
| New Dataset | 110/81.79% | 107/82.28% | 104/82.78% | 97/83.94% | 95/84.27% | 604/0% |

Table 4.3: Quantitative tracking results and comparison with baseline methods (see text for definitions). Each cell of the table shows the number of match errors and Match Error Correction Rate (MECR) of each method, respectively. Since we focus on correctly associating each tracklet with another, we evaluate the method by counting the number of errors made during association (rather than detection-based accuracy measurements such as recall, FPPI, etc) and MECR. An association error is defined for each possible match of a tracklet (thus at most two per tracklets, previous and next match). This measure can effectively capture the amount of fragmentization and identity switches in association. In the case of a false alarm tracklet, any association with this track is considered to be an error.

## 4.7 Conclusion

In this chapter, we present a new framework to coherently identify target associations and classify collective activities. We demonstrate that collective activities provide critical contextual cues for making target association more robust and stable; in turn, the estimated trajectories as well as atomic activity labels allow the construction of more accurate interaction and collective activity models.

# CHAPTER V

# Understanding Indoor Scenes using 3D Geometric Phrases

Consider the scene in Fig. 5.1.(a). A scene classifier will tell you, with some uncertainty, that this is a dining room ( *Pandey and Lazebnik* (2011); *Quattoni and Torralba* (2009); *Lazebnik et al.* (2006); *Fei-Fei and Perona* (2005)). A layout estimator (*Hoiem et al.* (2007); *Hedau et al.* (2009); *Lee et al.* (2009); *Wang et al.* (2010)) will tell you, with different uncertainty, how to fit a box to the room. An object detector (*Leibe et al.* (2004); *Dalal and Triggs* (2005); *Felzenszwalb et al.* (2010); *Xiang and Savarese* (2012)) will tell you, with large uncertainty, that there is a dining table and four chairs. Each algorithm provides important but uncertain and incomplete piece of information. This is because the scene is cluttered with objects which tend to occlude each other: the dining table occludes the chairs, the chairs occlude the dining table; all of these occlude the room layout components (i.e. the walls and floor).

It is clear that truly understanding a scene involves integrating information at multiple levels as well as studying the interactions between scene elements. A scene-object interaction describes the way a scene type (e.g. a dining room or a bedroom) influences objects' presence, and vice versa. An object-layout interaction describes the way the layout (e.g. the 3D configuration of walls, floor and observer's pose) biases

This chapter is based on the publications (*Choi et al.* (2013a)).

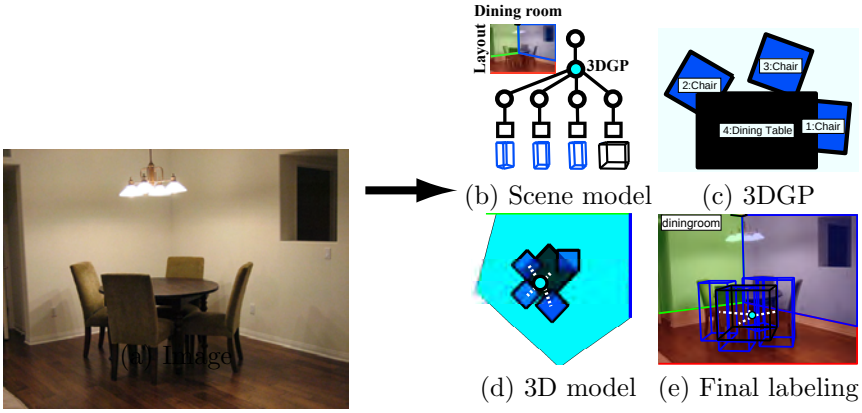



Figure 5.1: Our unified model combines object detection, layout estimation and scene classification. A single input image (a) is described by a scene model (b), with the scene type and layout at the root, and objects as leaves. The middle nodes are latent *3D Geometric Phrases*, such as (c), describing the 3D relationships among objects (d). Scene understanding means finding the correct parse graph, producing a final labeling (e) of the objects in 3D (bounding cubes), the object groups (dashed white lines), the room layout, and the scene type.

the placement of objects in the image, and vice versa. An object-object interaction describes the way objects and their pose affect each other (e.g. a dining table suggests that a set of chairs are to be found around it). Combining predictions at multiple levels into a global estimate can improve each individual prediction. As part of a larger system, understanding a scene semantically and functionally will allow us to make predictions about the presence and locations of unseen objects within the space.

We propose a method that can automatically learn the interactions among scene elements and apply them to the holistic understanding of indoor scenes. This scene interpretation is performed within a hierarchical interaction model and derived from a single image. The model fuses together object detection, layout estimation and scene classification to obtain a unified estimate of the scene composition. The problem is formulated as image parsing in which a parse graph must be constructed for an image as in Fig. 5.1.(b). At the root of the parse graph is the scene type and layout while the leaves are the individual detections of objects. In between is the core of the system, our novel *3D Geometric Phrases* (3DGP) (Fig. 5.1.(c)).

A 3DGP encodes geometric and semantic relationships between groups of objects which frequently co-occur in spatially consistent configurations. As opposed to previous approaches such as *Desai et al.* (2011); *Sadeghi and Farhadi* (2011), the 3DGP is defined using 3D spatial information, making the model rotation and viewpoint invariant. Grouping objects together provides contextual support to boost weak object detections, such as the chair that is occluded by the dining table.

Training this model involves both discovering a set of 3DGPs and estimating the parameters of the model. We present a new learning scheme which discovers 3DGPs in an unsupervised manner, avoiding expensive and ambiguous manual annotation. This allows us to extract a few useful sets of GPs among exponentially many possible configurations. Once a set of 3DGPs is selected, the model parameters can be learned in a max-margin framework. Given the interdependency between the 3DGPs and the model parameters, the learning process is performed iteratively (Sec. 5.4).

To explain a new image, a parse graph must estimate the scene semantics, layout, objects and 3DGPs, making the space of possible graphs quite large and of variable dimension. To efficiently search this space during inference, we present a novel combination of bottom-up clustering with top-down Reversible Jump Markov Chain Monte Carlo (RJ-MCMC) sampling (Sec. 5.3).

As a result of the rich contextual relationships captured by our model, it can provide scene interpretations from a single image in which i) objects and space interact in a physically valid way, ii) objects occur in an appropriate scene type, iii) the object set is self-consistent and iv) configurations of objects are automatically discovered (Fig. 5.1.(d,e)). We quantitatively evaluate our model on a novel challenging dataset, the *indoor-scene-object* dataset. Experiments show our hierarchical scene model constructed upon 3DGPs improves object detection, layout estimation and semantic classification accuracy in challenging scenarios which include occlusions, clutter and intra-class variation.

## 5.1 Background

Image understanding has been explored on many levels, including object detection, scene classification and geometry estimation.

The performance of generic object recognition has improved recently thanks to the introduction of more powerful feature representations (*Lowe* (2004); *Dalal and Triggs* (2005)). *Felzenszwalb et al.* (2010) proposed a deformable part model (DPM) composed of multiple HoG components which shows promising performance for single objects. To improve detection robustness, the interactions between objects can be modeled. Category-specific 2D spatial interactions have been modeled via contextual features by *Desai et al.* (2011), whereas *Sadeghi and Farhadi* (2011) model groups of objects as *visual phrases* in 2D image space that are determined by a domain expert. *Li et al.* (2012) identify a set of useful *visual phrases* from a training set using only 2D spatial consistency. Improving upon these, *Desai et al.* (2011) propose a method that can encode detailed pose relationships between co-appearing objects in 2D image space. In contrast to these approaches, our 3DGPs are capable of encoding both 3D geometric and contextual interactions among objects and can be automatically learned from training data.

Researchers have also looked at the geometric configuration of a scene. *Hoiem et al.* (2007) proposed to classify image segments into geometric categories using multiple features. *Geiger et al.* (2011) related traffic patterns and vanishing points in 3D. To obtain physically consistent representations, *Gupta et al.* (2010) incorporated the concept of physical gravity and reasoned about object supports. Several methods attempt to specifically solve indoor layout estimation (*Hedau et al.* (2009, 2010); *Wang et al.* (2010); *Zhao and Zhu* (2011); *Pero et al.* (2012); *Schwing and Urtasun* (2012); *Satkin et al.* (2012)). *Hedau et al.* (2009, 2010) proposed a formulation using a cubic room representation and showed that layout estimation can improve object detection. This initial attempt demonstrated promising results, however experiments

were limited to a single object type (bed) and a single room type (bedroom). Other methods by *Lee et al.* (2010); *Zhao and Zhu* (2011) have proposed to improve layout estimation by analyzing the consistency between layout and the geometric properties of objects without accounting for the specific categorical nature of such objects. *Fouhey et al.* (2012) incorporated human pose estimation into indoor scene layout understanding. However, *Fouhey et al.* (2012) does not capture relationships between objects or between an object and the scene type.

A body of work has focused on classifying images into semantic scene categories (*Fei-Fei and Perona* (2005); *Pandey and Lazebnik* (2011); *Quattoni and Torralba* (2009); *Lazebnik et al.* (2006)). *Li et al.* (2010) proposed an approach called *object bank* to model the correlation between objects and scene by encoding object detection responses as features in a SPM and predicting the scene type. They did not, however, explicitly reason about the relationship between the scene and its constituent objects, nor the geometric correlation among objects. Recently, *Pandey and Lazebnik* (2011) used a latent DPM model to capture the spatial configuration of objects in a scene type. This spatial representation is 2D image-based, which makes it sensitive to viewpoint variations. In our approach, we instead define the spatial relationships among objects in 3D, making them invariant to viewpoint and scale transformation. Finally, the latent DPM model assumes that the number of objects per scene is fixed, whereas our scene model allows an arbitrary number of 3DGPs per scene.

## 5.2 Scene Model using 3D Geometric Phrases

The high-level goal of our system is to take a single image of an indoor scene and classify its scene semantics (such as room type), spatial layout, constituent objects and object relationships in a unified manner. We begin by describing the unified scene model which facilitates this process.

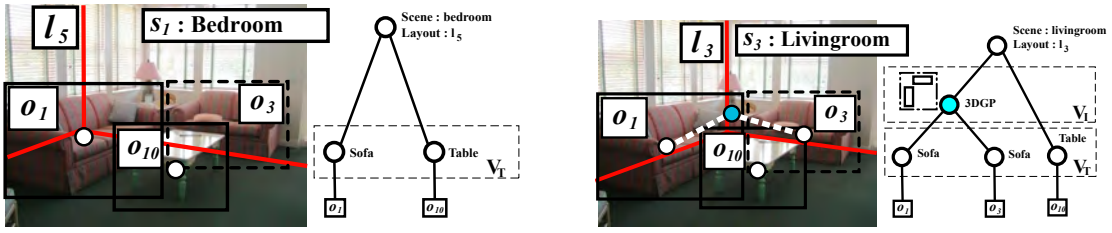



Figure 5.2: Two possible parse graph hypotheses for an image - on the left an incomplete interpretation (where no 3DGP is used) and on the right complete interpretation (where a 3DGP is used). The root node $S$ describes the scene type $s_1, s_3$ (bedroom or livingroom) and layout hypothesis $l_3, l_5$ (red lines), while other white and skyblue round nodes represent objects and 3DGPs, respectively. The square nodes $(o_1, ..., o_{10})$ are detection hypotheses obtained by object detectors such as *Felzenszwalb et al.* (2010) (black boxes). Weak detection hypotheses (dashed boxes) may not be properly identified in isolation (left). A 3DGP, such that indicated by the skyblue node, can help transfer contextual information from the left sofa (strong detections denoted by solid boxes) to the right sofa.

Image parsing is formulated as an energy maximization problem (Sec. 5.2.1), which attempts to identify the parse graph that best fits the image observations. At the core of this formulation is our novel *3D Geometric Phrase* (3DGP), which is the key ingredient in parse graph construction (Sec. 5.2.2). The 3DGP model facilitates the transfer of contextual information from a strong object hypothesis to a weaker one when the configuration of the two objects agrees with a learned geometric phrase (Fig. 5.2 right).

Our scene model $\mathcal{M} = (\Pi, \theta)$ contains two elements; the 3DGPs $\Pi = \{\pi_1, ..., \pi_N\}$ and the associated parameters $\theta$. A single 3DGP $\pi_i$ defines a group of object types (e.g. sofa, chair, table, etc.) and their 3D spatial configuration, as in Fig. 5.1(d). Unlike *Zhao and Zhu* (2011), which requires a training set of hand crafted composition rules and learns only the rule parameters, our method automatically learns the set of 3DGPs from training data via our novel training algorithm (Sec. 5.4). The model parameter $\theta$ includes the observation weights $\alpha, \beta, \gamma$, the semantic and geometric context model weights $\eta, \nu$, the pair-wise interaction model $\mu$, and the parameters $\lambda$

associated with the 3DGP (see eq. 5.1).

We define a parse graph $G = \{S, \mathbb{V}\}$ as a collection of nodes describing geometric and semantic properties of the scene. $S = (C, H)$ is the root node containing the scene semantic class variable $C$ and layout of the room $H$, and $\mathbb{V} = \{V_1, ..., V_n\}$ represents the set of non-root nodes. An individual $V_i$ specifies an object detection hypothesis or a 3DGP hypothesis, as shown in Fig. 5.2. We represent an image observation $I = \{O_s, O_l, O_o\}$ as a set of hypotheses with associated confidence values as follows. $O_o = \{o_1, ..., o_n\}$ are object detection hypotheses, $O_l = \{l_1, ..., l_m\}$ are layout hypotheses and $O_s = \{s_1, ..., s_k\}$ are scene types (Sec. 5.2.3).

Given an image $I$ and scene model $\mathcal{M}$, our goal is to identify the parse graph $G = \{S, \mathbb{V}\}$ that best fits the image. A graph is selected by i) choosing a scene type among the hypotheses $O_s$, ii) choosing the scene layout from the layout hypotheses $O_l$, iii) selecting positive detections (shown as $o_1$, $o_3$, and $o_{10}$ in Fig. 5.2) among the detection hypotheses $O_o$, and iv) selecting compatible 3DGPs (Sec. 5.3).

### 5.2.1 Energy Model

Image parsing is formulated as an energy maximization problem. Let $\mathbb{V}_T$ be the set of nodes associated with a set of detection hypotheses (objects) and $\mathbb{V}_I$ be the set of nodes corresponding to 3DGP hypotheses, with $\mathbb{V} = \mathbb{V}_T \cup \mathbb{V}_I$. Then, the energy of

parse graph $G$ given an image $I$ is:

$$
\begin{aligned}
E_{\Pi,\theta}(G,I) \;=\; & \underbrace{\alpha^{\top}\phi(C,O_s)}_{\text{scene observation}} + \underbrace{\beta^{\top}\phi(H,O_l)}_{\text{layout observation}} + \underbrace{\sum_{V\in\mathbb{V}_T}\gamma^{\top}\phi(V,O_o)}_{\text{object observation}} \\
& + \underbrace{\sum_{V\in\mathbb{V}_T}\eta^{\top}\psi(V,C)}_{\text{object-scene}} + \underbrace{\sum_{V\in\mathbb{V}_T}\nu^{\top}\psi(V,H)}_{\text{object-layout}} \\
& + \underbrace{\sum_{V,W\in\mathbb{V}_T}\mu^{\top}\varphi(V,W)}_{\text{object overlap}} + \underbrace{\sum_{V\in\mathbb{V}_I}\lambda^{\top}\varphi(V,Ch(V))}_{\text{3DGP}}
\end{aligned}
\tag{5.1}
$$

where $\phi(\cdot)$ are unary observation features for semantic scene type, layout estimation and object detection hypotheses, $\psi(\cdot)$ are contextual features that encode the compatibility between semantic scene type and objects, and the geometric context between layout and objects, and $\varphi(\cdot)$ are the interaction features that describe the pairwise interaction between two objects and the compatibility of a 3DGP hypothesis. $Ch(V)$ is the set of child nodes of $V$.

**Observation Features:** The observation features $\phi$ and corresponding model parameters $\alpha, \beta, \gamma$ capture the compatibility of a scene type, layout and object hypothesis with the image, respectively. For instance, one can use the spatial pyramid matching (SPM) classifier (*Lazebnik et al.* (2006)) to estimate the scene type, the indoor layout estimator (*Hedau et al.* (2009)) for determining layout and Deformable Part Model (DPM) (*Felzenszwalb et al.* (2010)) for detecting objects. In practice, rather than learning the parameters for the feature vectors of the observation model, we use the confidence values given by SPM (*Lazebnik et al.* (2006)) for scene classification, from (*Hedau et al.* (2009)) for layout estimation, and from the DPM (*Felzenszwalb et al.* (2010)) for object detection. To allow bias between different types of objects, a constant 1 is appended to the detection confidence, making the feature

two-dimensional as in *Desai et al.* (2011) [1].

**Geometric and Semantic Context Features:** The geometric and semantic context features $\psi$ encode the compatibility between object and scene layout, and object and scene type. As discussed in Sec. 5.2.3, a scene layout hypothesis $l_i$ is expressed using a 3D box representation and an object detection hypothesis $p_i$ is expressed using a 3D cuboid representation. The compatibility between an object and the scene layout ($\nu^\top \psi(V, H)$) is computed by measuring to what degree an object penetrates into a wall. For each wall, we measure the object-wall penetration by identifying which (if any) of the object cuboid bottom corners intersects with the wall and computing the (discretized) distance to the wall surface. The distance is 0 if none of the corners penetrate a wall. The object-scene type compatibility, $\eta^\top \psi(V, C)$, is defined by the object and scene-type co-occurrence probability.

**Interaction Features**: The interaction features $\varphi$ are composed of an object overlap feature $\mu^\top \varphi(V, W)$ and a 3DGP feature $\lambda^\top \varphi(V, Ch(V))$. We encode the overlap feature $\varphi(V, W)$ as the amount of object overlap. In the 2D image plane, the overlap feature is $A(V \cap W)/A(V) + A(V \cap W)/A(W)$ where $A(\cdot)$ is the area function. This feature enables the model to learn inhibitory overlapping constraints similar to traditional non-maximum suppression (*Dalal and Triggs* (2005)).

### 5.2.2 The 3D Geometric Phrase Model

The 3DGP feature allows the model to favor a group of objects that are commonly seen in a specific 3D spatial configuration, e.g. a coffee table in front of a sofa. The preference for these configurations is encoded in the 3DGP model by a deformation cost and view-dependent biases (eq. 5.2).

Given a 3DGP node $V$, the spatial deformation $(dx_i, dz_i)$ of a constituent object is a function of the difference between the object instance location $o_i$ and the learned

[1] This representation ensures that all observation feature associated to a detection has value distributed from negative to positive, so that graphs with different number of objects are comparable.

expected location $c_i$ with respect to the centroid of the 3DGP (the mean location of all constituent objects $m_V$). Similarly, the angular deformation $da_i$ is computed as the difference between the object instance orientation $a_i$ and the learned expected orientation $\alpha_i$ with respect to the orientation of the 3DGP (the direction from the first to the second object, $a_V$). Additionally, 8 view-point dependent biases for each 3DGP encode the amount of occlusion expected from different view-points. Given a 3DGP node $V$ and the associated model $\pi_k$, the potential function can be written as follows:

$$\lambda_k^\top \varphi_k(V, Ch(V)) = \sum_{p \in \mathcal{P}} b_k^p \, \mathbb{I}(a_V = p) - \sum_{i \in Ch(V)} d_k^{i\top} \varphi_k^d(dx_i, dz_i, da_i) \tag{5.2}$$

where $\lambda_k = \{b_k, d_k\}$, $\mathcal{P}$ is the space of discretized orientations of the 3DGP and $\varphi_d(dx_i, dz_i, da_i) = \{dx_i^2, dz_i^2, da_i^2\}$. The parameters $d_k^i$ for the deformation cost $\varphi_k^i$ penalize configurations in which an object is too far from the anchor. The view-dependent bias $b_k^p$ "rewards" spatial configurations and occlusions that are consistent with the camera location. The amount of occlusion and overlap among objects in a 3DGP depends on the view point; the view-dependent bias encodes occlusion and overlap reasoning. Notice that the spatial relationships among objects in a 3DGP encodes their relative positions in 3D space, so the 3DGP model is rotation and view-point invariant. Previous work which encoded the 2D spatial relationships between objects (*Sadeghi and Farhadi* (2011); *Li et al.* (2012); *Desai et al.* (2011)) required large numbers of training images to capture the appearance of co-occuring objects. On the other hand, our 3DGP requires only a few training examples since it has only a few model parameters thanks to the invariance property.[2]

[2]Although the view-dependent biases are not view-point invariant, there are still only a few parameters (8 views per 3DGP).

### 5.2.3 Objects in 3D Space

We propose to represent objects in 3D space instead of 2D image space. The advantages of encoding objects in 3D are numerous. In 3D, we can encode geometric relationships between objects in a natural way (e.g. 3D euclidean distance) as well as encode constraints between objects and the space (e.g. objects cannot penetrate walls or floors). To keep our model tractable, we represent an object by its 3D bounding cuboid, which requires only 7 parameters (3 centroid coordinates, 3 dimension sizes and 1 orientation.) Each object class is associated to a different prototypical bounding cuboid which we call the cuboid model (which was acquired from the commercial website www.ikea.com similarly to *Pero et al.* (2012).) Unlike *Hedau et al.* (2010), we do not assume that objects' faces are parallel to the wall orientation, making our model more general.

Similarly to *Hedau et al.* (2009); *Lee et al.* (2010); *Wang et al.* (2010), we represent the indoor space by the 3D layout of 5 orthogonal faces (floor, ceiling, left, center, and right wall), as in Fig. 5.1(e). Given an image, the intrinsic camera parameters and rotation with respect to the room space $(K, R)$ are estimated using the three orthogonal vanishing points (*Hedau et al.* (2009)). For each set of layout faces, we obtain the corresponding 3D layout by back-projecting the intersecting corners of walls.

An object's cuboid can be estimated from a single image given a set of known object cuboid models and an object detector that estimates the 2D bounding box and pose (Sec. 5.5). From the cuboid model of the identified object, we can uniquely identify the 3D cuboid centroid $O$ that best fits the 2D bounding box detection $o$ and pose $p$ by solving following optimization.

$$\hat{O} = \underset{O}{\operatorname{argmin}} \, ||o - P(O, p, K, R)||_2^2 \tag{5.3}$$

where $P(\cdot)$ is a projection function that projects 3D cuboid $O$ and generates a bounding box in the image plane. The above optimization is quickly solved with a simplex search method (*Lagarias et al.* (1998)). In order to obtain robust 3D localization of each objects and disambiguate the size of the room space given a layout hypothesis, we estimate the camera height (ground plane location) by assuming all objects are lying on a common ground plane. More details are discussed in the appendix B.

## 5.3 Inference

In our formulation, performing inference is equivalent to finding the best parse graph specifying the scene type $C$, layout estimation $H$, positive object hypotheses $V \in \mathbb{V}_T$ and 3DGP hypotheses $V \in \mathbb{V}_I$.

$$\hat{G} = \underset{G}{\operatorname{argmax}}\, E_{\Pi,\theta}(G, I) \tag{5.4}$$

Finding the optimal configuration that maximizes the energy function is NP-hard; assigning binary values to the detection hypotheses requires exponential time. To make this problem tractable, we introduce a novel bottom-up and top-down compositional inference scheme. Inference is performed for each scene type separately, so scene type is considered given in the remainder of this section.

**Bottom-up:** During bottom-up clustering, the algorithm finds all candidate 3DGP nodes $\mathbb{V}_{cand} = \mathbb{V}_T \cup \mathbb{V}_I$ given detection hypothesis $O_o$ (Fig. 5.3 top). The procedure starts by assigning one node $V_t$ to each detection hypothesis $o_t$, creating a set of candidate terminal nodes (leaves) $\mathbb{V}_T = \{\mathbb{V}_T^1, ..., \mathbb{V}_T^{K_o}\}$, where $K_o$ is the number of object categories. By searching over all combinations of objects in $\mathbb{V}_T$, a set of 3DGP nodes, $\mathbb{V}_I = \{\mathbb{V}_I^1, ..., \mathbb{V}_I^{K_{GP}}\}$, is formed, where $K_{GP}$ denotes the cardinality of the learned 3DGP model $\Pi$ given by the training procedure (Sec. 5.4). A 3DGP node $V_i$ is considered valid if it matches the spatial configuration of a learned 3DGP model

$\pi_k$. Regularization is performed by measuring the energy gain obtained by including $V_i$ in the parse graph.

To illustrate, suppose we have a parse graph $G$ that contains the constituent objects of $V_i$ but not $V_i$ itself. If a new parse graph $G' \leftarrow G \cup V_i$ has higher energy $0 < E_{\Pi,\theta}(G', I) - E_{\Pi,\theta}(G, I) = \lambda_k^\top \varphi_k(V_i, Ch(V_i))$, then $V_i$ is considered as a valid candidate. In other words, let $\pi_k$ define the 3DGP model shown in Fig. 5.4(c). To find all candidate 3DGP nodes $\mathbb{V}_I^k$ for $\pi_k$, we search over all possible configurations of selecting one terminal node among the sofa hypotheses $\mathbb{V}_T^{sofa}$ and one among the table hypotheses $\mathbb{V}_T^{table}$. Among those, only candidates that satisfy the regularity criteria are accepted as valid. In practice, this bottom-up search can be performed very efficiently (less than a minute per image) since there are typically few detection hypotheses per object type.

**Top-down:** Given all possible sets of nodes $\mathbb{V}_{cand}$, the optimal parse graph $G$ is found via Reversible Jump Markov Chain Monte Carlo (RJ-MCMC) sampling (Fig. 5.3 bottom). To efficiently explore the space of parse graphs, we propose 4 reversible jump moves, *layout selection*, *add*, *delete* and *switch*. Starting from an initial parse graph $G_0$, the RJ-MCMC sampling draws a new parse graph by sampling a random jump move, and the new sample is either accepted or rejected following Metropolis-Hasting rule. After $N$ iterations, the graph that maximizes the energy function $\text{argmax}_G E(G, I)$ is selected as the solution. The initial parse graph is obtained by 1) selecting the layout with highest observation likelihood (*Hedau et al.* (2009)) and 2) greedily adding object hypotheses that most improve the energy, similarly to *Desai et al.* (2011). The RJ-MCMC jump moves used with a parse graph at inference step $k$ are defined as follows.

***Layout selection:*** This move generates a new parse graph $G_{k+1}$ by changing the layout hypothesis. Among $|L|$ possible layout hypotheses (given by *Hedau et al.* (2009)), one is randomly drawn with probability $exp(l_k)/\sum_i^{|L|} exp(l_i)$, where $l_k$ is the

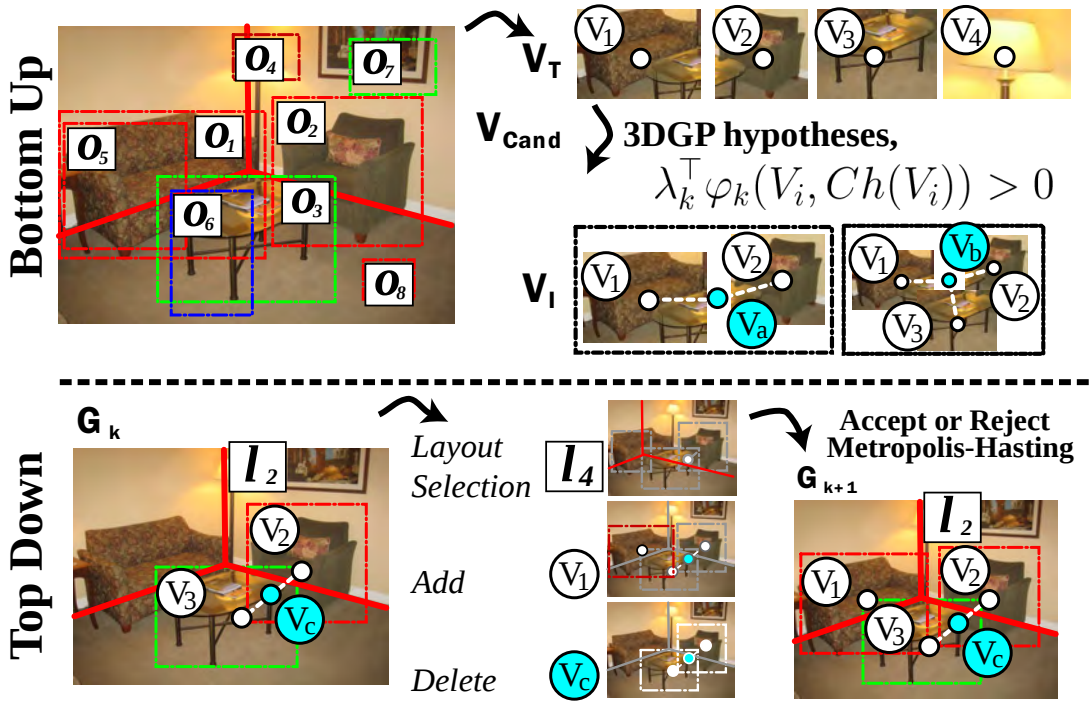



Figure 5.3: **Bottom-up:** Candidate objects $\mathbb{V}_T$ and 3DGP nodes $\mathbb{V}_I$ are vetted by measuring spatial regularity. Red, green and blue boxes indicate sofas, tables and chairs. Black boxes are candidate 3DGP nodes. **Top-down:** the Markov chain is defined by 3 RJ-MCMC moves on the parse graph $G_k$. Given $G_k$, a new $G'$ is proposed via one move and acceptance to become $G_{k+1}$ is decided using the Metropolis-Hasting rule. Moves are shown in the bottom-right subfigures. Red and white dotted boxes are new and removed hypotheses, respectively.

score of the $k^{th}$ hypothesis.

***Add:*** This move adds a new 3DGP or object node from $V_i \in \mathbb{V}_{cand} \setminus G_k$ into $G_{k+1}$. To improve the odds of picking a valid detection, a node is sampled with probability $exp(s_i) / \sum_j^{|\mathbb{V}_{cand} \setminus G_k|} exp(s_j)$, where $s_i$ is the aggregated detection score of all children. For example, in Fig. 5.3(bottom), $s_i$ of $V_c$ is the sum of the sofa and table scores.

***Delete:*** This move removes an existing node $V_i \in G_k$ to generate a new graph $G_{k+1}$. Like the *Add* move, a node is selected with probability $exp(-s_i) / \sum_j^{|G_k|} exp(-s_j)$.

## 5.4 Training

Given input data $x = (O_s, O_l, O_o)$ with labels $y = (C, H, V_T)$ per image, we have two objectives during model training: i) learn the set of 3DGP models $\Pi$ and ii) learn the corresponding model weights $\theta$. Since the model parameters and 3DGPs are interdependent (e.g. the number of model parameters increases with the number of GPs), we propose an iterative learning procedure. In the first round, a set of 3DGPs is generated by a propose-and-match scheme. Given $\Pi$, the model parameters $\theta$ are learned using a latent max-margin formulation. This formulation accommodates the uncertainty in associating an image to a parse graph $G$ similarly to *Felzenszwalb et al.* (2010); *Wang and Mori* (2011); i.e. given a label $y$, the root node and terminal nodes of $G$ can be uniquely identified, but the 3DGP nodes in the middle are hidden.

**Generating** $\Pi$: This step learns a set of 3DGPs, $\Pi$, which captures object groups that commonly appear in the training set in consistent 3D spatial configurations. Given an image, we generate all possible 3DGPs from the ground truth annotations $\{y\}$. The consistency of each 3DGP $\pi_k$ is evaluated by matching it with ground truth object configurations in other training images. We say that a 3DGP is matched if $\lambda_k^\top \varphi_k(V, Ch(V)) > th$ (see Sec. 5.3). A 3DGP model $\pi_k$ is added to $\Pi$ if it is matched more than $K$ times. This scheme is both simple and effective. To avoid redundancy, agglomerative clustering is performed over the proposed 3DGP candidates. Exploring all of the training images results in an over-complete set $\Pi$ that is passed to the parameter learning step.

**Learning** $\theta$ **and pruning** $\Pi$: Given a set of 3DGPs $\Pi$, the model parameters are learned by iterative *latent completion* and *max-margin* learning. In latent completion, the most compatible parse graph $G$ is found for an image with ground truth labels $y$ by finding compatible 3DGP nodes $V_I$. This maximizes the energy over the latent

variable (the 3DGP nodes), $\hat{h}_i$, given an image and label $(x_i, y_i)$.

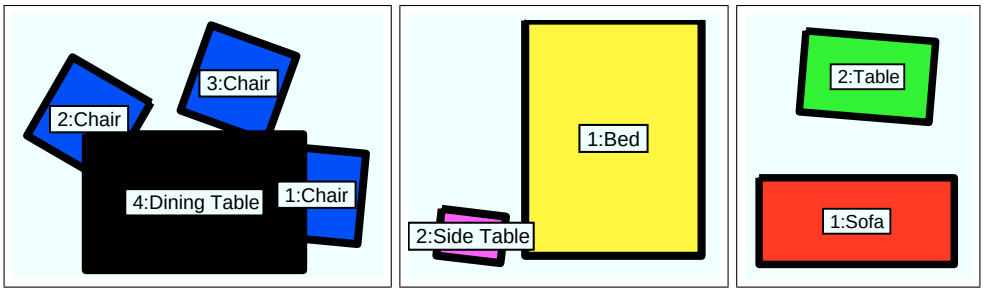
$$\hat{h}_i = \underset{h}{\operatorname{argmax}}\, E_{\Pi,\theta}(x_i, y_i, h) \tag{5.5}$$

After latent completion, the 3DGP models which are not matched with a sufficient number ($< 5$) of training examples are removed, keeping the 3DGP set compact and ensuring there are sufficient positive examples for max-margin learning. Given all triplets of $(x_i, y_i, \hat{h}_i)$, we use the cutting plane method (*Desai et al.* (2011)) to train the associated model parameter $\theta$ by solving the following optimization problem.

$$\begin{aligned} &\min_{\theta,\xi} \frac{1}{2}\|\theta\|^2 + C\sum_i \xi^i \\ &s.t. \quad \max_h E_{\Pi,\theta}(x_i, y, h) - E_{\Pi,\theta}(x_i, y_i, \hat{h}_i) \leq \xi^i - \delta(y, y_i),\ \forall i, y \end{aligned} \tag{5.6}$$

where $C$ is a hyper parameter in an SVM and $\xi^i$ are slack variables. The loss contains three components, $\delta(y, y_i) = \delta_s(C, C_i) + \delta_l(H, H_i) + \delta_d(V_T, V_{Ti})$. The scene classification $\delta_s(C, C_i)$ and detection $\delta_d(V_T, V_{Ti})$ losses are defined using hinge loss. We use the layout estimation loss proposed by *Hedau et al.* (2009) to model the layout estimation loss $\delta_l(H, H_i)$. The process of generating $\Pi$ and learning the associated model parameters $\theta$ is repeated until convergence.

Using the learning set introduced in Sec. 5.5, the method discovers 163 3DGPs after the initial generation of $\Pi$ and retains 30 after agglomerative clustering. After 4 iterations of pruning and parameter learning, our method retains 10 3DGPs. Fig. 5.4 shows selected examples of learned 3DGPs (the complete set is presented in the appendix B.)




(a) (b) (c)

Figure 5.4: Examples of learned 3DGPs. The object class (in color) and the position and orientation of each object is shown. Note that our learning algorithm learns spatially meaningful structures without supervision.

## 5.5 Experimental Results

**Datasets:** To validate our proposed method, we collected a new dataset that we call the *indoor-scene-object* dataset, which we contribute to the community. The *indoor-scene-object* dataset includes 963 images. Although there exist datasets for layout estimation evaluation (*Hedau et al.* (2009)), object detection (*Everingham et al.* (2010)) and scene classification (*Quattoni and Torralba* (2009)) in isolation, there is no dataset on which we can evaluate all the three problems simultaneously. The *indoor-scene-object* dataset includes three scene types: living room, bedroom, and dining room, with ∼300 images per room type. Each image contains a variable number of objects. We define 6 categories of objects that appear frequently in indoor scenes: sofa, table, chair, bed, dining table and side table. In the following experiments, the dataset is divided into a training set of 180 images per scene, and a test set of the remaining images. Ground truth for the scene types, face layouts, object locations and poses was manually annotated. We used $C = 1$ to train the system without tuning this hyper parameter.

**Scene Classifier:** The SPM (*Lazebnik et al.* (2006)) is utilized as a baseline scene classifier, trained via libSVM (*Chang and Lin* (2011)). The baseline scene classification accuracy is presented in Table 5.1. The score for each scene type is the

observation feature for scene type in our model ($\phi(C, O_s)$). We also train two other state-of-the art scene classifiers SDPM (*Pandey and Lazebnik* (2011)) and Object bank (*Li et al.* (2010)) and report the accuracy in Table. 5.1.

**Indoor layout estimation:** The indoor layout estimator as trained in *Hedau et al.* (2009) is used to generate layout hypotheses with confidence scores for $O_l$ and the associated feature $\phi(H, O_l)$. As a sanity check, we also tested our trained model on the indoor UIUC dataset (*Hedau et al.* (2009)). Our model with 3DGPs increased the original 78.8% pixel accuracy rate (*Hedau et al.* (2009)) to 80.4%. Pixel accuracy is defined as the percentage of pixels on layout faces with correct labels.

To further analyze the layout estimation, we also evaluated per-face estimation accuracy. The per-face accuracy is defined as the intersection-over-union of the estimated and ground-truth faces. Results are reported in Table. 5.2.

**Object detection:** The baseline object detector (DPM (*Felzenszwalb et al.* (2010))) was trained using the PASCAL dataset (*Everingham et al.* (2010)) and a new dataset we call the *furniture* dataset containing 3939 images with 5426 objects. The bounding box and azimuth angle (8 view points) of each object were hand labeled. The accuracy of each baseline detector is presented in Fig. 5.5 and Table 5.3. The detection bounding boxes and associated confidence scores from the baseline detectors are used to generate a discrete set of detection hypotheses $O_o$ for our model. To measure detection accuracy, we report the precision-recall curves and average precision (AP) for each object type, with the standard intersection-union criteria for detections (*Everingham et al.* (2010)). The marginal detection score $m(o_i)$ of a detection hypothesis

| | *Li et al.* (2010) | *Pandey and Lazebnik* (2011) | *Lazebnik et al.* (2006) | W/o 3DGP | 3DGP |
|---|---|---|---|---|---|
| Acc. | 76.9 % | 86.5 % | 80.5 % | 85.5 % | **87.7** % |

Table 5.1: Scene classification results using state-of-the-art methods (left-two), the baseline *Lazebnik et al.* (2006) (center) and our model variants (right-two). Our model outperforms all the other methods.

is obtained by using the log-odds ratio that can be approximated by the following equation similarly to *Desai et al.* (2011).

$$m(o_i) = \begin{cases} E_{\Pi}(\hat{G}, I) - E_{\Pi}(\hat{G}_{\setminus o_i}, I), \ o_i \in \hat{G} \\ E_{\Pi}(\hat{G}_{+o_i}, I) - E_{\Pi}(\hat{G}, I), \ o_i \notin \hat{G} \end{cases} \tag{5.7}$$

where $\hat{G}$ is the solution of our inference, $\hat{G}_{\setminus o_i}$ is the graph without $o_i$, and $\hat{G}_{+o_i}$ is the graph augmented with $o_i$. If there exists a parent 3DGP hypothesis for $o_i$, we remove the corresponding 3DGP as well when computing $\hat{G}_{\setminus o_i}$.

To better understand the effect of the 3DGP, we employ two different strategies for building the augmented parse graph $\hat{G}_{+o_i}$. The first scheme $M1$ builds $\hat{G}_{+o_i}$ by adding $o_i$ as an object hypothesis. The second scheme $M2$ attempts to also add a parent 3DGP into $\hat{G}_{+o_i}$ if 1) the other constituent objects in the 3DGP (other than $o_i$) already exist in $\hat{G}$ and 2) the score is higher than the first scheme (adding $o_i$ as an individual object). The first scheme ignores possible 3DGPs when evaluating object hypotheses that are not included in $\hat{G}$ due to low detection score, whereas the second scheme also incorporates 3DGP contexts while measuring the confidence of those object hypotheses.

**Results:** We ran experiments using the new *indoor-scene-object* dataset. To evaluate the contribution of the 3DGP to the scene model, we compared three versions algorithms: 1) the baseline methods, 2) our model without 3DGPs (including geometric and semantic context features), and 3) the full model with 3DGPs. In both 2) and

| Method | Pix. Acc | Floor | Center | Right | Left | Ceiling |
|---|---|---|---|---|---|---|
| *Hedau et al.* (2009) | 81.4 % | 73.4 % | 68.4 % | 71.0 % | 71.9 % | 56.2 % |
| W/O 3DGP | **82.8** % | 76.9 % | **69.3** % | **71.8** % | **72.5** % | **56.3** % |
| 3DGP | 82.6 % | **77.3** % | **69.3** % | 71.5 % | 72.4 % | 55.8 % |

Table 5.2: Layout accuracy obtained by the baseline (*Hedau et al.* (2009)), our model without 3DGP and with 3DGP. Our model outperforms the baseline for all classes.

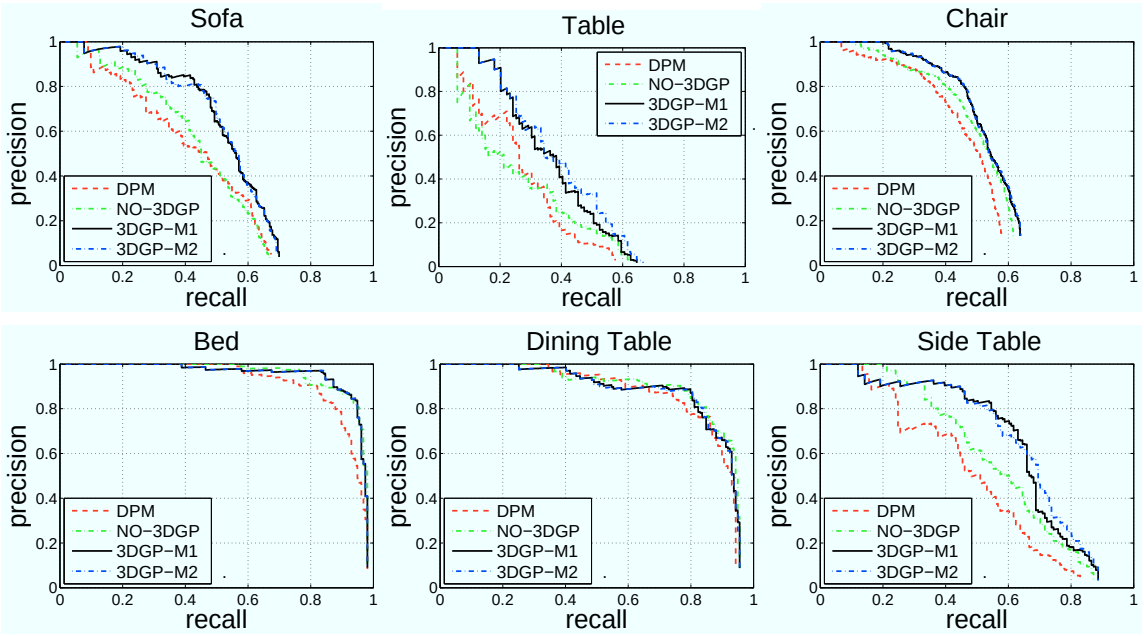



Figure 5.5: Precision-recall curves for DPMs (*Felzenszwalb et al.* (2010)) (red), our model without 3DGP (green) and with 3DGP using M1 (black) and M2 (blue) marginalization. Average Precision (AP) of each method is reported in Table.5.3.

3), our model was trained on the same data and with the same setup.

As seen in the Table 5.3, our model (without or with 3DGPs) improves the detection accuracy significantly $(2-16\%)$ for all object classes. We observe significant improvement using our model without 3DGPs for all objects except tables. By using 3DGPs in the model, we further improve the detection results, especially for side tables (+8% in AP). This improvement can be explained by noting that the 3DGP consisting of a bed and side-table boosts the detection of side-tables, which tend to be severely occluded by the bed itself (Fig. 5.4 (middle)). Fig. 5.7 provides qualitative results. Notice that M2 marginalization provides higher recall rates in lower precision areas for tables and side tables than M1 marginalization. This shows that the 3DGP can transfer contextual information from strong object detection hypotheses to weaker detection hypotheses.

The scene model (with or without 3DGPs) significantly improves scene classi-

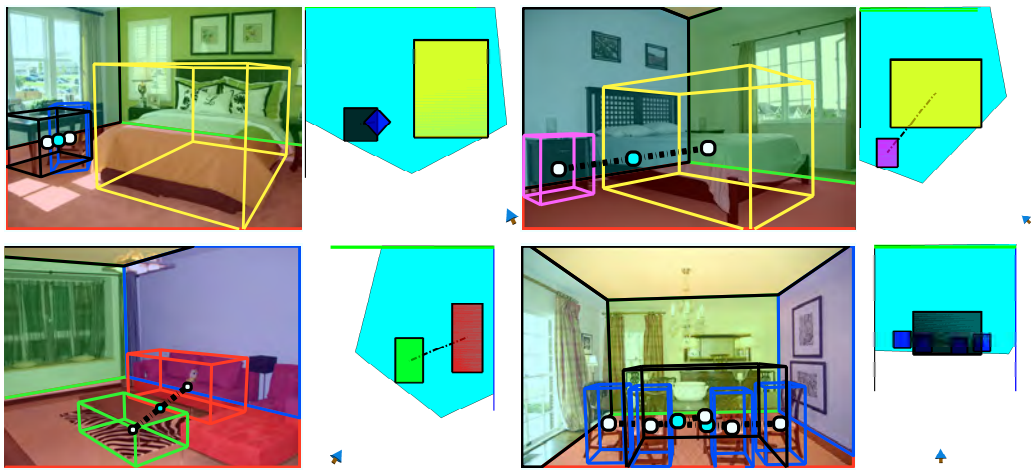


Figure 5.6: 2D and 3D (top-view) visualization of the results using our 3DGP model. Camera view point is shown as an arrow. This figure is best shown in color.

fication accuracy over the baseline (+7.2%) by encoding the semantic relationship between scene type and objects (Table. 5.1). The results suggest that our contextual cues play a key role in the ability to classify the scene. Our model also outperforms state-of-the-art scene classifiers (*Li et al.* (2010); *Pandey and Lazebnik* (2011)) trained on the same dataset.

Finally, we demonstrate that our model provides more accurate layout estimation (Table. 5.2) by enforcing that all objects lie inside of the free space (see Fig. 5.7). We observe that our model does equal or better than the baseline (*Hedau et al.* (2009)) in 94.1%(396/421) of all test images. Although the pixel label accuracy improvement is marginal compared to the baseline method, it shows a significant improvement in the

| Method | Sofa | Table | Chair | Bed | D.Table | S.Table |
|---|---|---|---|---|---|---|
| *Felzenszwalb et al.* (2010) | 42.4 % | 27.4 % | 45.5 % | 91.5 % | 85.5 % | 48.8 % |
| W/O 3DGP | 44.1 % | 26.8 % | 49.4 % | **94.7** % | **87.8** % | 57.6 % |
| 3DGP-M1 | **52.9** % | 37.0 % | 52.5 % | 94.5 % | 86.7 % | 64.5 % |
| 3DGP-M2 | **52.9** % | **38.9** % | **52.6** % | 94.6 % | 86.7 % | **65.4** % |

Table 5.3: Average Precision of the DPM (*Felzenszwalb et al.* (2010)), our model without 3DGP and with 3DGP. Our model significantly outperforms DPM baseline in most of the object categories.

floor estimation accuracy (Table. 5.2). We argue that the floor is the most important layout component since its extent directly provides information about the free space in the scene; the intersection lines between floor and walls uniquely specify the 3D extent of the free space.

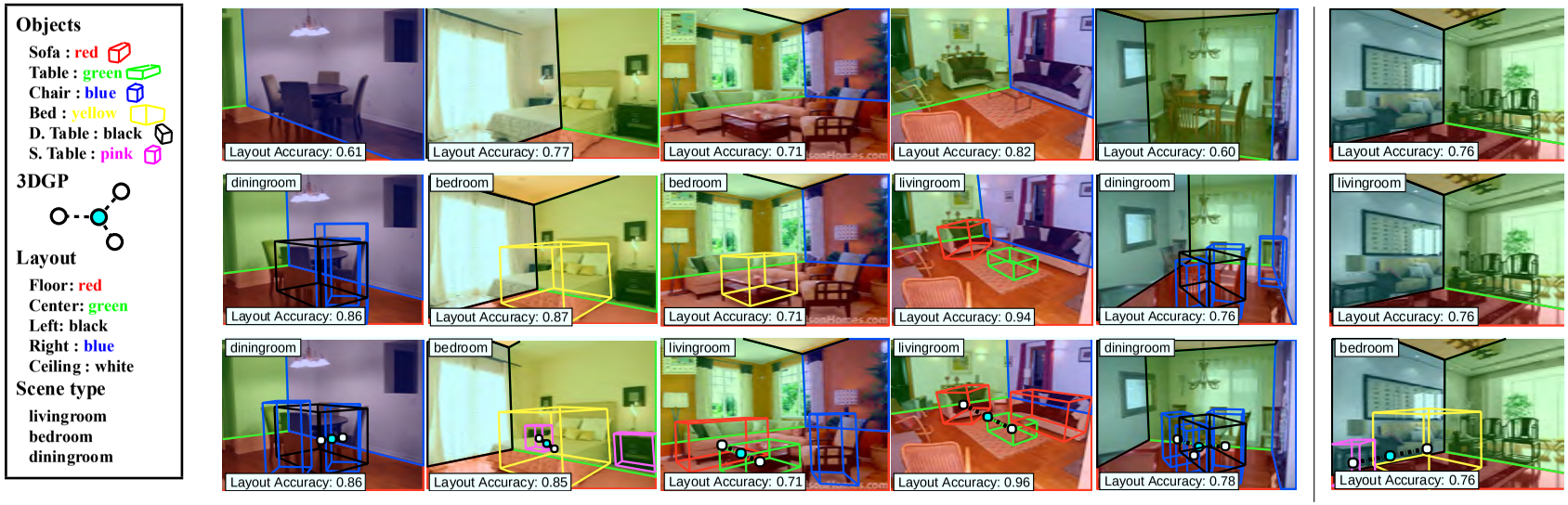



Figure 5.7: Example results. First row: the baseline layout estimator (*Hedau et al.* (2009)). Second row: our model without 3DGPs. Third row: our model with 3DGPs. Layout estimation is largely improved using the object-layout interaction. Notices that the 3DGP helps to detect challenging objects (severely occluded, intra-class variation, etc.) by reasoning about object interactions. Right column: false-positive object detections caused by 3DGP-induced hallucination. See supplementary material for more examples. This figure is best shown in color.

## 5.6 Conclusion

In this chapter, we proposed a novel unified framework that can reason about the semantic class of an indoor scene, its spatial layout, and the identity and layout of objects within the space. We demonstrated that our proposed object 3D Geometric Phrase is successful in identifying groups of objects that commonly co-occur in the same 3D configuration. As a result of our unified framework, we showed that our model is capable of improving the accuracy of each scene understanding component and provides a cohesive interpretation of an indoor image.

# CHAPTER VI

# Conclusion

In this thesis, I propose a number of novel algorithms for understanding complex human behaviors from visual media. Unlike current approaches that focus on recognizing activities using individual attributes, my approach utilizes the interaction among people to understand complex human behaviors. Firstly, in order to understand the individual motion of each agent in uncontrolled environment, I propose a novel model that is capable of identifying the camera motion and unknown number of targets in a joint fashion. I show that the algorithm is capable of estimating the motion of the camera as well as the targets' motion in 3D space given challenging video sequences, that are obtained by a moving camera. Secondly, a novel concept of crowd context is proposed to encode the spatio-temporal relationship among people. Collective activities of people are recognized using STL descriptor and RSTV model based on the crowd context. Experimental evaluation suggests that understanding the crowd context is critical to accurately recognize collective activities. Thirdly, I study the relationship among collectivity activities, interactions and activities performed by individuals in isolation. My study suggests that understanding all of these in a unified framework can provide better recognition results than those in isolated framework. Finally, I propose a novel algorithm that can learn interactions between objects, objects and space, and objects and semantic scene type. The al-

gorithm learns semantically meaningful set of 3D geometric phrases which, in turn, help improve the geometric space estimation, semantic scene classification and object detection. I believe that the model for learning the interactions can be extended to encode interactions between people, person and space, and person and objects, which are the key to understand complex human activities in images and videos.

## 6.1 Future Directions

In this section, I discuss a few possible extensions of the works introduced in this thesis.

### 6.1.1 Parallelizing the RJ-MCMC Particle Filtering for Real-Time Applications

The current version of the multiple target tracking algorithm discussed in chapter II requires about $100 \sim 200ms$ to track people in one time frame. Although the algorithm runs at near real time speed, it is desirable to further improve the efficiency as practical systems typically require to process $15 \sim 30$ frames per second. Also, notice that we could have drawn more samples in the MCMC procedure with the same time budget, if we would have had a more efficient algorithm. This can potentially generate trajectories with higher accuracy. I plan to extend our framework by parallelizing the MCMC sampling procedure to achieve higher efficiency. As we have typically $10 \sim 20$ people in the scene, if one process is assigned to each person to estimate the trajectories in parallel, we can achieve theoretically $10 \sim 20$ speed up in the overall tracking process. I plan to extend our framework for multiple target tracking by investigating this idea to provide a real-time algorithm.

### 6.1.2 Robust Tracking with Semi-batch Tracking Algorithm

Many of recent multiple target tracking algorithms based on tracking-by-detection paradigm could be classified into two categories: 1) tracking by online filtering (*Ess et al.* (2009); *Choi et al.* (2013b); *Wojek et al.* (2011)) or 2) tracking by data association (*Zhang et al.* (2008); *Pirsiavash et al.* (2011)). The former methods (including our work introduced in chapter II) focus on the estimation of targets' states in each time stamp $t$ using the information given up to time $t$. On the contrary, the later methods try to find the best association of all detections using the whole video information. These methods often show more robust detection and accurate tracking in practice, as they are leveraging on long-term information, i.e., the algorithms can identify pedestrians that are occluded by others by considering image frames before and after the occlusion. However, these methods are not applicable to real-time applications, such as autonomous vehicle or robots, since they require to have the past and the future information. We envision that a hybrid semi-batch tracking method, which has short-term buffers of image stream, can substantially improve tracking and detection accuracy, while maintaining the causality characteristic. I plan to explore this direction in the future.

### 6.1.3 Activity Discovery

As another future direction, I would like to investigate the problem of *discovering* complex human activities from unconstrained videos. The problem of *activity discovery* can be defined as "localizing interesting semantic human activities in space and time given a video sequence". Instead of trying to associate a scene (video) or individuals with a single activity label (*activity classification*), the activity discovery aims to discover a set of spatio-temporal volumes wherein specific types of activities appear.

In natural scenarios, there might be multiple different groups of people involved

in different activities. Some people might be talking to each other, while the others are walking around in the scene. Also, individuals can perform multiple activities simultaneously; for example, talking while waiting in a queue, cooking while watching TV, etc. The current methods based on the classification paradigm (employed in chapters III and IV) cannot properly understand such scene as they assume that the scene contains only one activity or that individuals can be associated with only one activity class. One may be able to alleviate the problem by introducing combinatorial activity classes, such as "talking-waiting" or "cooking-watching tv", clinging to the classification paradigm. However, such approach will require exponentially many definitions of the activities (that would be extremely ambiguous to define) and huge amount of training data to learn each combinatorial class ( that might be practically infeasible to obtain).

Activity discovery, on the other hand, can naturally identify multiple simultaneous activities in the scene or those of individuals by discovering multiple spatio-temporal volumes that are associated with a few set of activity labels. I believe that discovering multiple simultaneous activities will allow us to understand better the human activities in visual media and that the algorithms discussed in this thesis will provide an important theoretical foundation to the new direction.

### 6.1.4 Big Database Collection for Activity Recognition

Recently, computer vision enjoyed much improvement in various techniques such as object detection (*Dalal and Triggs* (2005); *Felzenszwalb et al.* (2010); *Leibe et al.* (2004)), scene classification (*Fei-Fei and Perona* (2005); *Quattoni and Torralba* (2009)), and simple action recognition (*Savarese et al.* (2008); *Liu et al.* (2011a); *Niebles et al.* (2008)), thanks to the advancement made in large dataset collection (*Everingham et al.* (2010); *Deng et al.* (2009); *Xiao et al.* (2010)).

Although many large databases (*Everingham et al.* (2010); *Deng et al.* (2009); *Xiao*

*et al.* (2010)) are proposed for visual recognition in images, there are only few and small databases (*Laptev and Lindeberg* (2003); *Liu et al.* (2009)) collected for (simple and complex) activity recognition in videos. Collecting large activity video datasets is substantially harder than image object/scene databases since videos are much larger in size (space issue) and activity classes are more ambiguous to be defined (class ambiguity). Although it is challenging, having a good and sufficiently large dataset is essential to the advancement in visual activity recognition. I plan to work on this direction as a future work.

# APPENDICES

# APPENDIX A

# Appendix A

## A.1 Interaction Feature

In this thesis, we model the interaction feature as a combination of three types of relative motion features, $\psi_l$, $\psi_p$, and $\psi_a$. Each of the feature vector encodes relative motion (distance and velocity), one's location in another's viewpoint, and co-occuring atomic action. All of them are represented as a histogram so as to capture a non-parametric statistics of interactions.

• $\psi_l$ is a feature vector that captures the relative position of a pair of people. In order to describe the motion of one respect to the other, $\psi_l$ is represented as a histogram of velocity and location difference between the two within a temporal window $(t - \triangle t,\ t + \triangle t)$.

• $\psi_p$ encodes a person's location with respect to the other's viewpoint. First, we define the $i^{th}$ target centric coordinate system for each time $t$ by translating the origin of the system to the location of the target $i$ and rotating the x axis along the

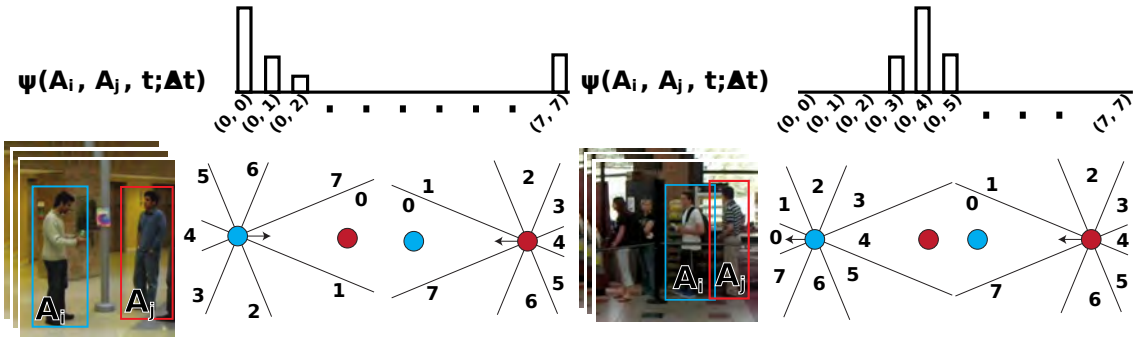



Figure A.1: Illustration of target centric coordinate and histogram $\psi_p$. **Left-bottom** and **Right-bottom** illustrate typical example of *facing-each-other* and *approaching* interaction. Given the location (circle) and pose (arrow) of target $A_i$ and $A_j$, each one's location in terms of the other's view point is obtained as a discretized angle (numbers on the figure). The histograms $\phi_p$ of each example (**top**) are built by counting number of co-occuring discretized angle in a temporal window.

viewing direction (pose) of the target $i$. At each time stamp $t$ in the temporal window, the angle of each target within the others' coordinate system is computed and discretized angle is obtained (see Fig.A.1) in order to describe the location of one person in terms of the viewpoint of the other. Given each location bin, histogram $\psi_p$ is built by counting number of occurrence of the bin number pair to encode the spatial relationship between two targets within a temporal window $(t-\triangle t,\, t+\triangle t)$.

• $\psi_a$ models co-occurrence statistics of atomic actions of the two targets within a temporal window $(t-\triangle t,\, t+\triangle t)$. It is represented as a $|\mathcal{A}|\times(|\mathcal{A}|+1)/2$ dimensional vector of $(a_i(t), a_j(t))$ histogram.

Note that the first two features $\psi_l, \psi_p$ are dependent on the trajectories of the two targets. Thus, change in association will result in a higher or lower value of an interaction potential.

## A.2 Tracklet Association Details

### A.2.1 Hypothesis Generations

For any pair of tracklets $\tau_i, \tau_j$ that are not co-present at the same time-stamp (thus can be linked), we generate $K$ path hypotheses to associate the two tracklets into a unique track. Such hypotheses are obtained by finding $K$-shortest paths between the two tracklets in a detection graph (Fig.A.2). The graph is built by connecting the residual detections between the two tracklets.

To illustrate, consider the example shown in Fig.A.2. Beginning from the last frame (shown as $t-1$) of preceding tracklet $\tau_i$, we find the residual detections at $t$ that have sufficient amount of overlap with the bounding box of $\tau_i$ at $t-1$. We add these detections as a pair of nodes (shown as square nodes in Fig.A.2) and a cost edge (link the two nodes) into the graph. These nodes are linked to the previous frame's tracklet by a directed edge. Subsequently, we add detections in time stamp $t+1$, by calculating the overlap between the added detection in time $t$ and all residual detections in time $t+1$. We add detection nodes in all time stamps between $\tau_i$ and $\tau_j$ iteratively and finish the graph building process by considering the connectivity between $\tau_j$ and detections at $t+2$. The detections in $t+2$ that do not overlap sufficiently with the bounding box of $\tau_j$ at the first frame are discarded.

As noted in the graph, there are exponential (and redundant) number of possible paths that link the two tracklets, which require extensive amount of computation. Especially, if we consider to take the interaction potential into account for tracklet association, it is required to compute an interaction feature for each possible path of target. This can result in infeasible amount of computation in target association. To avoid this issue, we use $K$-shortest path search method ($Yen$) that generate a concise set of path hypothesis to link the two tracklets (Fig.A.2). In practice, we consider the detection confidence to obtain the cost for simplicity. One can add more cost

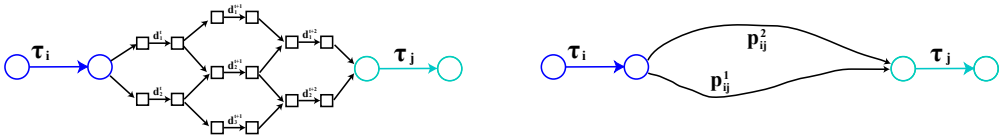



Figure A.2: Illustration of path hypothesis generation given detection residuals. **Left:** the graph is composed of detections in the temporal gap between $\tau_i$ and $\tau_j$. Each detection is represent as a pair of square nodes that are linked by a detection response edge. The cost $d$ associated with the edge encodes the detection confidence value. The detections in time $t+1$ that has enough overlap with the detections in time $t$ are added to the graph. **Right:** given the detection residual graph above, we can obtain a concise set of path proposals using $K$-shortest path search method. Note that there can be exponential number of possible path in the first graph.

features such as color similarity, motion smoothness, if desired. To avoid having no proposal when there are missing detections, we add one default hypothesis that link two tracklets in a shortest distance.

### A.2.2 Match Features

As discussed in the Chap.IV, each path $p_{ij}^k$ is associated to a cost value $c_{ij}^k$ that measures the likelihood that the two tracklets $\tau_i, \tau_j$ belong to the same target. We model this cost value as a linear weighted some of multiple match features: color difference, height difference, motion difference and accumulated detection confidences of the path.

$$c_{ij}^k = w_m^T d_k(\tau_i, \tau_j) \tag{A.1}$$

where $w_m$ is a model weight and $d_k(\tau_i, \tau_j)$ represent the vector representation of all the features. Each of the features is obtained by following: i) color difference is obtained by the Bhattacharyya distance between color histograms of $\tau_i$ and $\tau_j$, ii) height difference is encoded by computing the difference between average height of $\tau_i$ and $\tau_j$, iii) motion difference is computed by absolute difference in the velocity of $\tau_i$ and $\tau_j$, and iv) accumulated detector confidence is calculated by summing up the detection confidence in the path $p_{ij}^k$.

Given the match features, we obtain the cost of each path proposal by Eq.A.1. In the case of target initiation and termination, we use the cost value $c_{en}, c_{ex}$ to model the cost of initiating and terminating a target.

## A.3 Branch-and-Bound Method for Tracklet Association with Interaction Potential

The target association problem with the interaction potential can be written as:

$$\hat{f} = \underset{f}{\operatorname{argmin}}\ c^T f - \Psi(I, A, T(f)) \tag{A.2}$$
$$s.t.\ f_{en,i}, f_{i,ex}, f_{ij}^k \in \{0, 1\}$$
$$f_{en,i} + \sum_j \sum_k f_{ji}^k = f_{i,ex} + \sum_j \sum_k f_{ij}^k = 1$$

where the constraints are summarized as: 1) binary flow constraints (the flow variable should be 0 or 1 integer value specifying that a path is valid or not) and 2) inflow-outflow constraints (the amount of flow coming into a tracklet should be the same as the amount of flow going out of it and the amount is either 0 or 1). The $c$ vector is a cost vector that measures the likelihood of linking two tracklets $c_{ij}^k$ or the cost to initiate/terminate a target $c_{en}, c_{ex}$ and the second term encodes interaction potential which is dependent on the trajectories derived from tracklet association.

### A.3.1 The Non-Convex Quadratic Objective Function

Though the match likelihood is represented as a linear function, the interaction potential involves quadratic relationship between flow variables. As discussed in the Chap.IV, the interaction potential $\Psi(I, A, T(f))$ is composed of a sum of interaction

potentials each of which is associated to a single interaction variable.

$$\Psi(I, A, T) = \sum_{i,j} \Psi(A_i, A_j, I_{ij}, T) \tag{A.3}$$

$$\Psi(A_i, A_j, I_{ij}, T) = \sum_{t \in \mathcal{T}_V} \sum_{a \in \mathcal{I}} w_{ai}^{a} \cdot \psi(A_i, A_j, T, t; \triangle t)\ \mathbb{I}(a, I_{ij}) \tag{A.4}$$

Since the feature function $\psi$ is dependent on, at most two, flow variables, the overall objective function can be represented as a quadratic function.

Before moving into detailed description, we define the *head* and *tail* path of a tracklet $\tau_i$ as the path through which the flow comes into $\tau_i$ and the path through which the flow goes out from $\tau_i$, respectively. The *head* path of $\tau_i$ can be among the entering path $f_{en,i}$ and the path connecting from any other tracklet $\tau_l$, $f_{li}^{k}$. Similarly, the *tail* path of $\tau_i$ can be among the exiting path $f_{ex,i}$ and the path connecting to any other tracklet $\tau_m$, $f_{im}^{k}$. A tracklet $\tau_i$ is called *intact* in a certain temporal support $t \in (t_1,\ t_2)$, if the trajectory of the target is fully covered by the tracklet within the temporal support (i.e, the tracklet is not fragmentized within the time gap). Otherwise, it is called *fragmentized* in a certain temporal support $t \in (t_1,\ t_2)$.

In order to calculate the interaction between two targets $i$ and $j$ at certain time stamp $t$, we need to specify the trajectory of $A_i$ and $A_j$ in all time stamps $t \in (t - \triangle t,\ t + \triangle t)$ (the temporal support of an interaction, Sec.A.1), which can involve selecting at most two flow variables in our flow network.[1] If the both tracklets are *intact* within the temporal support of $I_{ij}^{t}$, the interaction potential does not get affected by tracklet association (we need to specify no flow variable to compute the interaction feature and thus it can be ignored). If only one of the tracklets is *fragmentized* and the other is *intact*, we need to specify only one *head* or *tail* path of the

[1]To be complete, it can involve upto four selections of path proposal to fully specify the trajectories of $A_i$ and $A_j$: *head* of $A_i$, *tail* of $A_i$, *head* of $A_j$ and *tail* of $A_j$ if the two tracklets are both fragmentized in both direction within the temporal support of an interaction. However, we ignore such cases since i) it rarely happens, ii) it make the algorithm to be over-complicated and iii) if the tracklets are too short there are not reliable information we can exploit.

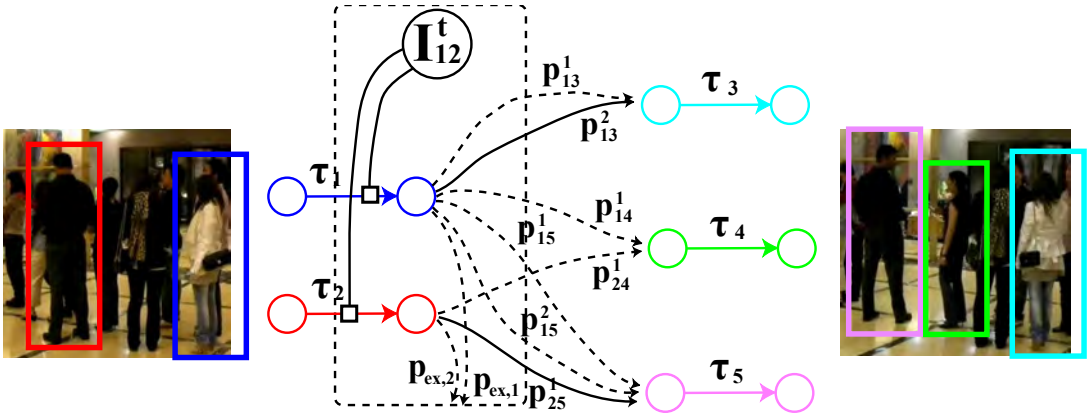



Figure A.3: Consider the case shown in the figure. In order to compute the interaction potential associated with $I^t_{ij}$, we need to specify the *tail* paths of both tracklet $\tau_i$ and $\tau_j$ since they are fragmentized in the temporal support of $I^t_{ij}$ (shown as a dotted box).

fragmentized tracklet. On the other hand, if the both $\tau_i$ and $\tau_j$ are fragmentized in the temporal support, we need to specify two flow variables to obtain the associated interaction feature (*head* or *tail* of $\tau_i$ and *head* or *tail* of $\tau_j$) (see Fig.A.3 for more details).

Since the objective function can be specified as a sum of quadratic and linear functions of flow variable $f$, the problem can be re-written as follows:

$$\begin{aligned} \hat{f} &= \underset{f}{\operatorname{argmin}}\ c^T f - \Psi(I, A, T(f)) \\ &= \underset{f}{\operatorname{argmin}}\ c^T f + c_I^T f + f^T H_I f \\ &\quad s.t.\ f \in \mathbb{S} \end{aligned} \tag{A.5}$$

$\mathbb{S}$ represent the feasible set for $f$ that satisfies the constraints discussed in previous section, the linear part of interaction potential $c_I$ can be obtained by accumulating the interaction potentials that involve only one selection of path (one of the two tracklets $\tau_i, \tau_j$ is *intact* within the temporal support), and $H_I$ can be obtained by accumulating all interaction potentials that involve two selections of flow variables (both of $\tau_i, \tau_j$

are fragmentized in the temporal support of the given interaction variable as in the example of Fig.A.3). Note that $H_I$ is not positive semi-definite (thus non-convex) and standard quadratic programming techniques are not applicable.

### A.3.2 Branch-and-Bound

Since the objective function is non-convex, we employ a novel Branch-and-Bound algorithm to solve the complicated tracklet association problem. The Branch-and-Bound (BB) algorithm we describe here find the global minimum of the objective function over the space $\mathbb{S}$. Starting from the initial subproblem $\mathcal{Q} = \mathbb{S}$, we split the space into two subspaces $\mathcal{Q}_0, \mathcal{Q}_1$ by setting 0 and 1 to a certain flow variable $f_i$ (ignoring/selecting a path). Given each subproblem (where some of flow variables are already set either 0 or 1), we find the lower bound and upper bound (of optimal solution) in the subproblem, $L(\mathcal{Q})$ and $U(\mathcal{Q})$. If the difference between $L$ and $U$ is smaller than a specified precision $\epsilon$ and $U(\mathbb{S})$ is smaller than the lower bound of any other subspace, we stop the iteration and yield the global solution. Otherwise, the algorithm iterate the steps of 1) selecting a subproblem, 2) splitting the subproblem, and 3) finding the lower and upper bound in the subproblem. This is summarized in Algorithm.4.

**Algorithm 4** Branch and Bound (BB) Tracklet Association

$\mathcal{Q} = \mathbb{S}$

$L_0 = L(\mathcal{Q})$

$U_0 = U(\mathcal{Q})$

$\mathcal{L} = \{\mathcal{Q}\}$

**while** $U_k - L_k > \epsilon$, k++ $<$ maxIter **do**

Select a subproblem $\mathcal{Q} \in \mathcal{L}_k$ for which $L(\mathcal{Q}) = L_k$.

Split $\mathcal{Q}$ into $\mathcal{Q}_0$ and $\mathcal{Q}_1$

Form $\mathcal{L}_{k+1}$ from $\mathcal{L}_k$ by removing $\mathcal{Q}$ and adding $\mathcal{Q}_0$ and $\mathcal{Q}_1$

$L_{k+1} = \min_{\mathcal{Q} \in \mathcal{L}_{k+1}} L(\mathcal{Q})$

$U_{k+1} = \min_{\mathcal{Q} \in \mathcal{L}_{k+1}} U(\mathcal{Q})$

**end while**

In following sections, we discuss about how we compute the lower and upper bound of a subproblem $\mathcal{Q}$ (Sec.A.3.3) and which variable is to be split to provide subproblems $\mathcal{Q}_0$ and $\mathcal{Q}_1$ (Sec.A.3.4).

### A.3.3 Lower Bound

In this section, we discuss about the lower bound function that we optimize over in each iteration of our BB algorithm. To make it efficient to solve, we find a linear lower bound function:

$$L(f) = (c + c_I + l)^T f \leq (c + c_I)^T f + f^T H_I f, f \in \mathcal{Q} \tag{A.6}$$

Since the whole interaction potential is represented as a sum of interaction potentials associated with a single interaction variable, it suffices to show that the $l^T f$ is less than or equal to $f^T H f$ within one interaction potential (associated to a single interaction variable $I_{ij}^k$). Thus, we decompose the whole Hessian $H$ into summation of $H_i$ and

show that there exists $l_i$ which is a linear vector that yields a lower bound of $f^T H_i f$, where $i$ denotes an index that enumerates all interaction variables $I_{ij}^k$. It is trivial to show that $l^T f \leq f^T H f$, if $l_i^T f \leq f^T H_i f$, $\forall i$ where $l = \sum_i l_i$ and $H = \sum_i H_i$. The matrix $H_i$ can be obtained by computing the corresponding interaction potential $\Psi(A_i, A_j, I_{ij}^t, T(f))$ given each possible configuration of path flows, e.g. selecting the two solid paths shown in the Fig.A.3.

$$H_i(a,b) = -\frac{1}{2}\Psi(A_i, A_j, I_{ij}^t, T(f)) \; where \; f_a = f_b = 1 \tag{A.7}$$

To obtain the lower bound of $f^T H_i f$, we note on the two characteristics of our problem: i) the variables are binary and ii) there must be one and only one inflow and outflow for each tracklet $\tau_i$. These two facts can be easily derived from the basic constraints of the problem ($\mathbb{S}$). Given these, we notice that always two elements in $H_i$ are selected with symmetry (shown as red box in Fig.A.4) and the values are added to produce $f^T H_i f = H_i(a,b) + H_i(b,a)$ where $a$ and $b$ are the indices of the selected variables in $f$. Thus, it is easy to show that,

$$\min_k H_i(a,k) + \min_k H_i(b,k) \leq H_i(a,b) + H_i(b,a) \tag{A.8}$$

From this, we obtain the lower bound vector $l_i$ for $H_i$ as

$$l_i(a) = \min_k H_i(a,k) \tag{A.9}$$

see Fig.A.4 for illustration. The overall lower bound function is obtained by summing up all lower bounds associated to each interaction variable. $l = \sum_i l_i$.

Given the lower bound vector $l$, the lower bound of $\mathcal{Q}$ is obtained by applying binary integer programming on the lower bound with the given constraints of $\mathcal{Q}$, $\bar{f} = \text{argmin}_f (c + c_I + l)^T f, \; s.t. \; f \in \mathcal{Q}$. The upper bound is set to be infinite if there is

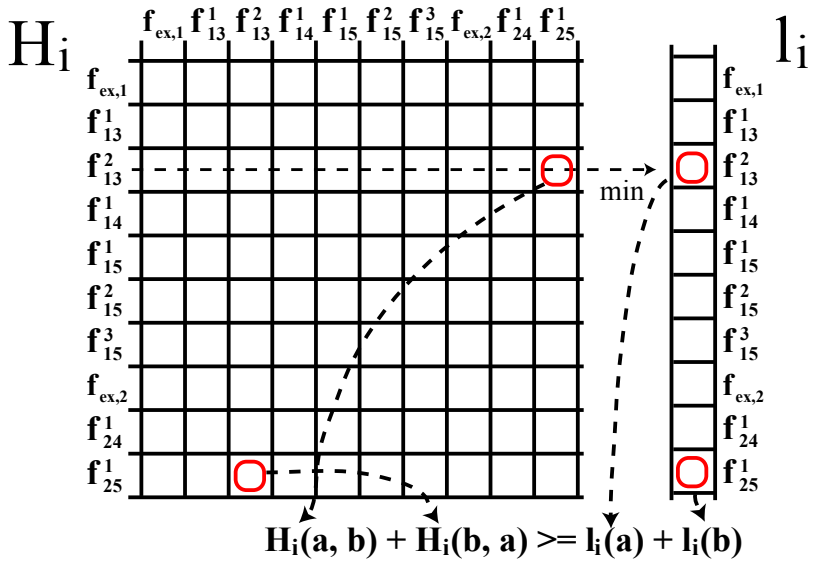



Figure A.4: Illustration of lower bound $L$ computation for the interaction variable corresponding to Fig.A.3. Each element of the Hessian $H_i$ is obtained by computing the corresponding interaction potential $\Psi(A_i, A_j, I^t_{ij}, T)$ given the flow configuration. A linear lower bound $l^T f$ is derived from $f^T H f$ by taking the minimum of each row in the hessian $H$ matrix. Note that only one configuration can be selected in the matrix $H$ with symmetry since no two flow coming out from one tracklet $\tau_i$ or $\tau_j$ can be set simultaneously. The example shows the case when solid edges in Fig.A.3 are selected.

no feasible solution, or set to be the value of original objective function if the solution $\bar{f}$ we obtained is feasible.

### A.3.4 Split Variable Selection

Though the presented lower bound can generate quite tight lower bound in our problem, not all the variables in $f$ have the same "tightness". Setting some variable one or zero will have more uncertainties in the difference between the lower bound and actual objective function, and some will generate smaller differences. To efficiently split the space and find the solution, we choose the variable to be selected based on the selecting 'most ambiguous' variable first strategy.

In order to measure the ambiguity, we derive upper bound vector $u_i$ from $H_i$ by

$$u_i(a) = \max_k H_i(a, k) \tag{A.10}$$

Notice that we take the maximum of a given row in contrast to the minimum in lower bound case (Eq.A.9). Similar to the lower bound vector case, we can obtain full upper bound vector $u$ by accumulating over different interaction variables. It is trivial to show that :

$$l^T f \leq f^T H f \leq u^T f \tag{A.11}$$

Notice that if the value of $l(a)$ is the same as $u(a)$, the value added up in the final objective function by selecting a flow variable $a$ does not make any difference among the above three functions (less ambiguous). However, if the difference between $l(a)$ and $u(a)$ is large, it means that the variable is more ambiguous. Therefore, we choose the variable to be split by finding the variable that has largest difference, $\text{argmax}_a\, u(a) - l(a)$.

# APPENDIX B

# Appendix B

## B.1 Complete Set of Learned GPs $\Pi$

Fig. B.1 shows the 10 GPs $\Pi$ learned by the proposed training method. As shown in the figure, the training method learns GPs that appear frequently in realistic indoor scenes. Notice that training method can learn GPs with arbitrary numbers of constituent objects and that the cardinality of a GP is not predefined.

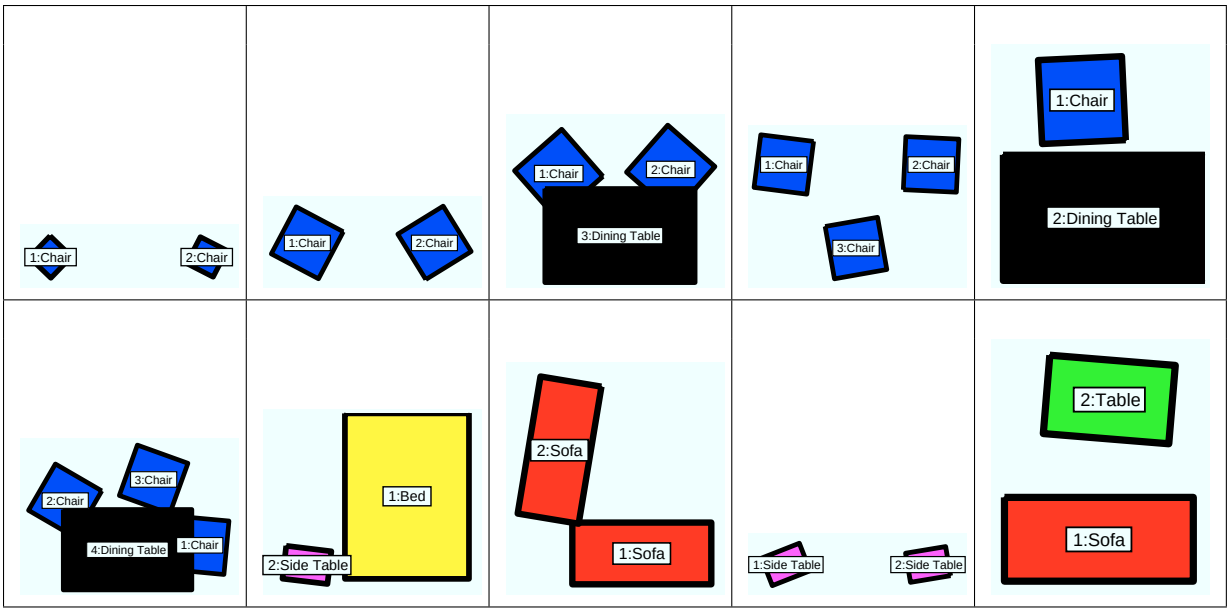



Figure B.1: The complete set of learned GP models Π generated by our learning algorithm. Notice that all the learned GPs embed spatially meaningful configurations of objects. A GP hypothesis can have arbitrary orientation.

## B.2 Example results

In Fig. B.2, we present additional examples of results. The left columns show the output of the baseline layout estimator (*Hedau et al.* (2009)). The middle columns show the result of our system projected into the 2D image. The right columns show the results of our system in 3D, from a top-down view. These example results suggest that our method is capable of producing spatially consistent interpretation of indoor scenes, in which the configurations of objects, layout and scene type are compatible.

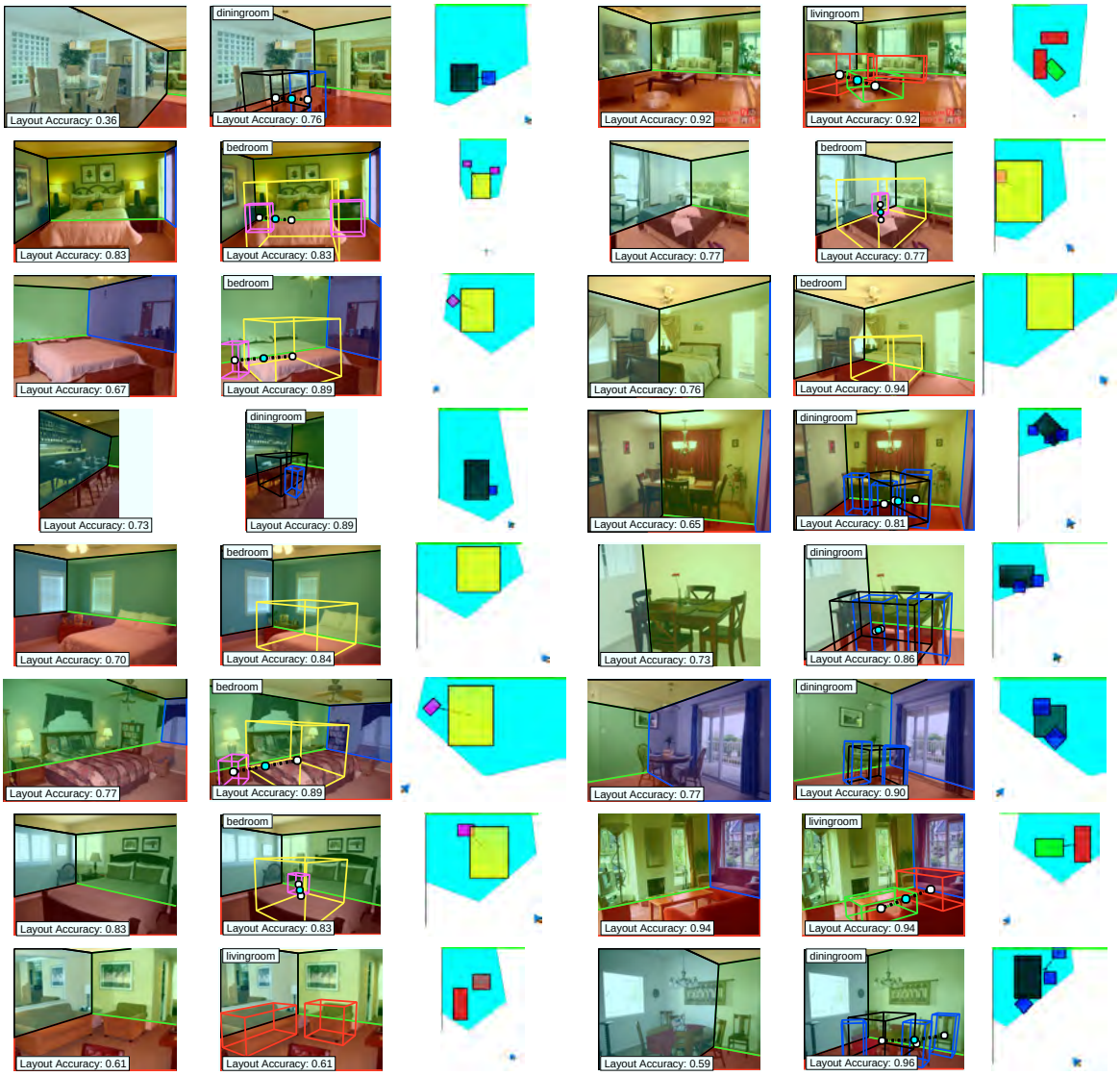



Figure B.2: Example results obtained by the baseline layout estimator (*Hedau et al.* (2009)), ours overlaid on the image and ours shown in 3D space (top-view). The camera viewpoint is shown as blue arrows.

## B.3 Robust 3D Object Localization via a Common Ground Plane Assumption

We estimate the 3D extent of the room space by using the method introduced in *Rother* (2002); *Hedau et al.* (2010). Given an image, one can find three (mutually orthogonal) vanishing points by identifying the points where many parallel lines are

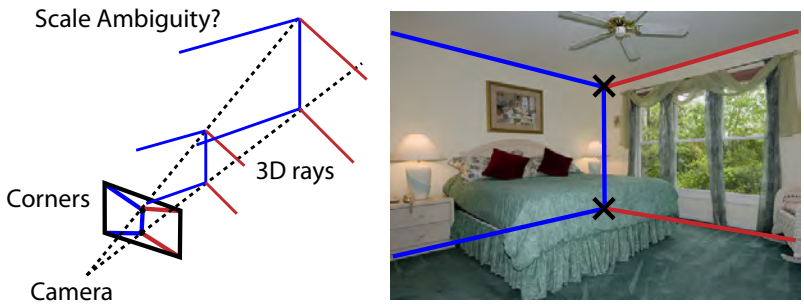



Figure B.3: Given the camera parameters $K, R$ and a layout hypothesis $l_i$ (shown as a lines on the image), the 3D room representation can be obtained by finding the cubes that are intersecting with the 3D rays at its corners (corner rays). Corner rays can be obtained by identifying the rays that intersect both the camera aperture and the layout corners (shown as black crosses) in the image plane. Due to scale ambiguity, there exist infinitely many cubes that are consistent with a layout hypothesis. We identify the unique cube by applying the common ground plane assumption (see text).

intersecting. From this, we can estimate the intrinsic camera parameters and camera rotation with respect to the room space $(K, R)$ using the vanishing points (*Rother* (2002)). Given the pair of camera parameters $K, R$ and a layout hypothesis $l_i$, we obtain the corresponding 3D cubic room representation by finding a 3D cuboid that is consistent with the hypothesis $l_i$ (Fig. B.3). Such a 3D cuboid can be estimated upto scale due to scale ambiguity as shown in the Fig. B.3.

Given a 2D bounding detection $o$ and associated pose $p$, we localize the object in 3D space as a 3D cuboid $O$ by the following optimization,

$$\hat{O} = \underset{O}{\operatorname{argmin}} \, ||o - P(O, p, K, R)||_2^2 \tag{B.1}$$

where $P(\cdot)$ is a camera projection function that projects the 3D cuboid $O$ and generates a fitted bounding box in the image plane. We measure the fitness of a projected bounding box by evaluating the euclidean distance between $o$ and $P(O)$. The above optimization is quickly solved with a simplex search method (*Lagarias et al.* (1998)).

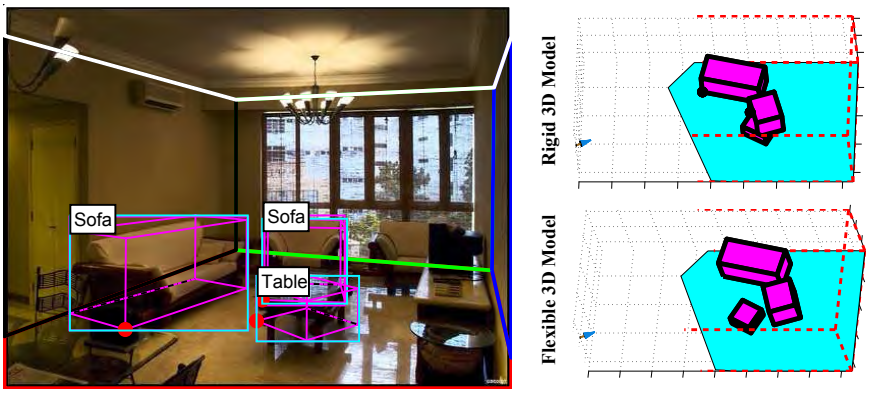



Figure B.4: 3D interpretation of the room and objects given the layout and object hypotheses. The left image shows an example of an image with wall face layouts and object hypotheses (bounding box and reprojected polygon) in the image plane. The right two images show the estimated room space (blue arrow for the camera, red-dotted lines for edges and cyan plane for the ground floor) and object cuboids (magenta colored boxes) in 3D space (top: with rigid 3D model and bottom: with flexible 3D - common ground model). As shown in the figure, the rigid 3D model assumption introduces huge error in 3D localization (table is located below and at the simlar distance as a sofa), yet the common grounded model enables the system to obtain a better 3D estimation.

In practice, an individually estimated 3D cuboid model $\hat{O}$ may be inaccurate due to noisy detection output and intra-class variation in the size of objects as shown in Fig. B.4. Also, in order to estimate the absolute scale of the 3D room space, we need to first find the camera height $h_c$. In order to tackle these issues, we introduce a flexible model that allows each object to have a small variation in its size and assumes a shared ground plane (*Hoiem et al.* (2008)). Each object class is assumed to have a cuboid model with given mean dimensions and one degree of variance in the scale $\alpha$. Given a set of object hypotheses with an associated 3D cuboid $O_i$, we can obtain the scale $\alpha_i$ of all of the objects and the height of the camera $h_c$ using the following optimization:

$$\underset{h_c,\alpha_i}{\operatorname{argmin}} \sum_i (\alpha_i min_y(O_i) - h_c)^2 + C \sum_i log(\alpha_i)^2, \ s.t. \ \forall \alpha_i > 0 \qquad \text{(B.2)}$$

where $min_y(\cdot)$ gives the minimum $y$ value of a 3D cuboid (bottom of the cuboid). The objective function penalizes i) having any objects floating or submerged into the floor and ii) objects deformed too much from the mean 3D model. For any configuration of positive object hypotheses, we run this optimization to obtain the 3D configuration of the image as shown in the Fig. B.4 right bottom. We use $C = 0.1$ in practice.

## B.4 Loss Definition

Given a ground truth label $y_i = (C, H, V_T)$ and an estimated label $y$ of an image $i$, we define the loss function $\delta(y, y_i)$ as a combination of three components: i) object detection loss $\delta_d(V_T, V_{Ti})$, ii) scene classification loss $\delta_s(C, C_i)$ and iii) layout estimation loss $\delta_l(H, H_i)$. Here, $V_T$ represents all positive sets of detection hypotheses in $y$, $H$ is the selected layout hypothesis in $y$ and $C$ is the scene type of $y$.

The detection loss is represented as a sum of individual detection losses. Considering the whole set of detection hypotheses $\mathbb{V}_T$, the detection loss is defined as follows:

$$\begin{aligned}\delta_d(V_T, V_{Ti}) = \sum_{V \in \mathbb{V}_T} & \mathbb{I}(V \in V_T) l_{fp}(V, V_{Ti}) \\ & + \mathbb{I}(V \notin V_T) l_{fn}(V, V_{Ti})\end{aligned} \tag{B.3}$$

The false positive loss $l_{fp}(V, V_{Ti})$ is set to 1 if $V$ does not overlap with any ground truth object with an overlap ratio *Everingham et al.* (2010) larger than 0.5. On the other hand, the false negative loss $l_{fn}(V, V_{Ti})$ is set to 1 if $V$ does overlap with any ground truth object in $V_{Ti}$ with an overlap ratio larger than 0.5.

We incorporate the hinge loss (*LeCun et al.* (2006)) as a classification loss.

$$\delta_s(C, C_i) = \mathbb{I}(C \neq C_i) \tag{B.4}$$

Finally, the layout loss is defined similarly to the one proposed by *Hedau et al.* (2009).

$$\delta_l(H, H_i) = \delta_{l1}(H, H_i) + \delta_{l2}(H, H_i) \tag{B.5}$$

$$\delta_{l1}(H, H_i) = \sum_{k \in [1,5]} d(H_k, H_{ki}) \tag{B.6}$$

$$\delta_{l2}(H, H_i) = \sum_{k \in [1,5]} 1 - \frac{Area(H_k \cap H_{ki})}{Area(H_k \cup H_{ki})} \tag{B.7}$$

where $H_k$ is the $k^{th}$ face of a layout hypothesis $H$, e.g. floor, left wall, or ceiling. $d(H_k, H_{ki}) = 1$ if one of the two is visible and the other is not. $d(H_k, H_{ki}) = 0$ if both are visible or not visible.

# BIBLIOGRAPHY

# BIBLIOGRAPHY


Amer, M. R., and S. Todorovic (2011), A chains model for localizing participants of group activities in videos, in *Proc. of International Conference on Computer Vision (ICCV)*.

Andriluka, M., S. Roth, and B. Schiele (2008), People-tracking-by-detection and people-detection-by-tracking, in *CVPR*.

Avidan, S. (2007), Ensemble tracking, in *PAMI*.

Bay, H., A. Ess, T. Tuytelaars, and L. V. Gool (2008), Surf: Speeded up robust features, *CVIU*.

Belongie, S., J. Malik, and J. Puzicha (02), Shape matching and object recognition using shape contexts, *PAMI*, *24*(4), 509–522, doi:10.1109/34.993558.

Bibby, C., and I. Reid (2008), Robust real-time visual tracking using pixel-wise posteriors, in *ECCV*.

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, Springer.

Breiman, L., and A. Cutler (2004), Random forest, [online].

Breitenstein, M. D., F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool (2009), Robust tracking-by-detection using a detector confidence particle filter, in *ICCV*.

Chang, C.-C., and C.-J. Lin (2001), *LIBSVM: a library for support vector machines*, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chang, C.-C., and C.-J. Lin (2011), LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology*, *2*, 27:1–27:27.

Choi, W., and S. Savarese (2010), Multiple target tracking in world coordinate with single, minimally calibrated camera, in *ECCV*.

Choi, W., and S. Savarese (2012), A unified framework for multi-target tracking and collective activity recognition, in *ECCV*.

Choi, W., and S. Savarese (2013), *Recognizing Complex Human Activities via Crowd Context*, Springer.

Choi, W., K. Shahid, and S. Savarese (2009), What are they doing? : Collective activity classification using spatio-temporal relationship among people., in *VSWS*.

Choi, W., C. Pantofaru, and S. Savarese (2011a), Detecting and tracking people using an rgb-d camera via multiple detector fusion, in *Challenges and Opportunities in Robot Perception, ICCV*.

Choi, W., K. Shahid, and S. Savarese (2011b), Learning context for collective activity recognition., in *CVPR*.

Choi, W., Y. W. Chao, C. Pantofaru, and S. Savarese (2013a), Understanding indoor scenes using 3d geometric phrases, in *CVPR*.

Choi, W., C. Pantofaru, and S. Savarese (2013b), A general framework for tracking multiple people from a moving camera, *Pattern Analysis and Machine Intelligence (PAMI)*.

Comaniciu, D., and P. Meer (2002), Mean shift : A robust approach toward feature space analysis, in *PAMI*.

Dalal, N., and B. Triggs (2005), Histograms of oriented gradients for human detection, in *CVPR*.

Demšar, J. (2006), Statistical comparisons of classifiers over multiple data sets, *The Journal of Machine Learning Research*, *7*, 1–30.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009), ImageNet: A Large-Scale Hierarchical Image Database, in *CVPR09*.

Desai, C., D. Ramanan, and C. C. Fowlkes (2011), Discriminative models for multi-class object layout, *IJCV*.

Dollar, P., V. Rabaud, G. Cottrell, and S. Belongie (2005), Behavior recognition via sparse spatio-temporal features, in *VS-PETS*.

Ess, A., B. Leibe, K. Schindler, and L. van Gool (2008), A mobile vision system for robust multi-person tracking, in *CVPR*.

Ess, A., B. Leibe, K. Schindler, and L. van Gool. (2009), Robust multi-person tracking from a mobile platform, *PAMI*.

Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman (2010), The pascal visual object classes (voc) challenge, *IJCV*, *88*(2), 303–338.

Fanti, C., L. Zelnik-Manor, and P. Perona (2005), Hybrid models for human motion recognition, in *CVPR*, vol. 1, pp. 1166–1173 vol. 1, doi:10.1109/CVPR.2005.179.

Fei-Fei, L., and P. Perona (2005), A bayesian hierarchical model for learning natural scene categories, *CVPR*, pp. 524–531.

Felzenszwalb, P., and D. Huttenlocher (2006), Efficient belief propagation for early vision, in *IJCV*.

Felzenszwalb, P., R. Girshick, D. McAllester, and D. Ramanan (2010), Object detection with discriminatively trained part based models, *PAMI*, *32*(9).

Ferrari, V., M. Marin-Jimenez, and A. Zisserman (2008), Progressive search space reduction for human pose estimation, in *CVPR*.

Fischler, M. A., and R. C. Bolles (1981), Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. of the ACM*, *24*(6).

Fouhey, D. F., V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic (2012), People watching: Human actions as a cue for single-view geometry, in *ECCV*.

Geiger, A., C. Wojek, and R. Urtasun (2011), Joint 3d estimation of objects and scene layout, in *NIPS*.

Gorelick, L., M. Blank, E. Shechtman, M. Irani, and R. Basri (2007), Actions as space-time shapes, *Transactions on Pattern Analysis and Machine Intelligence*, *29*(12), 2247–2253.

Gower, J., and G. Dijksterhuis (2004), *Procrustes Problems*, Oxford University Press.

Gupta, A., A. Efros, and M. Hebert (2010), Blocks world revisited: Image understanding using qualitative geometry and mechanics., in *ECCV*.

Hakeem, A., and M. Shah (07), Learning, detection and representation of multi-agent events in videos, *AI*, *171*, 586–605.

Hartley, R. I., and A. Zisserman (2000), *Multiple View Geometry in Computer Vision*, Cambridge University Press.

Hedau, V., D. Hoiem, and D. Forsyth (2009), Recovering the spatial layout of cluttered room, in *ICCV*.

Hedau, V., D. Hoiem, and D. Forsyth (2010), Thinking inside the box: Using appearance models and context based on room geometry, in *ECCV*.

Helbing, D., and P. Molnar (1995), Social force model for pedestrian dynamics, *Physical review E*, *51*(5), 4282.

Hoiem, D., A. A. Efros, and M. Hebert (2007), Recovering surface layout from an image, *IJCV*.

Hoiem, D., A. A. Efros, and M. Herbert (2008), Putting objects in perspective, *IJCV*.

Intille, S., and A. Bobick (2001), Recognizing planned, multiperson action, *CVIU*.

Izadi, S., et al. (2011), Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera, in *ACM Symposium on User Interface Software and Technology*.

Joachims, T., T. Finley, and C.-N. Yu (2009), Cutting-plane training of structural svms, *Machine Learning*.

Kammerl, J. (2011), Octree point cloud compression in PCL, `http://pointclouds.org/news/compressing-point-clouds.html`.

Khan, Z., T. Balch, and F. Dellaert (2005), MCMC-based particle filtering for tracking a variable number of interacting targets, *PAMI*.

Kim, T., S.-f. Wong, and R. Cipolla (2007), Tensor canonical correlation analysis for action classification, in *CVPR*.

Kohli, P., and P. Torr (2010), Dynamic graph cuts and their applications in computer vision, *Studies in Computational Intelligence*.

Kuhn, H. W. (1955), The hungarian method for the assignment problem, in *Naval Research Logistics Quarterly*.

Kwak, S., W. Nam, B. Han, and J. Han (2011), Learning occlusion with likelihoods for visual tracking, in *ICCV*.

Lagarias, J. C., J. A. Reeds, M. H. Wright, and P. E. Wright (1998), Convergence properties of the nelder–mead simplex method in low dimensions, *SIAM J. on Optimization*.

Lan, T., Y. Wang, G. Mori, and S. Robinovitch (2010a), Retrieving actions in group contexts, in *International Workshop on Sign Gesture Activity*.

Lan, T., Y. Wang, W. Yang, and G. Mori (2010b), Beyond actions: Discriminative models for contextual group activities, in *NIPS*.

Laptev, I., and T. Lindeberg (2003), Space-time interest points, in *ICCV*.

Laptev, I., M. Marszalek, C. Schmid, and B. Rozenfeld (2008), Learning realistic human actions from movies, in *CVPR*.

Lazebnik, S., C. Schmid, and J. Ponce (2006), Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in *CVPR*.

Leal-Taixe, L., G. Pons-Moll, and B. Rosenhahn (2011), Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker, in *Workshop on Modeling, Simulation and Visual Analysis of Large Crowds, ICCV*.

LeCun, Y., S. Chopra, R. Hadsell, M. Ranzato, and F. Huang (2006), A tutorial on energy-based learning, MIT Press.

Lee, D., M. Hebert, and T. Kanade (2009), Geometric reasoning for single image structure recovery, in *CVPR*.

Lee, D., A. Gupta, M. Hebert, and T. Kanade (2010), Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces, in *NIPS*.

Leibe, B., A. Leonardis, and B. Schiele (2004), Combined object categorization and segmentation with an implicit shape model, in *Statistical Learning in Computer Vision, ECCV*.

Li, C., D. Parikh, and T. Chen (2012), Automatic discovery of groups of objects for scene understanding, in *CVPR*.

Li, L.-J., H. Su, E. P. Xing, and L. Fei-Fei (2010), Object bank: A high-level image representation for scene classification & semantic feature sparsification, in *NIPS*.

Li, R., R. Chellappa, and S. K. Zhou (2009), Learning multi-modal densities on discriminative temporal interaction manifold for group activity recognition, in *CVPR*.

Liu, J., S. Ali, and M. Shah (2008), Recognizing human actions using multiple features, in *CVPR*.

Liu, J., J. Luo, and M. Shah (2009), Recongizing realistic actions from videos "in the wild", in *CVPR*.

Liu, J., B. Kuipers, and S. Savarese (2011a), Recognizing human actions by attributes, in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*.

Liu, J., M. Shah, B. Kuipers, and S. Savarese (2011b), Cross-view action recognition via view knowledge transfer, in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*.

Lowe, D. G. (2004), Distinctive image features from scale-invariant keypoints, *IJCV*, *60*(2), 91–110, doi:10.1023/B:VISI.0000029664.99615.94.

Lu, W.-L., and J. J. Little (2006), Simultaneous tracking and action recognition using the pca-hog descriptor, in *Proceedings of the The 3rd Canadian Conference on Computer and Robot Vision*.

Lv, F., and R. Nevatia (2007), Single view human action recognition using key pose matching and viterbi path searching, in *CVPR*.

Marszalek, M., I. Laptev, and C. Schmid (2009), Actions in context, *CVPR*.

Ni, B., S. Yan, and A. Kassim (2009), Recognizing human group activities with localized causalities, in *CVPR*.

Niebles, J. C., H. Wang, and L. Fei-Fei (2008), Unsupervised learning of human action categories using spatial-temporal words, *IJCV*.

Niebles, J. C., C.-W. Chen, , and L. Fei-Fei (2010), Modeling temporal structure of decomposable motion segments for activity classification, in *Proceedings of the 12th European Conference of Computer Vision (ECCV)*, Crete, Greece.

Oliva, A., and A. Torralba (2007), The role of context in object recognition, *Trends in Cognitive Sciences*, *11*(12), 520 – 527.

Pandey, M., and S. Lazebnik (2011), Scene recognition and weakly supervised object localization with deformable part-based models, in *ICCV*.

Patron, A., M. Marszalek, A. Zisserman, and I. Reid (2010), High five: Recognising human interactions in tv shows, in *Proc. BMVC*.

Pellegrini, S., A. Ess, K. Schindler, and L. van Gool (2009), You'll never walk alone: Modeling social behavior for multi-target tracking, in *ICCV*.

Pero, L. D., J. Bowdish, D. Fried, B. Kermgard, E. L. Hartley, and K. Barnard (2012), Bayesian geometric modeling of indoor scenes, in *CVPR*.

Pirsiavash, H., D. Ramanan, and C. Fowlkes (2011), Globally-optimal greedy algorithms for tracking a variable number of objects, in *CVPR*.

Quattoni, A., and A. Torralba (2009), Recognizing indoor scenes, in *CVPR*.

Ramanan, D., D. Forsyth, and A. Zisserman (2007), Tracking people by learning their appearance, *PAMI*.

Ramin Mehran, A. O., and M. Shah (2009), Abnormal crowd behavior detection using social force model, in *CVPR*.

Rodriguez, M., S. Ali, and T. Kanade (2009), Tracking in unstructured crowded scenes, in *ICCV*.

Rother, C. (2002), A new approach for vanishing point detection in architectural environments, *Journal Image and Vision Computing (IVC)*.

Rusu, R. B., and S. Cousins (2011), 3d is here: Point cloud library (pcl), in *ICRA*, Shanghai, China.

Ryoo, M. S., and J. K. Aggarwal (2009), Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities, in *ICCV*.

Ryoo, M. S., and J. K. Aggarwal (2010), Stochastic representation and recognition of high-level group activities, *IJCV*.

Sadeghi, A., and A. Farhadi (2011), Recognition using visual phrases, in *CVPR*.

Satkin, S., J. Lin, and M. Hebert (2012), Data-driven scene understanding from 3d models, in *BMVC*.

Savarese, S., A. DelPozo, J. Niebles, and L. Fei-Fei (2008), Spatial-temporal correlatons for unsupervised action classification, in *WMVC*.

Schwing, A. G., and R. Urtasun (2012), Efficient exact inference for 3d indoor scene understanding, in *ECCV*.

Scovanner, P., and M. Tappen (2009), Learning pedestrian dynamics from the real world, in *ICCV*.

Shitrit, H. B., J. Berclaz, F. Fleuret, and P. Fua (2011), Tracking multiple people under global appearance constraints, in *ICCV*.

Singh, V. K., B. Wu, and R. Nevatia (2008), Pedestrian tracking by associating tracklets using detection residuals, in *IMVC*.

Song, Y., L. Goncalves, and P. Perona. (2003), Unsupervised learning of human motion, *PAMI*, *25*(25), 1–14.

Swears, E., and A. Hoogs (2011), Learning and recognizing complex multi-agent activities with applications to american football plays, in *WACV*.

Tomasi, C., and T. Kanade (1991), Detection and tracking of point features, in *Carnegie Mellon University Technical Report*.

Turaga, P., R. Chellappa, V. S. Subrahmanian, and O. Udrea (2008), Machine recognition of human activities: A survey, *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*.

Tuzel, O., F. Porikli, and P. Meer (2007), Human detection via classification on riemannian manifolds, in *CVPR*.

Viola, P., and M. Jones (2003), Robust real-time face detection, *IJCV*, *57*(2), 137–154.

Viola, P., M. Jones, and D. Snow (2003), Detecting pedestrians using patterns of motion and appearance, in *ICCV*.

Wang, H., S. Gould, and D. Koller (2010), Discriminative learning with latent variables for cluttered indoor scene understanding, in *ECCV*.

Wang, Y., and G. Mori (2011), Hidden part models for human action recognition: Probabilistic versus max margin, *PAMI*.

Weston, J., and C. Watkins (1998), Multi-class support vector machines.

Wojek, C., S. Walk, and B. Schiele (2009), Multi-cue onboard pedestrian detection, in *CVPR*.

Wojek, C., S. Walk, S. Roth, and B. Schiele (2011), Monocular 3d scene understanding with explicit occlusion reasoning, in *CVPR*.

Wong, S., T. Kim, and R. Cipolla (2007), Learning motion categories using both semantics and structural information, in *CVPR*.

Wu, B., and R. Nevatia (2007), Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors, in *IJCV*.

Xiang, Y., and S. Savarese (2012), Estimating the aspect layout of object categories, in *CVPR*.

Xiao, J., J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba (2010), Sun database: Large-scale scene recognition from abbey to zoo, in *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pp. 3485–3492, IEEE.

Yamaguchi, K., A. C. Berg, T. Berg, and L. Ortiz (2011), Who are you with and where are you going?, in *CVPR*.

Yao, A., J. Gall, and L. Van Gool (2010), A hough transform-based voting framework for action recognition, in *CVPR*.

Yen, J. Y. (), Finding the k shortest loopless paths in a network, *Management Science*.

Yu, T., T.-K. Kim, and R. Cipolla (2010), Real-time action recognition by spatiotemporal semantic and structural forest, in *Proc. of British Machine Vision Conference (BMVC)*.

Zhang, L., Y. Li, and R. Nevatia (2008), Global data association for multi-object tracking using network flows, in *CVPR*.

Zhao, Y., and S.-C. Zhu (2011), Image parsing via stochastic scene grammar, in *NIPS*.

Zhou, B., X. Wang, and X. Tang (2012), Understanding collective crowd behaviors: Learning mixture model of dynamic pedestrian-agents, in *CVPR*.