ABSTRACT

Title of dissertation: LEARNING VISUAL PATTERNS: IMPOSING ORDER ON OBJECTS, TRAJECTORIES AND NETWORKS

Ryan M. Farrell,
Doctor of Philosophy, 2011

Dissertation directed by: Professor Larry S. Davis
Department of Computer Science

Fundamental to many tasks in the field of computer vision, this work considers the understanding of observed visual patterns in static images and dynamic scenes . Within this broad domain, we focus on three particular subtasks, contributing novel solutions to: (a) the subordinate categorization of objects (avian species specifically), (b) the analysis of multi-agent interactions using the agent trajectories, and (c) the estimation of camera network topology.

In contrast to object recognition, where the presence or absence of certain parts is generally indicative of basic-level category, the problem of subordinate categorization rests on the ability to establish salient distinctions amongst the characteristics of those parts which comprise the basic-level category. Focusing on an avian domain due to the fine-grained structure of the category taxonomy, we explore a pose-normalized appearance model based on a volumetric poselet scheme. The variation in shape and appearance properties of these parts across a taxonomy provides the cues needed for subordinate categorization. Our model associates the

underlying image pattern parameters used for detection with corresponding volumetric part location, scale and orientation parameters. These parameters implicitly define a mapping from the image pixels into a pose-normalized appearance space, removing view and pose dependencies, facilitating fine-grained categorization with relatively few training examples.

We next examine the problem of leveraging trajectories to understand interactions in dynamic multi-agent environments. We focus on perceptual tasks, those for which an agent's behavior is governed largely by the individuals and objects around them. We introduce *kinetic accessibility*, a model for evaluating the perceived, and thus anticipated, movements of other agents. This new model is then applied to the analysis of basketball footage. The kinetic accessibility measures are coupled with low-level visual cues and domain-specific knowledge for determining which player has possession of the ball and for recognizing events such as passes, shots and turnovers.

Finally, we present two differing approaches for estimating camera network topology. The first technique seeks to partition a set of observations made in the camera network into individual object trajectories. As exhaustive consideration of the partition space is intractable, partitions are considered incrementally, adding observations while pruning unlikely partitions. Partition likelihood is determined by the evaluation of a probabilistic graphical model, balancing the consistency of appearances across a hypothesized trajectory with the latest predictions of camera adjacency. A primarily benefit of estimating object trajectories is that higher-order statistics, as opposed to just first-order adjacency, can be derived, yielding resilience

to camera failure and the potential for improved tracking performance between cameras. Unlike the former centralized technique, the latter takes a decentralized approach, estimating the global network topology with local computations using sequential Bayesian estimation on a modified multinomial distribution. Key to this method is an information-theoretic appearance model for observation weighting. The inherently distributed nature of the approach allows the simultaneous utilization of all sensors as processing agents in collectively recovering the network topology.

LEARNING VISUAL PATTERNS: IMPOSING ORDER ON OBJECTS, TRAJECTORIES AND NETWORKS

by

Ryan M. Farrell

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2011

Advisory Committee:
Professor Larry S. Davis, Chair/Advisor
Professor Trevor Darrell
Professor David W. Jacobs
Professor David Mount
Professor Derek C. Richardson, Dean's Representative

Dedication

It is with a grateful heart that I dedicate my dissertation

to my wife, Sally, to our children, Sam, Liam and Zoe,

to my brother, Dylan, and to my parents, Jim and Kathy.

Acknowledgments

In considering the efforts culminating in this thesis, there are a great many people to whom I am indebted and whose contributions, support and friendship I wish to recognize. It hopefully comes as no surprise that I prefer to say too much by way of thanksgiving than too little; especially with my memory, I already fear having left someone out.

I first thank my advisor, Professor Larry S. Davis, who has provided me with constant and invaluable guidance and feedback since I first undertook this research. Though the demands on his time and attention were great, I cannot recall a time when he was unwilling to sit down and discuss my work. It was always clear that his priority was on helping me to develop as a researcher, from the initial selection of interesting problems all the way through crafting a well-written paper for publication. While I have a great deal yet to learn, much of what I have learned and the experience that I have acquired has been at his hands.

There are a great many others at the University of Maryland who have imparted of both scholarship and friendship during my graduate studies. I selected Professors David Jacobs and Dave Mount as members of my committee based, not only on similarity between their research and my own, but on the esteem I acquired for them both as a student in their courses and working as a teaching assistant for them. I thank them, together with Professor Derek Richardson, for the time that they have devoted as members of my committee. I also thank Professor Amitabh Varshney, my initial academic advisor, and Professor Ramani Duraiswami, for help-

ing me to get started on the right foot. I also want to thank Professor Mike Hicks for willingly, though perhaps unknowingly, assuming a mentoring role as I have navigated graduate school. He is one who I have come to greatly respect, and one with whom I have spent many enjoyable lunchtime discussions.

Amongst my peers, I wish to thank three individuals in particular, Ani Kembhavi, Vlad Morariu, and Ajay Mishra, each of whom have been there for me not only as colleagues, but as close friends. I also thank my officemates over several years, both for their input and feedback regarding my research as well as their patience with my whiteboard scrawlings and occasional teleconferenced calls: Samah Ramadan, Shiv Naga Prasad Vitaladevuni, Son Tran, Zhe Lin, Zhuolin Jiang, Arpit Jain, and Xi (Stephen) Chen. I wish to thank the others in Professor Davis' research group including William Schwartz, Mohamed Hussein, Hazem El-Alfy, Radu Dondera, Abhinav Gupta, Tom Yeh, especially my co-authors Behjat Siddiquie and Om Oza. Other collaborators I thank include Jean-Francois Savard, Professor Gerald Borgia, Yuancheng (Mike) Luo and David Doermann. I also thank my friend Mike Lam, both for our regular yet unforgettable discussions.

I also wish to thank my collaborators and colleagues at other institutions, particularly Professor Trevor Darrell, who graciously hosted me as a visiting researcher to his group during the summer of 2010. I believe that just as Professor Davis had earlier, he saw something in me despite my limited track record. I am grateful not only for his guidance that summer, but for his continued direction alongside that of Professor Davis. Others who made my summer at UC Berkeley fulfilling include Mario Fritz, Tinne Tuytelaars, Mario Christoudias, Ashley Eden, Brian Kulis, Sanja

Fidler and Matthieu Salzmann. And a special thanks go to Bill and Cindy Scott, who opened both their hearts and their home to me for the summer. I will always remember their kindness and friendship with much gratitude.

I wish also to acknowledge and thank my collaborators and co-authors from various other internships: Phil David at the Army Research Lab; Dennis Lucarelli, Anshu Saksena, and I-Jeng Wang at the Johns Hopkins University Applied Physics Lab and Roberto Garcia and Professor Andreas Terzis at Johns Hopkins; Hasan Ozdemir and K-C Lee of the Panasonic Research Lab; and last but not least, Evelyne Tzoukermann, Tony Davis, Robert Rubinoff, Gene Chipman, Amit Bagga, Bageshree Shevade, Geetu Ambwani, Hongzhong Zhou, and Olivier Jojic at Streamsage. Two others in particular: Jan Neumann who has been a tremendous resource for me, and Kay Anderson whose pervasive cheerfulness and humor always brought a smile.

I wish also to return and thank those who have served as mentors for me over the years. These individuals include Eleanora Jadwin, Bev Elkins, Ted Wassam, Sue Antink, Arnie Lim, Professor Scott Klemmer (who, as a graduate student, was my first research "advisor"), Professor James Landay, and Professor Bryan Morse. I also wish to thank Tim Spahr, a mentor to me in more ways than one.

I now wish to take a moment to acknowledge several of the many dear friends who have leaped to our family's aid at times when it was most needed and who have helped us to keep life balanced. I express my thanks to the Bauns, the Botts, the Browns, the Bywaters, the Curtis', the Gerharts, the Heaths, the Horrocks, the Lamberts and especially the Steward family and Bill and Linda (Pop-pop and

Nonna) Braswell. In particular, I want to thank Tom and Susan Draper and Rick and Kathi Kramer. Both families understand well the sacrifices necessitated by completion of one's doctoral study and have been dear, dear friends, supporting our family in innumerable ways.

And now we come to those who do, and always will, hold the dearest places in my heart... my family. I first thank my brother, Dylan, who I love, appreciate and respect more than he may ever know. I thank my uncle, Steve, who shared so much with me, and whose humor and wit I so dearly miss. I thank my aunt, Lois, both for her thoughtfulness and for sharing with me much of my heritage.

While time has provided me with an ever increasing appreciation for my upbringing, opportunities to formally thank one's parents are few and far between. My father, Jim, instilled in me my love of design, construction, analysis and visualization, specifically my love and curiosity for all things scientific and mathematical. My mother, Kathy, and my grandmother, Betty, both teachers, endowed me with an excitement toward education, a love of learning, and a taste for tutelage. They have, in so many ways, supported and encouraged my education since long before I first set foot in a formal classroom.

The scholarly pursuit of my doctorate has been the greatest academic endeavor of my life thus far. Its completion, however, is perhaps most appreciated by she who has been my greatest support throughout this adventure, my best friend and dear wife, Sally. She has celebrated with me the triumphs and endured with me the discouragements, for her constant and unfailing support I am most grateful. I am

also deeply grateful to our precious children, Sam, Liam and Zoe, whose smiles and hugs have greeted me each day as I return home... these four constitute the dearest and greatest of all my endeavors.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

CCTV	Closed-Circuit TeleVision
DBN	Dynamic Bayesian Network
DTIM	Discriminative Temporal Interaction Manifold
EM	Expectation-Maximization
GPS	Global Positioning System
HOG	Histogram of Oriented Gradients
HMM	Hidden Markov Model
HSV	Hue-Saturation-Value Color Space
KDE	Kernel Density Estimate (also known as Parzen Windows)
K-L Divergence	Kullback-Leibler Divergence
L2-norm	Euclidean Norm or Vector Magnitude
MAP	Maximum A Posteriori Estimate
MATLAB	MATrix LABoratory, a scientific application by The Mathworks
MCMC	Markov Chain Monte Carlo
MHT	Multiple Hypothesis Tracker
MLE	Maximum Likelihood Estimate
MLN	Markov Logic Network
MPEG	Moving Picture Experts Group
PHOW	Pyramidal Histogram Of Words
PNAD	Pose-Normalized Appearance Descriptor
RF	Random Forest
RGB	Red-Green-Blue Color Space
SPRT	Sequential Probability Ratio Test
SVM	Support Vector Machine

Chapter 1

Introduction

1.1 Preliminaries

Learning visual patterns is fundamental to many problems in the field of computer vision. Crucial to the ability to successfully learn and recognize such patterns is an appropriate model or representation of the visual phenomena observed. In this dissertation, we specifically consider three problems:

i) subordinate object categorization (specifically for avian species);

ii) analysis of multi-agent interactions using the agent trajectories; and

iii) estimation of camera network topology.

For each of these problems, we provide both background on the problem and a discussion of previous work in the respective domain and other related research. We then describe our solution, detailing the novel characteristics of our proposed representation and how the constraints it imposes effectively address the given problem.

1.2 Subordinate Categorization

We first consider the problem of subordinate categorization, namely, identifying the object in a novel image not only as an instance of a given basic-level category

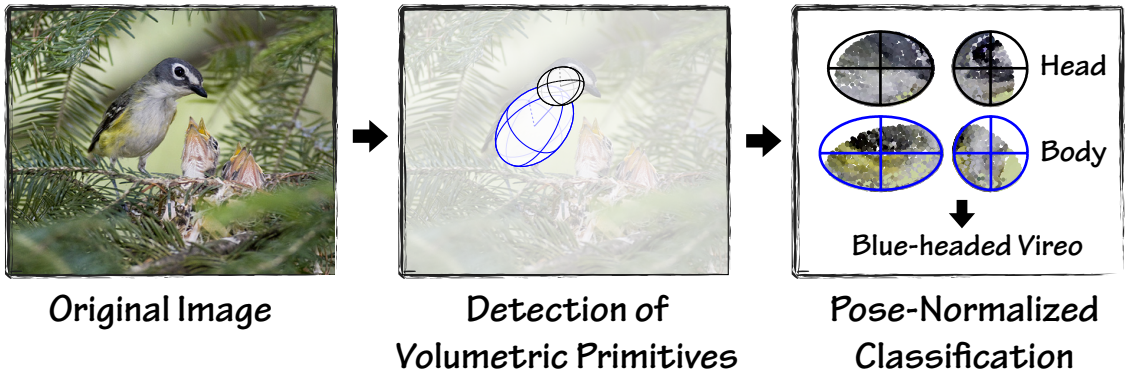


Figure 1.1: **Subordinate Categorization - Proposed Approach.** Basic-level categories are modeled by a configuration of volumetric primitives or parts. Detection recovers these parts and enables application of a pose-normalized appearance model for classification within a taxonomy of subordinate categories.

(*e.g.* frog or automobile), but rather by assigning it to a finer-grained category label (*e.g.* spring peeper, *Pseudacris crucifer*, or Lamborghini Countach). Basic-level categories are defined largely by a collection of constituent parts. Elephants, for example, have (in addition to their head and body) four legs, a small tail, large ears, a large proboscis (trunk), and tusks. Basic-level categorization thus relies principally upon the recognition of these object parts that are either category-specific (like the elephant's trunk, tusks and large ears) or shared amongst a subset of other basic-level categories (like the head, body and legs and perhaps tail).

Most computational approaches for categorization rely upon learning coarse characterizations of shape and/or appearance based purely on local or global image statistics, instead of explicitly modeling parts and/or pose. In the context of subordinate categorization, however, distinguishing features can be so subtle that

localization of semantic parts can be crucial. Consider, for example, the differences between African (genus *Loxodonta*) and Asian (genus *Elephas*) elephants: African elephants have much larger ears, generally have a concave back, are generally less hairy, and both male and female African elephants have tusks (whereas only male Asian elephants do). Coarse-grained image statistics generally fall short of capturing such details.

Figure 1.2: **Comparison of Subordinate Species.** The elephant on the left is an African Elephant (genus *Loxodonta*), the one on the right an Asian Elephant (genus *Elephas*). Photo Credits: [110] and [35], respectively.

In this work, we focus on subordinate categorization in the avian domain, a taxonomy comprised of approximately ten thousand unique species, each with generally the same set of anatomical parts: a beak or bill, a head, a body, two wings, two feet and a tail. There is clearly great variation in both shape and appearance across these parts (consider, for example, the distinctive bills of pelicans, hummingbirds and toucans). It is not the presence or absence of these parts that allows us to differentiate amongst bird species, rather, it is precisely these variations in shape (*e.g.* aspect, cross-section, proportions relative to other parts) and appearance (particularly part colors, streaking patterns, contrasting stripes or patches) that allows

visual discrimination between species.

Motivated by Biederman's Recognition-by-Components theory [8], our approach represents objects as a constellation of volumetric primitives, each a semantically meaningful part. To facilitate finding these volumetric parts in an image, we extend the poselet model recently proposed by Bourdev *et al*. [12, 13] to accommodate such primitives instead of just keypoints. Given these semantic parts, ellipsoids for the bird's head and body in our case, we describe a pose-normalized appearance model which implicitly maps image pixels into a pose-normalized space, thus removing view and pose dependencies. This pose-normalized (or semantic) appearance space facilitates discrimination amongst closely-related species, improving accuracy in subordinate-level categorization.

This work is currently under review.

1.3 Analyzing Interactions in Multi-Agent Environments

We next examine the task of understanding the behavior of multiple interacting agents using their trajectories. Within the vast space of multi-agent environments, we focus on domains dominated by perceptual tasks, those for which an agent's behavior is governed largely by the locations and movements of the other individuals and objects around them. As a model for such perceptual tasks, we introduce *kinetic accessibility*, a representation based on the spatiotemporal geometry of trajectories which is well-suited for evaluating an individual's behavior in light of the perceived,

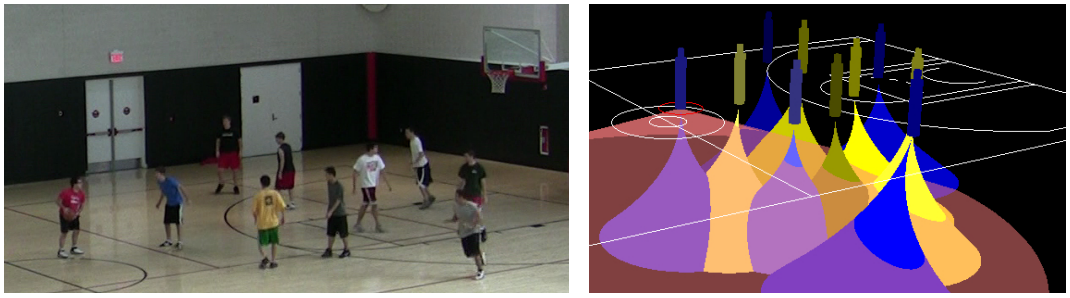


Figure 1.3: **Multi-Agent Interaction Analysis - Proposed Approach.** Interactions in a multi-agent environment are analyzed using *kinetic accessibility*, a spatiotemporal framework for considering individual perception and behavior in the context of agent and object trajectories.

and thus anticipated, movements of other agents.

The arena in which we apply our new approach is the sport of basketball, where two five-player teams each try to get the playing ball into a small basket or hoop at their end of the playing field. A reasoning framework based on Markov Logic Networks [112] allows integration of the domain-specific rules of basketball, low-level visual cues and quantitative kinetic accessibility-based measures. Such measures describe the feasibility of actions such as an agent completing a pass to a given teammate, allowing recognition of events such as passes, shots and turnovers.

1.4 Topology Estimation in Camera Networks

While there are fascinating new challenges introduced by the growing ubiquity of mobile video-recording devices (*e.g.* cell phones, tablets, netbooks), this work focuses on a long-standing challenge for static networks of cameras, one that is par-

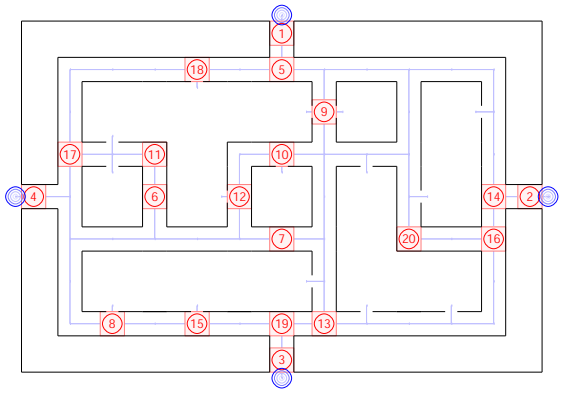


Figure 1.4: **Estimation of Camera Network Topology.** This illustration shows an office environment with twenty cameras placed throughout it. The objective is to leverage the collection of observations throughout the network to recover the topology of the network.

ticularly relevant in the surveillance domain. Prerequisite to identifying events of interest or interpreting an agent's activities is the ability to track observed individuals over extended periods of time. If cameras overlap and are appropriately calibrated, tracking between cameras is relatively straightforward. In many circumstances, however, sensors have disjoint fields of view, due to factors such as large coverage areas, target resolution requirements or physical building layouts, and it is in such non-overlapping networks that identity maintenance is so difficult.

We introduce two methods for recovering the topological relationships between non-overlapping cameras. While the latter approach uses a distributed or decentralized architecture, the first system collects all of the observations throughout the network and processes them centrally. This technique aims to partition the set of observations into unique object trajectories, each observation assigned to the correct

individual's sequence of observations. As exhaustive consideration of the partition space is intractable, partitions are considered incrementally, adding observations while pruning unlikely partitions. A probabilistic graphical model is employed to determine each candidate partition's likelihood, balancing the consistency of observed appearances with current predictions of camera adjacency. While prior solutions generally recover only first-order camera adjacency, the proposed trajectory-based method provides higher-order statistics, yielding resilience to camera failure and the potential for improved tracking performance between cameras.

The decentralized approach is amenable to a network of smart cameras which couple the ability to communicate with other camera nodes with onboard processing capabilities. Under this model, the global network topology is computed in parallel using sequential Bayesian estimation on a modified multinomial distribution. A crucial aspect of this system is the integration of an information-theoretic appearance model in which easily-confused objects are discounted and salient or conspicuous appearances are weighted more heavily. As the system acquires additional observations, the global topology estimate converges toward the correct underlying representation.

This work was previously published [39, 38].

1.5 Overview of the Thesis

The dissertation is organized as follows... We tackle the problem of subordinate categorization in Chapter 2, using the avian domain as a testbed for the proposed approach. Our approach couples a volumetric poselet-based detection model with a pose-normalized part appearance model. Chapter 3 introduces *kinetic accessibility*, our proposed approach for reasoning about agent interactions based on their trajectories, and describes its application in the domain of basketball. Leveraging trajectories at a more global scale, Chapters 4 and 5 present two approaches to the problem of estimating camera network topology, equivalently, the task of deriving a model describing how objects transition between cameras. While one approach is centralized and the other distributed, both approaches rely heavily on the individual observations, utilizing factors such as object appearance and temporal constraints. Chapter 6, the conclusion, both provides a summary of the contributions contained in this work and suggests some research directions for future investigation.

Chapter 2

Subordinate Categorization Using Volumetric Primitives and Pose-Normalized Appearance

2.1 Introduction

In recent years, the computer vision community has devoted extensive efforts toward the development of computational techniques for object recognition. These efforts, however, have focused almost exclusively on basic-level categories; relatively few have addressed the broad continuum of fine-grained or subordinate categories which lies between the two extremes of individuals (e.g. face recognition, biometrics) and basic-level categories (e.g. Caltech-256 *etc.*), see Figure 2.1.

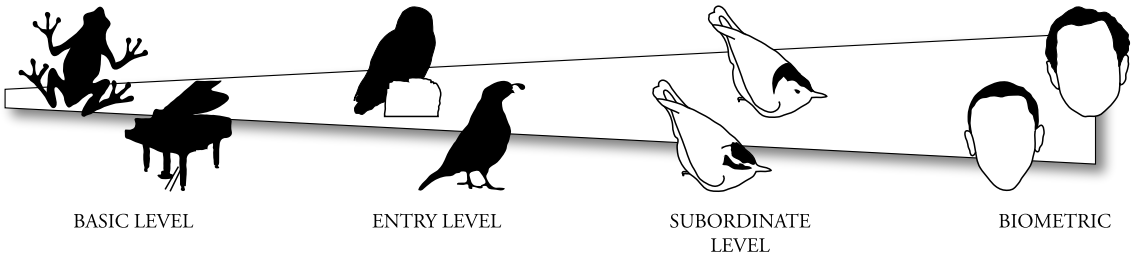


Figure 2.1: **Categorization Spectrum.** This illustration shows the range of categorization levels, ranging from basic-level categories (*e.g.* frog vs. piano) to the individual/biometric level (*e.g.* face recognition). Between these extremes we find entry-level categories (*e.g.* more readily identified as "owl" and "quail" than "bird") and subordinate level categories (*e.g.* individual species).

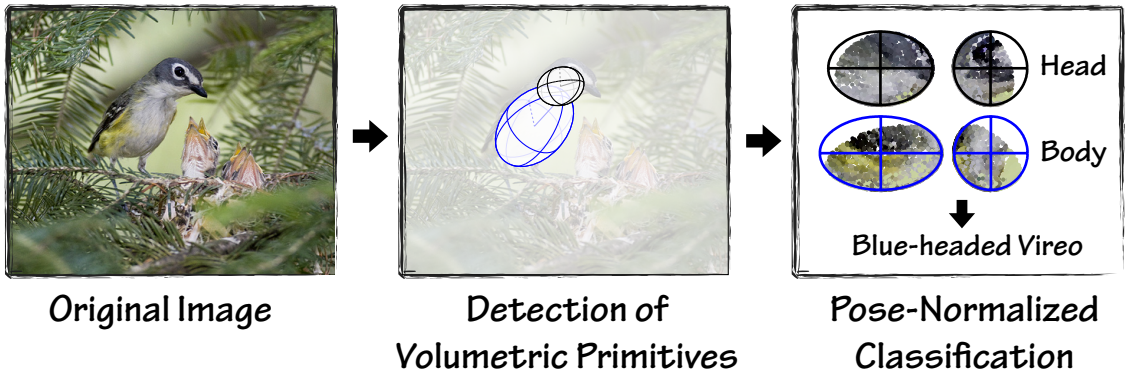


Figure 2.2: **Overview of the Proposed Approach.** Basic-level categories are modeled by a configuration of volumetric primitives or parts. Detection recovers these parts and enables application of a pose-normalized appearance model for classification within a taxonomy of subordinate categories.

In cognitive psychology, Rosch *et al*. [113] proposed that, whereas basic-level categories are principally defined by their parts, subordinate level categories are distinguished by the differing properties of these parts. This theory suggests that the capacity to differentiate subordinate categories hinges not only on the successful recognition of individual parts but, perhaps more particularly upon understanding how these part "properties" vary across subordinate categories. While recent advances on part-based and attribute-based recognition are promising, general and view-independent identification of part-specific attributes in novel images remains somewhat elusive.

We tackle the problem of subordinate categorization, proposing a solution that simultaneously addresses the challenges of localizing and describing the class-defining parts. Our approach (see Figure 2.2) builds upon the Poselet detection

framework recently proposed by Bourdev *et al.* [12, 13]. The strength that we see in this framework is that, in theory, the model allows for specific types of training annotations to be recovered from detections in test images. Our approach is also motivated by Biederman's theory of non-accidental arrangements of geometric primitives [8, 9]. We use a simple configuration of volumetric primitives to represent the basic-level class. Then, following Rosch *et al.*, variations in the shape, configuration and appearance of these volumetric parts provide the basis for subordinate discrimination.

Our proposed approach contributes three main innovations:

(i) a framework, based on Poselets, for detecting volumetric part models, used both to find the basic-level object and to convey information about part shape and configuration;

(ii) a pose-normalized appearance model (similar to representations such as Active Appearance Models [25] and Morphable Models [70] used in the domain of faces) which is used to effectively compare part appearances in a test image to those of subordinate category training examples; and

(iii) a classification model, based on Stacked Evidence Trees [93], which aggregates information about part properties (shape, configuration and appearance) and leverages the underlying taxonomy.

We demonstrate experimentally that the proposed approach enhances the performance for view-independent recognition of subordinate categories.

2.2 Related Work

The problem of subordinate categorization has been previously examined. Bar-Hillel *et al.* [5] performed experiments on two subclasses for each of six basic categories (e.g. Grand vs. Upright Pianos). Nilsback and Zisserman [98, 99] considered subordinate categories of flowers (introducing the 17- and 102-category Oxford Flowers datasets), whereas Martínez-Muñoz *et al.* [93] considered subordinate categorization of stonefly larvae, a domain which exhibits tremendous visual similarity. These approaches focused primarily on discriminative learning of image features, an approach that does not generalize for view-independent categorization of part-based objects that exhibit significant pose variation.

There are various methods that have been proposed for learning part-based object representations. Constellation models [19, 128] and their computationally attractive variants [27, 47] are composed of a set of local part detectors together with one or more probability density functions describing the parts' relative locations. Felzenszwalb and Huttenlocher [44] proposed an efficient framework implementing Fischler and Elschlager's Pictorial Structure model [49], which represents an object by a collection of parts, interconnected as if by elastic springs. This Deformable Part Model has culminated in Felzenszwalb *et al.*'s recent work using Latent SVMs [43] to discriminatively train class-specific object detectors. Ferrari among others have explored the use of contours in object representation [48]. While these models perform well for objects that exhibit minimal articulation or pose variation, they are unsatisfactory for objects with high intra-class variability or significant articulation.

There is also a growing body of work that seeks to leverage similarities between categories to improve recognition performance. We consider two principal areas of interest: first, class taxonomies or hierarchies and, second, attribute-based models. Unsupervised hierarchical approaches range from constructing latent topic hierarchies [6] to sharing classifiers [4] or visual parts [122] to constructing efficient classification trees [59, 92]. Each such approach provides insights or advances toward efficiently solving basic-level classification. These unsupervised approaches, however, cannot be readily applied to the problem of distinguishing closely-related subordinate categories which, by definition, share a common set of parts and yet can have both subtle and drastic appearance variation.

Techniques that leverage the semantic class hierarchy should possess an inherent advantage over those that do not. Supervised methods that utilize such information (as contained in WordNet for example) include the sharing of training examples across semantically similar categories [46] and combining information from different levels of the semantic hierarchy [137]. Deng *et al*. [29] consider exploiting the semantic hierarchy in the context of more than 10,000 categories (using the ImageNet [30] dataset).

A growing interesting in attribute-based recognition has produced some notable advances. Representative work in this area includes Farhadi *et al*. [36, 37], Kumar *et al*. [78] Lampert *et al*. [79] and Wang and Forsyth [127]. These techniques often learn discriminative models from attribute-labeled training data and subsequently apply the learnt models to estimate the appropriate visual attributes present in a test image. Attribute-based models are particularly well-suited for ad-

dressing the one-shot learning problem (previously considered in [41, 42, 45, 94] among others). Note that while these approaches are effective for the recovery of object level attributes such as brown, furry, spotted and even four-legged, they are generally insufficient to model subtle differences between parts necessary for subordinate categorization.

An interesting exception is the innovative work of Branson *et al.* [17] which proposes improving recognition accuracy by interleaving computation with attribute queries made to a human subject. This method performs effective, though not automatic, recognition in a large, 200-category bird dataset [129][1]. Additionally, in the context of subordinate categorization, the attribute-based work of Berg *et al.* [7] is also of interest as it attempts to discover (and localize) visual attributes which can be used to differentiate classes within a basic-level category (e.g. stiletto, running shoe, sandal, *etc.*). This approach is somewhat limited, however, in that its training data is segmented from any background and also must be in a similar pose/orientation.

Before proceeding to describe our approach, we first visit the theory initially put forth by Marr and Nishihara [91] and later extended with Biederman's geons [8] which suggests that object perception is largely governed by recognition of three-dimensional parts in particular configurations. While subsequent research has questioned certain aspects such as view invariance [120], this theory of perception as the search for arrangements of non-accidental structures has survived. Biederman *et al.*

[1]Additional details on the CUB-200 dataset can be found in Section 2.6 which describes our experiments.

revisited it in the specific context of subordinate-level classification [9]. This theory provides support for the proposed approach which models a basic-level category with geometric primitives, and then couples the statistical variation of the parts' shape and arrangement with their appearance to represent subordinate classes.

2.3 Subordinate Categorization in an Avian Domain

We begin by considering more closely the problem of subordinate categorization, highlighting some of the ways it differs from basic-level categorization. The seminal work of Rosch *et al.* [113] provided experimental evidence in support of a distinction between levels of abstraction within a taxonomy: superordinate, basic, and subordinate (in decreasing order of inclusivity). Rosch *et al.* contend that basic-level categories generally possess the highest cue validity $P(category|cue)$, as superordinate-level categories, being more inclusive, have fewer attributes in common and subordinate-level categories share most of their attributes with contrasting subordinate categories.

2.3.1 Basic- and Subordinate-Level Categorization

Objects within a superordinate category tend to share common material and/or functional properties (sensory-motor "affordances" to use Gibson's terminology [53]). In contrast, a (and perhaps the) key characteristic of categories at the basic-level is shape. Rosch *et al.* include in their definition of shape "the structural relationship of the parts of an object to each other - for example, the visual representation of

the legs, seat, and back of a chair and of the way in which those parts of the chair are placed in relation to one another."

This notion of basic-level shape as a fixed set of parts in an expected arrangement agrees strongly with Biederman's theory of Recognition-by-Components [8] which suggests that a category may be represented by volumetric components or primitives called "geons" (blocks, cylinders, cones, etc) in a particular configuration. While Biederman's theory presents a broad perspective on the human recognition process (edge extraction and parsing, identification of components, matching to known configurations, object identification), we focus on this underlying representation of basic-level categories: a configuration of volumetric parts.

This basic-level representation is intuitive for many natural categories. Objects within a category such as automobiles, for example, share a common set of parts: a main chassis (engine, passenger compartment, trunk) and four wheels. Similarly, trees have a trunk, branches and a leaf canopy (the roots are generally not visible) while dogs have a head, body, four legs and a tail. Within such categories, the configuration and "connectivity" of these parts is generally highly constrained.

Differentiation amongst subordinate categories (*e.g.* between sports cars and sedans or even different brands/models), however, must rely on more than the simple presence and configuration of these parts. We thus consider *properties* of these parts, including quantitative properties such as shape variation (aspect, relative size) or structural relationships (relative position/angle) and qualitative appearance properties such as color, material and texture.

We have selected birds as the domain for our experimental evaluation for a

variety of reasons. There are several basic-level categories for which vision datasets include many subordinate classes. None are larger than the recently introduced Caltech/UCSD Birds dataset (CUB-200) [129] which includes 200 distinct avian species. While some categories are readily identified by their unique shape, pose, or appearance, the distinctions between other categories are very subtle. Due to highly variable appearances and articulation, birds are also extremely challenging to even detect, consistently the most difficult across the 20 standard VOC categories. Ultimately, however, the principal motivation for our use of birds as a domain in which to explore subordinate visual categorization is their suitability for our pose-normalized representation.

2.3.2 Pose-Normalized Appearance Representation

Following Rosch's prototype theory which defines basic-level categories by their parts, we distinguish subordinate categories based on both the geometric shape and the photometric appearance properties of these basic-level parts. In describing our appearance representation, we begin with a basic-level object, represented as constellation of volumetric parts. The detection process provides estimates for each part's respective parameters: location, scale and orientation. The geometric shape and arrangement properties can be used to influence categorization. Within the domain of birds, taxonomic guidance by shape is intuitive; even individuals with minimal expertise in recognizing birds can correctly assign a silhouette to its respective family (*e.g.* duck, heron, hawk, owl, songbird, *etc.*).

As far as the volumetric part appearance properties, the primary difficulty is relative pose variation with respect to the camera, an issue that complicates the comparison of part appearances observed from different angles. To overcome this challenge, we propose a pose-normalization approach leveraging the detected volumetric parts. Fundamental to our approach, this technique imposes a surface parameterization on the volumetric part, the parameterization serving as a basis for a non-parametric representation of the surface. Comparisons between images are made not in image space, but on a distribution of patch descriptors in the parameterized space of estimated surface normals.

In our part model, we have two ellipsoids, one for the head and one for the body. For a given ellipsoid, we take the pose parameters: ellipsoid center (x,y), scale (cross-section and axial aspect ratio) and orientation (represented as a quaternion) and generate the transformation that maps points on a unit sphere onto the ellipsoid's surface. The inverse of this transformation allows us to map image points (that are within the ellipsoid's silhouette) back onto the unit sphere. Instead of parameterizing in the sphere's space, we randomly sample points on the sphere, transform them onto the ellipse's surface and compute their normals (using the inverse transform), ensuring that they are visible. We can then find the tangent plane at this location and for a square patch, tangent to the ellipse, centered at a sampled point, we locate the corners of this tangent patch and project them back to the image. We then take this rectangle in the image and warp it's pixel contents onto a small image square. We can then extract a descriptor from this image square (we use a color-SIFT descriptor at the nominal orientation). We couple the appearance

Figure 2.3: **Pose-Normalized Appearance Descriptor (PNAD).** Locations are sampled from the surface of a volumetric primitive; at each sampled location (parameterized as a surface normal), an image patch of the local tangent plane is extracted and encoded with a feature descriptor (SIFT in our implementation); the final pose-normalized appearance descriptor, or *PNAD*, is obtained by concatenating the patch feature descriptor with the parameterized surface location.

information together with the parameterized location by concatenating the normal vector onto the appearance descriptor. This process is shown in Figure 2.3. After sampling several such points, we obtain a non-parametric representation of the visible portion of the ellipsoidal part.

2.4 Volumetric Object Localization

As suggested in the introduction, the primary requirement for successful differentiation of subordinate categories is an ability to find parts and understand how these parts vary (or alternatively, how the "properties" of these parts vary) across different subordinate categories. To address the problems of localizing and describing the class-defining parts simultaneously, we adopt the Poselet framework recently

proposed by Bourdev *et al.* [12, 13], but with an object model comprised of volumetric primitives instead of 2D or 3D keypoints. We provide a brief description of the approach while highlighting changes needed for our volumetric implementation.

2.4.1 Birdlets: Volumetric Primitive Templates

While the Poselet framework represents a basic-level category as a constellation of 2D keypoints, our approach creates "Birdlets", templates based instead on solid volumetric primitives, consistent with Biederman's notion of basic-level categories as arrangements of 3D geometric primitives. Where the former technique estimates the image location of each keypoint, the utility of using the volumetric parts lies in its potential to estimate various geometric quantities for each of the volumetric elements that collectively comprise the basic-level category model. Examples of such geometric attributes (or "properties") include part location, size/aspect, and orientation, and can encode intrinsic category characteristics such as the cross-section or aspect of a bird's body relative to the size of its head.

This volumetric model is particularly well suited for birds, as the avian counterparts for interior mammalian joints (*e.g.* shoulders, elbows, hips, knees) are often obscured by plumage and thus very difficult to locate in a typical image. Moreover, skeletal features such as many of the keypoints used in the Poselet model capture body part proportions (e.g. cross-section, aspect) within the object poorly. The proposed model, therefore, includes visible point features such as the beak tip, eyes, wingtips, feet, and tail, only to assist in configuration alignment, but the model fo-

cuses on its two volumetric components. The bird's head and body are represented by prolate ellipsoids (a sphere stretched along one axis), each having 7 parameters: image location(x,y), 3D-orientation(a 3-DOF quaternion), and scale(circular cross-section and axial length). Where one could try to model a bird with additional primitives, this simplified version (or "partial version" [8, p. 131] as Biederman calls it) captures the essence of shape and enables the pose-normalized appearance representation.

2.4.2 Training and Detection

The Poselet framework requires images annotated with configuration landmarks (2D or 3D keypoint locations in Bourdev *et al.*, location, orientation and scale of volumetric primitives in our case). These annotations serve to help find training examples that share similar local pose or configuration (it need not be fully identical, just for the part or keypoints in question). In this manner, images depicting similar poses relative to the camera are grouped together.

Birdlet training takes a certain *base* training image and determines a *selection window* overlapping some subset of the volumetric parts (in our case, this could be the head, the body or both). Next, the *pose distance* to each of the other training images is computed, based on the similarity in parameters for this subset of parts (*i.e.* can the two images be registered to one another such that the parts align well). Specifically, this distance is computed using terms for rotation (geodesic distance on 4D surface of quaternion rotations), scale (computed on cross-section and aspect

Figure 2.4: **Birdlet Training.** This illustration shows the process of selecting the positive training set. Training examples with similar pose (within the window of interest) are identified and an aligning similarity transform determined. Then for each training example, a HOG description is extracted for the aligned patch. These features collectively form the positive training set.

after scaling to equal volume) and translation (generally ignored as single ellipsoids can be brought into precise alignment as can the dipoles formed by ellipsoid centers).

As depicted in Figure 2.4, the $n-1$ closest training images are selected (while the figure shows 10 for ease of illustration, we nominally use $n = 50$) and the similarity transform to align them to the base image is determined. With this transform, the parts now line up (as best as can be done with the 2D similarity transformation) and the corresponding image features should now be well aligned also. Now, for each of n training images (the base and the $n-1$ closest in terms

of pose distance) which have been transformed into alignment, the pixels in the selected window are mapped into a canonical rectangular *patch* (96×64 in our case) and a HOG vector [28] is extracted (the concatenation of HOG features across 8×8 blocks). These n HOG vectors are used as positive examples, together with a much larger set of negative HOG vectors (extracted from other random windows in the training data), are used to train an SVM classifier to discriminate this birdlet from background patterns. Like Bourdev *et al.*, we use a retraining stage, collecting false positives predicted by the initial classifier and feeding these as additional negative examples in order to train the final classifier for this birdlet.

For detection, our birdlet classifier will evaluate patches in a test image using a sliding window (scanning over locations and scales), responding with a probability of how similar each scanned patch appears to the positive examples that the classifier was trained with. Windows with high response probabilities are labeled as *activations* for the given birdlet.

The great benefit that we saw in the framework of Bourdev *et al.* is that the birdlets we train facilitate detection, but moreover provide information about the pose or part-configuration. A birdlet activation provides an estimate or vote toward the parameters of those volumetric parts that overlapped the birdlet's selection window. Hence, whereas other techniques typically learn a model on latent parts, the birdlet model maps the image patterns within the selection window to the semantically meaningful volumetric primitives, inherently providing a level of visual correspondence across instances (and views).

Many such birdlet templates are trained, binding images cues from the training

set with their counterpart volumetric part annotations. The collection of birdlets is then applied to a test image producing a set of birdlet activations. Each activation has an associated probability (derived from the corresponding classifier's response) as well as the distribution on part parameters it acquired during training (this distribution is a simple tabulation on the parameters of the overlapping parts once aligned). The birdlet normalizes the distribution relative to the height of the patch, such that for a given activation window, the normalized location and relative size information can be scaled up the activation window, thus converting it to a prediction in the test image. Our implementation uses a non-parametric (kernel density estimate) density to represent each ellipsoids 7-D parameter space.

The final step is to cluster the set of activations into one or more *final detections* with the corresponding volumetric part estimates. The approach that we have taken for this clustering is to compute the pairwise consistency of activation, determined by symmetric K-L divergence between the parameter distributions of the corresponding parts shared by the activations' respective birdlets. We take the pair of activations with the highest consistency (and activation probability or response) and draw the volumetric parts' parameters from their distributions. In theory we can sample from the combined distribution, however, in practice, we found it effective to predict the parameters of each birdlet's base training image (for some birdlets, there are small clusters of examples with similar pose, and thus only a few training examples that share similar parameters).

2.5 Integrated Classification

Our approach uses an integrated classification technique based on the Stacked Evidence Trees model proposed by Martinez-Muñoz *et al.* [93]. The authors describe this approach as an alternative to dictionary learning, being instead a way of "discriminatively structuring the evidence in the training set". This model (see Figure 2.5) relies on a Random Forest [18] constructed such that all leaf nodes of the constituent random trees are required to have a specified minimum number (*e.g.* 20) of training samples. In this manner, when a query sample is passed through a random tree and reaches a particular leaf node, the tree returns the distribution across class labels corresponding to training examples that reached that node. For a given image, features are extracted densely. As these features are dropped through the trained random forest, the class label distribution vectors are collected and aggregated into an "evidence" vector, each feature effectively voting for the category of the image. A second-stage ("stacked") multiclass adaboost classifier is then applied to the class distribution evidence vector, producing the final category prediction.

The Stacked Evidence Trees model was selected principally for the way that it complements the Pose-Normalized Appearance model, providing an attractive solution to the problem of varying surface visibility. In general, a volumetric primitive has only half of its surface facing the camera, the remaining half is not visible. As the visible/occluded portions are different for each image (*e.g.* a bird facing the camera vs. facing left vs. facing right), it is desirable not only to map the visible portions into a common (pose-normalized) space, but moreover, to effectively mask

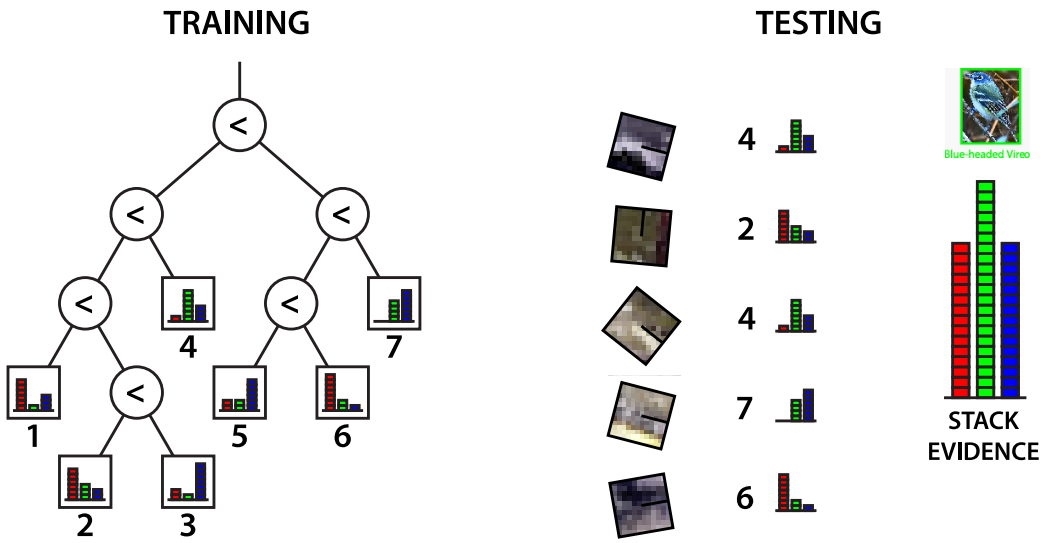


Figure 2.5: **Stacked-Evidence Tree Classifier.** The stacked-evidence tree model is based on a random forest constructed from training data. For a test image, various features are extracted and each one is dropped through the trained random forest. As a given feature reaches a leaf node, the class distribution across the training instances for that node is added to the stacked evidence histogram. Accumulated across the test features, this stacked evidence histogram (vector) is used to determine the test image's class.

which part(s) of this common space should be used for classifying each given image, see Figure 2.6.

As described earlier, the Pose-Normalized Appearance space allows us to compare corresponding parts. Specifically, a PNAD (Pose-Normalized Appearance Descriptor) feature, see Figure 2.3, couples local appearance information with parameterized surface location. However, due to the issue of feature visibility, one cannot simply quantize this joint appearance/surface location space and use a bag-of-words

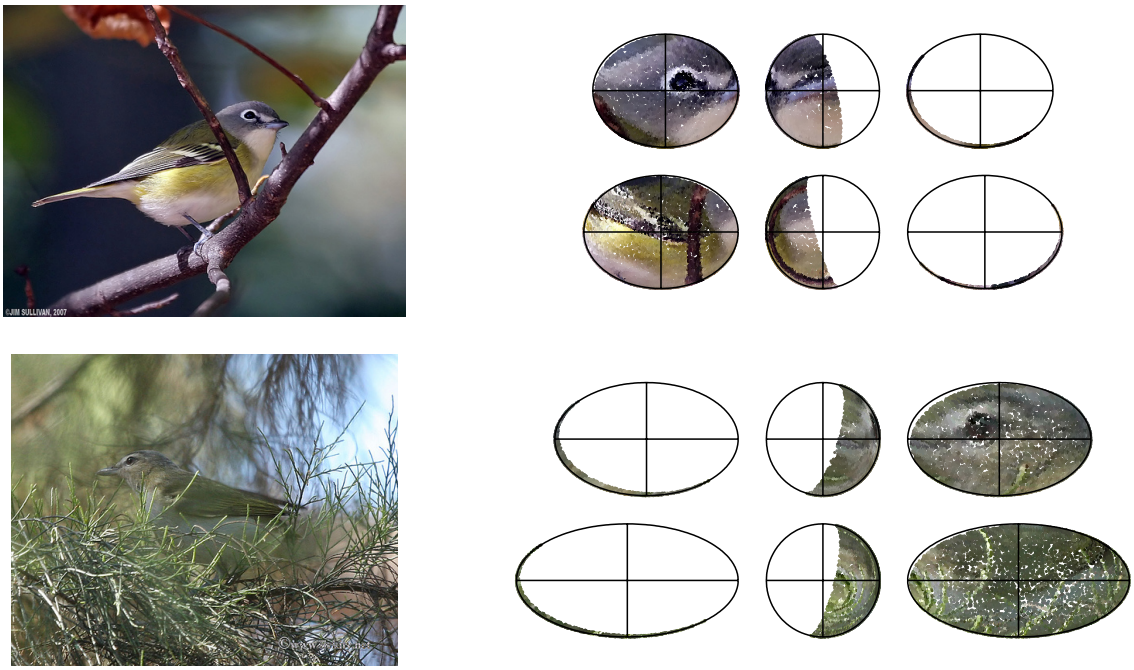


Figure 2.6: **Ellipsoids and Feature Visibility.** For each ellipsoid (head/body), only half of the parameterized surface is visible. Coupling pose normalization with the stacked-evidence tree classifier facilitates comparison of parts whose visible subsets differ.

approach for classification. The Stacked Evidence Tree on the other hand becomes a highly-efficient retrieval tool, taking a test feature and finding a set of training features (namely those in the corresponding leaf nodes) that are similar both in appearance and surface location, and ultimately returning the class label distribution across this similar set.

An appealing characteristic of the Stacked classifier is the ability to combine multiple feature types by merely concatenating various evidence. In our case, we view this as the means to combine part appearance (PNADs) together with other potential sources of discriminative information. We consider combining shape and

arrangement parameters (*e.g.* part cross-section/aspect, relative sizes/orientations between parts, *etc.*) as well as taxonomic training data.

One additional potential source of information which we are not currently using is the birdlet activations that contributed to the detection. When a given birdlet is trained, the other examples selected as positive patches (based on similar configuration) may collectively convey information at test time about the category of detections involving a high-probability activation of the birdlet in question.

2.6 Experimental Results

Now that we have described detection of volumetric primitives, pose-normalized appearance representation, and integrated classification, we present some experiments in support of this framework.

2.6.1 Dataset, Implementation Details, *etc.*

First utilized by Branson *et al.* [17], the Caltech-UCSD Birds 200 dataset [129] (see Figure 2.7) currently offers the largest number of subordinate categories for a single basic-level category. We organized the entire dataset into its proper taxonomic hierarchy (order, family, genus, species) and then selected two families to fully annotate with both 2D keypoints and 3D volumetric primitives (ellipsoids), the vireo and woodpecker families. These annotations, together with near-duplicate groupings (so that near-duplicates do not straddle test-training splits), will be made publicly available to other researchers. While many annotation tasks are well-suited

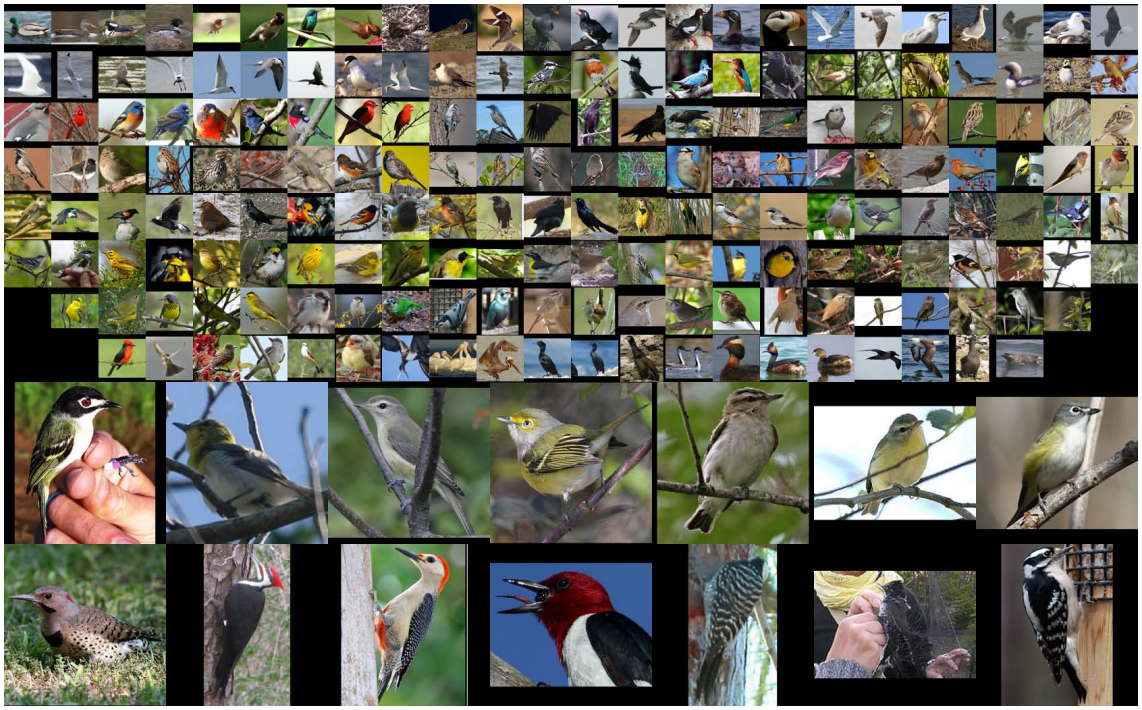


Figure 2.7: **Caltech-UCSD Birds 200 Dataset.** This illustration shows images of the 200 species that are included in the CUB200 dataset. The subset of species which we have labeled with volumetric annotations and subsequently used in our experiments are shown in the lower portion at a larger scale.

to crowdsourcing, we felt that proper annotation of the ellipsoids was non-trivial and accordingly have a smaller dataset than would be desirable.

As the authors of [13, 12] have only released their code for detection with a pre-trained human detection model, we had to reimplement the extensive Poselet framework in its entirety. In our birdlet implementation, we utilized LIBSVM [22] together in conjunction with Platt's algorithm [86] for converting SVM scores to probabilities. The random forest used for integrated classification was adapted from the Weka [61] machine learning package.

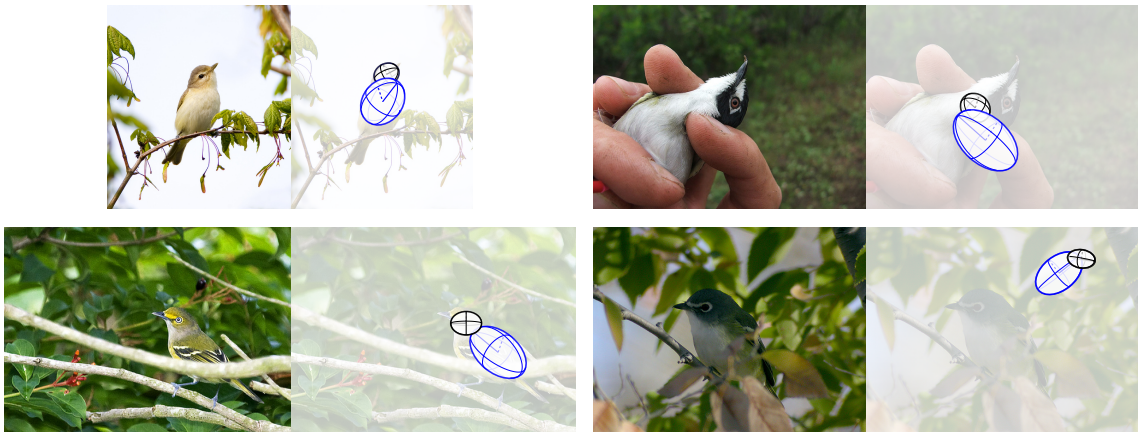
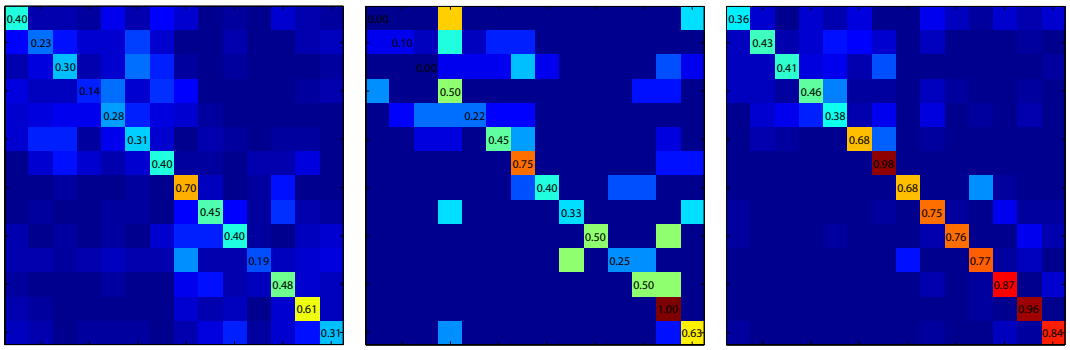


Figure 2.8: **Example Volumetric Primitive Detections.** Here are four representative detections. In the top-left and lower-left images, the bird is detected and localized with reasonable accuracy. The top-right and lower-right images depict false positive detections, however. In the top-right, the birdlets incorrectly interpreted a finger as the bird's body, and the lower-right image is typical of false detections at the incorrect scale and location.

2.6.2 Volumetric Part Localization

Before we can consider our primary objective of subordinate categorization, we evaluate the detection of our volumetric part model. To train the birdlet model, we used a training split that included 15 images of each category (together with their mirrored annotations) for a total of 420 training images/annotations. The resulting birdlets (we train a set of 100 birdlets) are applied toward detection on the remaining 492 test images.

Some examples detection results are illustrated in Figure 2.8. The two shown on the left are accurate detections relative to the ground truth, those on right are mistakes. Comparing the detected parts to the test images' ground-truth annota-



(a) Baseline - PHOW/SVM (b) Detected parts (top 20%) - PNADs (c) Ground truth part locations - PNADs

Figure 2.9: **Classification Confusion Matrices.** Depicts the classification for the following techniques (a) the PHOW/SVM Baseline (37.12% MAP), (b) the PNAD-RF performance on the top 20% of detections (40.25% MAP), and (c) the PNAD-RF performance on the ground truth part locations (66.58% MAP).

tions, we find that while many of the detections have significant errors (*e.g.* those in Figure 2.8), many detections are reasonably accurate. As it is pointless to try to classify these false detections, we run the classification on the more accurate detections as described below.

2.6.3 Subordinate Categorization

We now describe our subordinate categorization results. We establish a baseline using a pyramidal histogram of color-SIFT words approach (using the VLFeat toolbox [126] implementation), providing it the ground-truth bounding box to assist in localizing the bird. The performance across test-training splits is 37.12% mean-average precision. Anecdotally, this approach is comparable to the multiple-

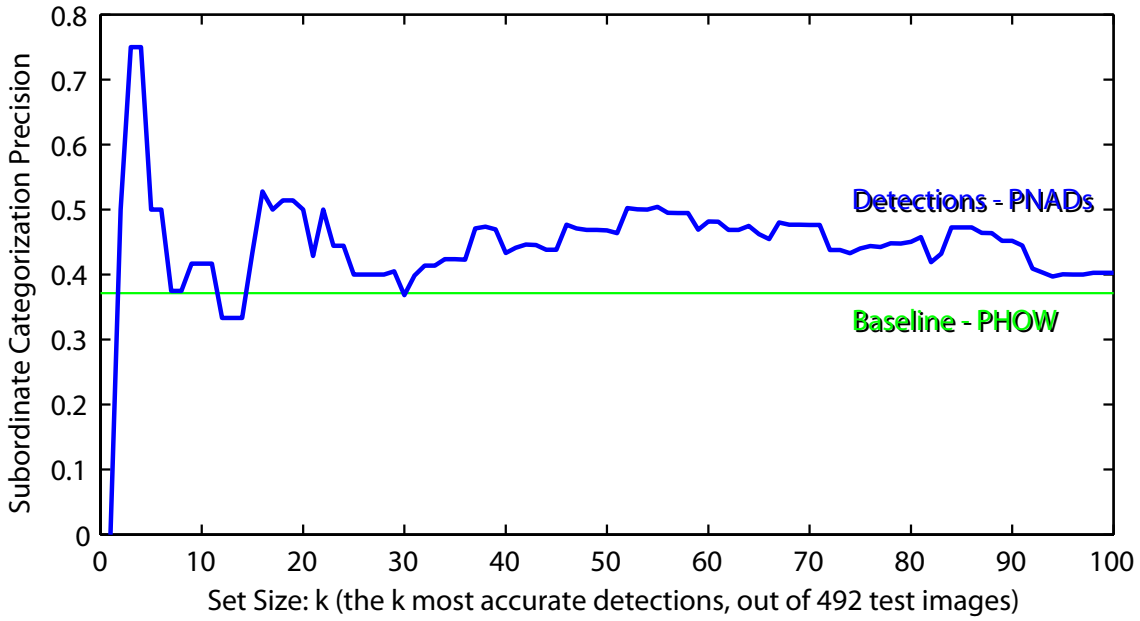


Figure 2.10: **Classification of Volumetric Detections.** For the k top-ranked detections, this plots the corresponding PNAD-RF classification performance (using mean-average precision).

kernel learning approach used by Branson *et al.* [17] (37.02% on this same subset of categories). Figure 2.9(a) shows a confusion matrix for the Baseline PHOW/SVM classifier. Next we turn to Figure 2.9(c), which illustrates the potential performance of the PNAD-RF (Pose-Normalized Appearance Descriptor coupled with the Random Forest classifier) technique. This approach achieves a mean-average precision across the categories of 66.58% by computing the PNAD features on the ground truth ellipsoids.

Our objective then is to evaluate the same PNAD-RF method on the estimated ellipsoids from our real detections. Figure 2.10 shows the mean classification accuracy for sets of increasing size (see Figure 2.11 for detailed results on the 24

most accurate detections). The plot shows that, for the most accurate 20% of the detections, the subordinate classification accuracy was above the baseline performance. For the top 10% of detections, accuracy was as much as 10% higher than that of the baseline. In Figure 2.9(b), the confusion matrix for the most accurate 20% of the detections is shown, a mean-average precision of 40.25%. We believe that the performance could be even higher if the birdlet training had a larger pool of training examples to draw upon.

Figure 2.11: **Visual Results for Top-Ranked Detections.** Results for the 24 most accurate detections. Correct classifications are framed in green, incorrectly classified detections are framed in red with the predicted class depicted below.

Chapter 3

Modeling Multi-person Interaction using Kinetic Accessibility

3.1 Introduction

This paper addresses the problem of understanding the behavior and interactions between many individuals in a video of a highly dynamic scene. From a given individual's perspective, day-to-day behavior is largely governed by two factors: first, the tasks or objectives which the individual intends to complete; and second, the perception of one's surroundings and of other people and objects therein. Well more than a half-century ago, Lewin postulated [84] that behavior is a function of a person and their environment, formally $B = f(P, E)$. Depending on the task to be performed, behavior may be more dependent on the individual person or on the dynamic circumstances around them. To motivate an emphasis on perceptual tasks, where behavior is heavily influenced by the environment and by surrounding objects, we consider Gibson's theory of locomotion. Gibson described [54] automobile driving as

> "...a *a perceptually governed series of reactions* by the driver of such a sort as to keep the car headed into the middle of the field of safe travel."

The driver's moment-to-moment actions are determined only in part by the person's intended destination, but are made particularly in reaction to both the static

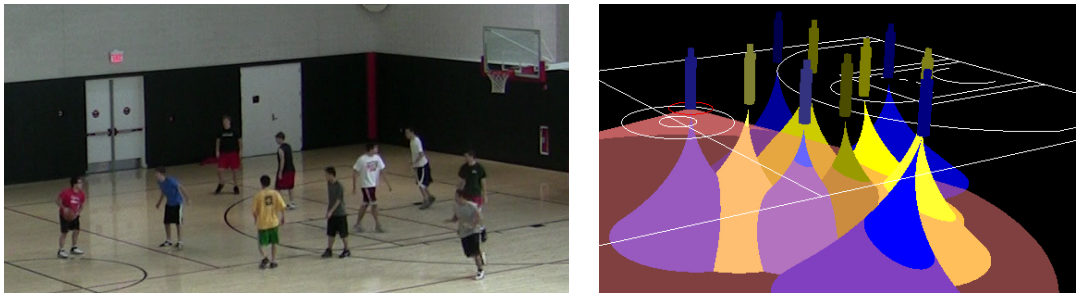


Figure 3.1: Interactions in a multi-agent environment are analyzed using *kinetic accessibility*, a spatiotemporal framework for considering individual perception and behavior in the context of agent and object trajectories.

surroundings (the road, curbs, parked cars, trees) and the dynamic elements of the environment (stop lights, traffic, and most particularly other nearby drivers). Our work focuses on analysis in the "reactive" domain of continuous team sports where agent behavior is influenced partially by structured game objectives, but particularly in response to both static (court/field boundaries, baskets/goals) and dynamic elements (teammates and opponents).

Our primary contribution is a framework for analyzing multi-agent interactions based on the perception of inter-agent motion. Rather than approaching these interactions with techniques such as grammars or local action analysis, we focus on an agent's reactions to other agents' movements. To model each individual, ground-plane trajectories are associated with local per-player information (such as possible ball possession). To model the individual's perception of their environment, we introduce *kinetic accessibility* measures which model the time-varying spatial relationships between individuals. These measures are useful for representing the feasibility of events such as the existence of an unobstructed path in space-time

allowing one player to throw the ball to a teammate. As both the local features and the accessibility measures are inherently uncertain, a probabilistic logic-based reasoning framework (Markov Logic Networks [112]) is employed to perform inference. Evaluation is performed on a new publicly available basketball dataset [3], identifying time intervals in which perceptually-driven events such as passing to a teammate or taking an open shot occur. The improvement achieved with the proposed kinetic accessibility measures demonstrates their utility in modeling multi-agent interaction in highly perceptual environments.

3.2 Background and Related Work

A highly active area of research within the multimedia and computer vision communities addresses the detection and recognition of events, individual actions and the broader activities they comprise. While an exhaustive review of such techniques is infeasible, the interested reader is referred to survey papers by Aggarwal and Cai [1], Turaga *et al.* [124] and most recently Poppe [108].

We consider representative work in the following three areas of the literature: (*i*) single-person action recognition, (*ii*) multi-agent activity recognition, and (*ii*) sports analysis and event recognition in sports videos.

3.2.1 Action Recognition

In the category of single-person action recognition, Bobick and Davis [10] introduced motion-history and motion-energy images, features based on temporal

aggregation of figure silhouettes. Subsequent silhouette-based approaches include Yilmaz and Shah [130], who use differential geometry to describe points on the surface of the volume created by stacking the silhouette contours, and Gorelick *et al.* [58], who use features which characterize the interior of this volume derived using a solution to the Poisson equation.

Other approaches use features derived from the local motion (optical flow) field. Such work includes Efros *et al.* [34], Dollar *et al.* [33], Laptev *et al.* [80, 81] and Ke *et al.* [73, 74], among others. These methods sample the optical flow field either sparsely (via interest point detection) or densely, and then use machine learning techniques such as SVMs or Adaboost to select discriminative features. Others such as Parameswaran and Chellapa [102], and Junejo *et al.* [71] focus on identifying space-time features of an action that are view invariant.

3.2.2 Multi-Agent Event Recognition

Another body of work treats the problem of recognizing activities, where one or more people perform a sequence of actions. Several approaches utilize dynamic Bayesian networks (DBNs) such as hidden Markov models (HMMs) to learn and recognize such action sequences. The works of Brand and Kettnaker [16], Oliver *et al.* [101], and Gong and Xiang [56] all use variants of the hidden Markov model, respectively using HMMs, coupled HMMs (CHMMs) and dynamic multi-linked HMMs (DML-HMMs). Park and Agarwal [103] use a DBN to model the evolution of joint body-poses between two interacting individuals, recognizing pairwise-actions such

as kicking, shaking hands, approaching, etc.

Ivanov and Bobick [67], on the other hand, model interaction sequences as a grammar, performing recognition by parsing the series of observed actions. Hongeng and Nevatia [64] recognize multi-agent activities by integrating finite state machines with temporal relations (before, during, after). Vaswani *et al.* [125] take a geometric approach, analyzing group behavior by considering how the convex hull determined by the group member locations changes over time.

Other techniques look at the problem of activity recognition focusing on the individual objects' trajectories and both spatial and temporal relationships between them. Chan *et al.* [21], for example, jointly consider the recognition of complex events together with linking of trajectory fragments. Recently, Zhou *et al.* [136] used time-varying features extracted from two tracks to recognize activities such as chasing, following, and meeting. Sadilek and Kautz [115] develop a framework for analyzing trajectory relationships using GPS tracks for Capture the Flag.

3.2.3 Sports Analysis, and Event Recognition in Sports Videos

While in the surveillance domain events of interest do not occur often, events of interest occur in the sports domain with much greater frequently due to the continuous interaction between players.

Li *et al.* [85] recently introduced the Discriminative Temporal Interaction Manifold (DTIM) for the recognition of group activities such as predetermined plays in

American football. Intille and Bobick [66] and Swears and Hoogs [119] also consider the domain of American football. Both use Bayesian techniques, the former relying on temporal and logical relationships amongst trajectories, the latter learning play models from track distributions. Gupta *et al.* [60] attempt to learn a storyline model (grammar) for baseball, a technique that benefits from the discrete nature of baseball but may not be readily extended to continuous sports.

Christopher Needham's dissertation [23] addresses tracking and modeling of sports players, learning to recognize player positions in sports such as indoor soccer. Kristan *et al.* [77] describe a closed-world particle filtering approach to track multiple interacting individuals. Perse *et al.* [107] attempt to recognize plays in basketball matching trajectory segments to templates. Zhou *et al.* [135] and Peker [105] analyze basketball and other sports using low-level MPEG features.

There are several works which exploit the audio domain and even subtitle text to develop multi-modal approaches for recognizing events. Such approaches include Liu *et al.* [87] for summarizing racquet sports such as ping-pong and tennis, Nepal *et al.* [97] in basketball, Leonardi *et al.* [83] in soccer, and Rui *et al.* [114] and Gong *et al.* [57] in baseball.

While our work on kinetic accessibility was developed independently, the work of Kang *et al.* [72] has particular relevance to our framework. They applied very similar techniques for the performance evaluation of soccer players, though the theory they proposed lacked many of the details our approach provides.

3.3 Understanding Behavior using Kinetic Accessibility

This work presents an approach based on *kinetic accessibility* (described below) for modeling the behavior of agents performing perceptual tasks in highly interactive environments. Examples of such perceptual tasks include steering an automobile, competitive team sports (see Figure 3.2 for example) and pedestrians walking in urban environments. Modeling how an agent perceives the presence and movement (both spatial and temporal) of other individuals provides insight into the actions and behavior of the agent.

3.3.1 Human Motion Model

Before delving into the details of the kinetic accessibility model, we first consider the underlying representation for human motion. A single model is assumed to be applicable to all players, primarily due to the lack of sufficient training data to calibrate individual per-player models. The model is characterized by the parameters $\mathcal{M} = (\tau_{react}, V_{max}, A_{max})$.

Consider an individual $\mathcal{I}$ that has been tracked up to the present time T_0, and denote $\mathcal{I}$'s trajectory by $\mathbf{X}^{(\mathcal{I})} = \{\ldots, \vec{x}_{-2}^{(i)}, \vec{x}_{-1}^{(i)}, \vec{x}_{0}^{(i)}\}$ where each vector position is used to represent the coordinate location on the ground plane, $\vec{x}_t^{(i)} = \left(px_t^{(i)}, py_t^{(i)}\right)$. To model $\mathcal{I}$'s motion going forward in time $\{\vec{x}_1^{(i)}, \vec{x}_2^{(i)}, \ldots\}$, the velocity and acceleration vectors at time T_0 are given by $\vec{v}_0^{(i)}$ and $\vec{a}_0^{(i)}$. Unless describing relationships between individuals, the superscript $(\mathcal{I})$ is dropped for simplicity of notation, yield-

ing $\vec{x}_t = (x_t, y_t)$, $\vec{v}_0 = (vx_0, vy_0)$, $\vec{a}_0 = (ax_0, ay_0)$, *etc.*

If an individual were to continue with its current velocity and acceleration, then its position as a function of time τ could be predicted simply as

$$\vec{x}_\tau = \vec{x}(\tau) = \vec{x}_0 + \int_0^\tau \vec{v}_0 + \vec{a}_0 t \; dt \tag{3.1}$$

$$= \vec{x}_0 + \vec{v}_0\tau + \frac{1}{2}\vec{a}_0\tau^2 \tag{3.2}$$

To allow the model to handle reactive motion, a change in acceleration is permitted; this is where the model parameters, $\mathcal{M}$, come into play. Before any change in acceleration is allowed, a minimal response time, τ_{react}, must pass. After this delay, the acceleration will "instantly" be fixed to $\vec{a}_{react}$ instead of $\vec{a}_0$. The magnitude of the acceleration and velocity vectors are bounded however, $||\vec{a}_{react}|| \leq A_{max}$ and $||\vec{v}_t|| \leq V_{max}$. This dynamic motion model is specified formally as

$$\vec{x}(\tau) = \begin{cases} \vec{x}_0 + \int_0^\tau \vec{v}_0 + \vec{a}_0 t \; dt & \text{if} \quad 0 \leq \tau \leq \tau_{react} \\ \vec{x}(\tau_{react}) + \int_0^{\tau - \tau_{react}} \vec{v}_0 + \vec{a}_{react} t \; dt & \text{if} \quad \tau_{react} < \tau \leq \tau_{peak} \\ \vec{x}(\tau_{peak}) + \vec{v}_{\tau_{peak}}(\tau - \tau_{peak}) & \text{if} \quad \tau_{peak} \leq \tau \end{cases} \tag{3.3}$$

where the individual's motion fall into one of three phases: (1) a pre-reaction phase, where the initial velocity $\vec{v}_0$ and acceleration $\vec{a}_0$ apply; (2) an acceleration phase, where the velocity is adjusting due to the response acceleration $\vec{a}_{react}$; and (3) a final phase that starts at time τ_{peak}, when the velocity magnitude reaches the maximum allowable $||\vec{v}_0 + \vec{a}_{react}(\tau_{peak} - \tau_{react})|| = V_{max}$.

For simplicity we employ an isotropic acceleration model, meaning that the maximum acceleration is *not* dependent on either the current direction or rate of

Figure 3.2: **Example Basketball Scenario.** The player with the ball determines which player to pass to mostly by the availability of unobstructed passing lanes travel.

3.3.2 The Kinetic Space-time Envelope

Next, we consider an individual positioned at the origin and able to move in any direction within the plane, but constrained to a constant velocity v — no faster, no slower.

Constant Speed, Direction Unknown. The location of this individual after some time t must lie on the locus of points $(vtcos(\theta), vtsin(\theta))$ for some angle θ. If instead of a fixed velocity, we instead bound the velocity to be at most v, then the location at time t must lie on or within that circle. At each time t, the *envelope* of points where the individual could be is an x-y cross-section of a cone in x-y-t space, where the cone's apex is at the origin and its axis points in the $\hat{t}$ direction. Figure 3.3(a)

shows such a scenario with the cross-sectional slices for various times and a single vertical section along the x-t plane.

Initially Stationary, with Acceleration. In reality, an object's movement is governed both by it's current motion and by any forces acting upon it. So, if an object is to move in any direction with equal likelihood, it must initially be at rest. If it is assumed to be at rest and is given the ability to accelerate (due to self-applied force) in any direction with some maximum acceleration, then the space-time "cone" looks like the one depicted in Figure 3.3(b).

Initially Moving, with Acceleration. Suppose now that the initially stationary constraint is removed, and instead, some initial velocity vector is provided. This change results in a "moving cone" like the one depicted in Figure 3.3(c).

Full Motion Model. The only constraints that are missing to complete our construction of the model described in Section 3.3.1 are the reaction time and a maximum velocity. Incorporating these constraints we arrive at the model described, a visualization of which is given in Figure 3.3(d).

Full Model, with Reach. All models thus far have treated individuals as point particles, whereas in reality, players can reach to catch a ball. We incorporate a finite reaching range at time zero and propagate this over time, producing the model depicted in Figure 3.3(e).

3.3.3 Multi-Agent Kinetic Accessibility

The objective is to model interactions between many agents. The human movement model and corresponding space-time envelope just described provide the foundation for an intuitive representation of spatio-temporal interaction which we term *kinetic accessibility*. This representation is designed to consider where each of the various agents could be within a small window of time (perhaps a second or two), and how perception of these relative locations influences agent behavior. Returning to the scenario depicted in Figure 3.2, the player with the ball needs to know which of his teammates are open for a pass.

To determine whether player p_i can pass to a teammate p_j, we assume a maximum speed with which a player might throw the ball (we have currently used 40 ft/s) and consider the conical envelope in space time generated by an object moving with fixed velocity (again see Figure 3.3(a)). This cone represents the potential position of the ball over time if thrown at the present time. We now want to consider where the teammate p_j might be. If we consider the kinetic space-time envelope (see Figure 3.3(e)) of p_j this represents where p_j could be over time. The intersection of the ball's conical surface within the volume of the envelope for p_j indicates the locations in space-time where p_j could receive a pass thrown by p_i at the current time.

What remains is to ensure that the opponents cannot intercept the pass. By intersecting all 9 other players and projecting these intersections radially with respect to the time axis centered at player p_i's current position, we get a kinetic

accessibility map such as that shown in Figure 3.4. The x-axis measures the radial direction from 0 to 2π. Radial "passing lanes" from the apex along the surface of the ball cone correspond to vertical lines descending along the kinetic accessibility map. The y-axis simultaneously measures time and radial distance from where the ball is thrown. Horizontal lines in the map correspond to circles (as in 3.3(a)); lines nearer to the top of the map represent smaller circles closer to the point (and time) of release.

To determine whether a pass from p_i to p_j would potentially be intercepted, we inspect the map for in the angular range where p_j's space-time envelope intersects with the ball cone. Descending vertically from the top of the map (the point where p_i is throwing the ball from), we determine which fraction of the radials in this range are unobstructed by the opponent player's cones. This fraction indicates the possibility of p_j receiving an unobstructed pass.

3.4 Incorporating Local Information

In addition to the kinetic accessibility quantities, we also use local measures such as tracking the ball and detecting dribbling activity. These methods are described below.

To locate the ball, we use a Partial Least Squares (PLS)-based object detector [116], which uses histogram and co-occurrence features that primarily represent the shape of the object. Locating an object within a large image (1920×1080) using a scanning window approach can be quite slow. We speed up the detection

process by adopting a two stage process; in the first stage a Kernel Density Estimation (KDE)-based color model is used to identify potential locations of the ball. In the second stage, only the regions which have a high probability of being the ball according to the color model are considered by the PLS detector. A KDE-based model can be implemented efficiently using lookup tables and as a result this two stage process is about two orders of magnitude faster than running the PLS detector over the entire image. The color model also improves the detection accuracy, as it helps discard regions like the players' heads, which could be confused with the ball based on shape alone. The ball detection is performed independently on each frame. Detections are merged through time by matching each detection to the closest one in the succeeding frame, thus obtaining potential ball tracks. A simple track-filtering procedure helps discard outlying false positives, leaving behind the ball's trajectory. Example results from this process are shown in Figure 3.5.

3.4.1 Ball Possession

To utilize kinetic accessibility for recognition, the player in possession of the ball needs to be known. This is determined using two different measures based on the proximity of the players from the ball.

Image plane distance: Here we compute the distance of each player from the ball (distance between bounding box centers). The image plane distances are not very reliable as there is no notion of depth. Hence, instead of assigning the ball to the

closest player, we perform a soft assignment given by:

$$P_{ball}(p_i) = \frac{\frac{1}{d_i}}{\sum_{i=1}^{10} \frac{1}{d_i}} \tag{3.4}$$

where $P_{ball}(p_i)$ denotes the probability of player p_i having the ball and d_i is the image distance between player p_i and the ball. Such a measure assigns a higher probability to players close to the ball. If no player is within distance d_{thresh} from the ball, then $P_{ball}(p_i)$ is set to $\frac{1}{10}$ for all players.

World coordinate distance: A homography mapping the image plane coordinates onto the court surface enables us to compare distances in world coordinates. During a game, the ball is in contact with the court surface only when it is bounced. Such moments can be easily identified by looking for well defined minima in the y-coordinate values of the ball trajectory. Once a bounce is identified, the world coordinates (in the court reference frame) of the ball at the point of bounce can be computed using the homography. Similarly, the world coordinates of each player are determined from the tracks and the player closest to the ball is assumed to be in possession of the ball.

Generally, the probabilities derived from image-based distances are applied. At those times when a bounce is detected, however, the more accurate world-based estimates are used.

3.5 Probabilistic Reasoning Model

Recognizing sports events using the local and global features described in Sections 3.3 and 3.4 requires a framework that combines these probabilistic cues

Table 3.1: Probabilistic Predicates

Source	Probabilistic Predicate	Description of How Computed
Kinetic	`CanPass`(P_1,P_2,T)	Derived from the kinetic accessibility (as described in Section 3.3.3), this indicates the probability that player P_1 has a clear path available to pass to P_2 at time T.
Kinetic	`ClearShot`(P,T)	Also derived from kinetic accessibility; if at time T, player P can pass the ball in the direction of the basket (without opponent in position to block shot).
Low-level	`HasBall`(P,T)	Produced by the ball detection and tracking framework (described in Section 3.4.1), assigns a probability to each player (these probabilities sum to one).
Low-level	`BallNearBasket`(T)	The probability of being near the basket is determined by image plane distance using the results of the ball detection/tracking framework.
Given	`SameTeam`(P_1,P_2)	Provided manually.
Tracking	*`Trajectories`*	*Currently information such as player location and velocity are not used in the logical inference process.*

together with domain-specific knowledge. Similar to Tran and Davis [123], we use Markov Logic Networks (MLNs) [112] to aggregate probabilistic evidence from these sources, simultaneously incorporating higher-level game knowledge and enforcing logical constraints such as mutual exclusivity.

Markov Logic Networks provide a convenient theoretical framework for combining logical rules and uncertain observations. Each such rule or formula F_i is expressed in first-order logic and is given a real-valued weights w_i. These rules are used to construct a Markov network, each node corresponding to a possible grounded

atom and the network, collectively, representing the joint distribution $P(X)$ across the set X of all grounded atoms. For each logical formula F_i, a clique C_i is formed amongst the nodes for the corresponding set of grounded atoms $x_{\{i\}}$, assigning a potential function $\phi_i\left(x_{\{i\}}\right) = \exp\left(w_i f_i\left(x_{\{i\}}\right)\right)$. Given this network, probabilistic inference may be conducted for a particular assignment x of the ground atoms as

$$P(X = x) = \frac{1}{Z}\prod_i \phi_i\big(x_{\{i\}}\big) = \frac{1}{Z}\exp\left(\sum_i w_i f_i\big(x_{\{i\}}\big)\right) \tag{3.5}$$

where $Z = \sum_{X\in\mathbf{X}} \exp\left(\sum_i w_i f_i\big(x_{\{i\}}\big)\right)$ is a normalizing term. As such networks generally contain cycles, sampling techniques such as MCMC are typically employed to perform inference. In our experiments, we utilize the MaxWalkSat algorithm implemented in Alchemy [76] to perform probabilistic inference for the Basketball sports domain.

3.5.1 Logic Rules for Basketball

We now describe the logical rules that are used to incorporate motion information from the ball and players together with the kinetic accessibility measures. Table 3.1 describes the predicates that are generated (with probabilistic weights) from these two information sources.

Given these predicates which express the information in our uncertain observations, we use rules such as the following to produce the desired interaction labels:

$$\begin{aligned}&\texttt{SameTeam}(P_i, P_j) \ \wedge\ \texttt{HasBall}(P_i, T_i) \ \wedge \\ &\texttt{HasBall}(P_j, T_{i+1}) \ \wedge\ \texttt{CanPass}(P_i, P_j, T_i) \implies \texttt{Pass}(P_i, P_j, T_i)\end{aligned} \tag{3.6}$$

$$\neg \texttt{HasBall}(P_i, T_i) \ \vee \ \neg \texttt{HasBall}(P_j, T_{i+1}) \implies \neg \texttt{Pass}(P_i, P_j, T_i) \tag{3.7}$$

$$\texttt{HasBall}(P_i, T_i) \ \wedge \ \texttt{ClearShot}(P_i, T_i) \ \wedge \ \texttt{BallNearBasket}(T_{i+1}) \implies \texttt{Shot}(P_i, T_i) \tag{3.8}$$

3.6 Experimental Results

Our approach seeks to demonstrate that in perceptual tasks, behavior is not governed exclusively by purpose or objective but is strongly influenced by spatial cues. The experiments presented below are designed to show the utility of the proposed kinetic accessibility constructs in the video analysis of a structured multi-person environment.

As hinted throughout the technical portion of the paper, we use the basketball domain for experimental evaluation. While tracking multiple basketball players from overhead omnidirectional cameras[1] is feasible (particularly when multiple views are available), this is a highly atypical angle and perspective from which to view a basketball game. In analyzing a basketball video taken from a more typical camera angle, the problems of tracking many visually confusable individuals with very frequent occlusions arise, topics which continue to be heavily researched and are

[1] We are aware of only two previously published basketball datasets, the CVBASE [106] and APIDIS [51] datasets. CVBASE provides two minutes of 720×576 resolution data for two synchronized overhead cameras, an insufficient quantity of video for our purposes. The APIDIS project presents a very nice dataset, comprised of 7 loosely-synchronized cameras with 1600×1200 resolution. There is only one minute of data posted publicly on the website. Recently, the authors generously provided us with the full game of video data. As there was insufficient time to process this new data, we hope to present results on it in a future submission.

beyond the scope of this paper. We therefore introduce a new dataset [3], comprised of three 40-minute basketball games recorded at 1920×1080 resolution and 30fps. The players have been manually tracked in a portion of the dataset. We use a 5000-frame manually tracked sequence in our experiments.

3.6.1 OpenGL Implementation

The kinetic accessibility framework was partially inspired by the work of Hoff *et al.* [63] on generating Voronoi diagrams using graphics hardware. Though not explicitly required by the model, graphics hardware greatly simplifies the generation of kinetic accessibility features due to their inherently geometric nature. The alternative would require analytical intersection of these iteratively generated conical surfaces. The implementation used in the paper utilizes a combination of OpenGL color, depth and stencil buffers to render the accessibility features or maps.

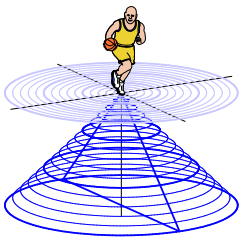
An important part of the framework is the motion model $\mathcal{M}$, whose parameter values were determined empirically. The values for $\mathcal{M}$ were extrapolated from real data (see Figure 3.6) and set to $V_{max} \approx 20ft/s$, and $A_{max} \approx 20ft/s^2$ [2]. The ball's peak velocity was estimated to be $\approx 40ft/s$. As currently implemented, the reaction time τ_{react} is not considered.

[2]The value used for A_{max} is higher than that observed, but endows the model with a greater ability to handle stronger responses than those observed.

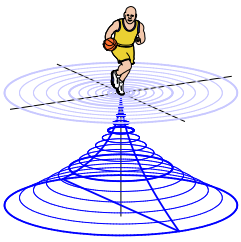
3.6.2 Performance Evaluation

To evaluate the performance of the our approach, we evaluate how accurately various events (`MaintainPossession`, `Pass`, `Shot`, `Turnover`) are estimated on a 5000-frame manually ground-truthed video. For computational feasibility we subdivide the video into 1-second intervals and predict the label for each 1-second interval. Note that the handful of intervals where the ball is out of play are not considered. The results are shown visually in Figure 3.7.

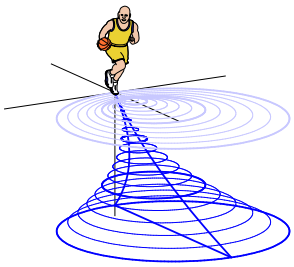
The model predicts the player with possession of the ball with 63.0% precision and 72.8% recall. Accurate detection of the ball serves to anchor the inference of the `Pass` events. While the ball is easily detected if a player has possession for a long period of time (especially when dribbling), it's difficult when the player receives it and then quickly passes again. As a result, when a brief possession is not detected, both the preceding and subsequent passes not successfully predicted. As can be observed in the figure, 4 (almost 5, see player 7 near from 3500) of the 8 shots are accurately detected by the system. While one of the turnovers (near frame 1100) can be inferred, the other is missed because the brief possession by player 8 just before frame 3000 was not detected. Overall, these results are encouraging and indicate that the kinetic accessibility measures encode valuable information about the behavior (events) of the players.



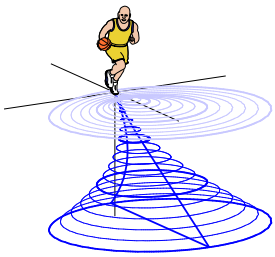
(a) Constant Speed, Direction Unknown



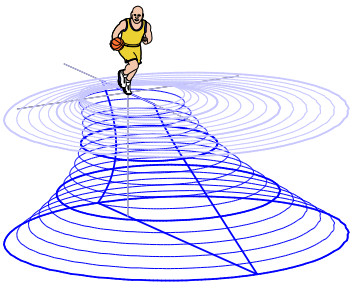
(b) Initially Stationary, with Acceleration



(c) Initially Moving, with Acceleration



(d) Full Motion Model



(e) Full Model, with Reach

Figure 3.3: **The Kinetic Space-time Envelope.** These illustrations depict the construction of the human motion space-time envelope with increasing levels of complexity. Each shows the plane in which the agent moves, with a time-evolving envelope of potential occupancy shown below (and projected up onto) the plane. In the case that the individual is (a) moving with fixed speed in an unknown direction, this time-evolving envelope sweeps out a regular cone. If initially stationary and allowed to accelerate at some maximal acceleration A_{max}, it sweeps out a tapered cone (b). (c) shows an individual initially in motion, but allowed to accelerate. (d) shows the full motion model, which extends (c) by constraining the magnitude of the velocity to V_{max}. While the earlier models treat the individual as a point particle, (e) enhances the full motion model of (d), account for an individual's ability to reach an arm's length in any direction

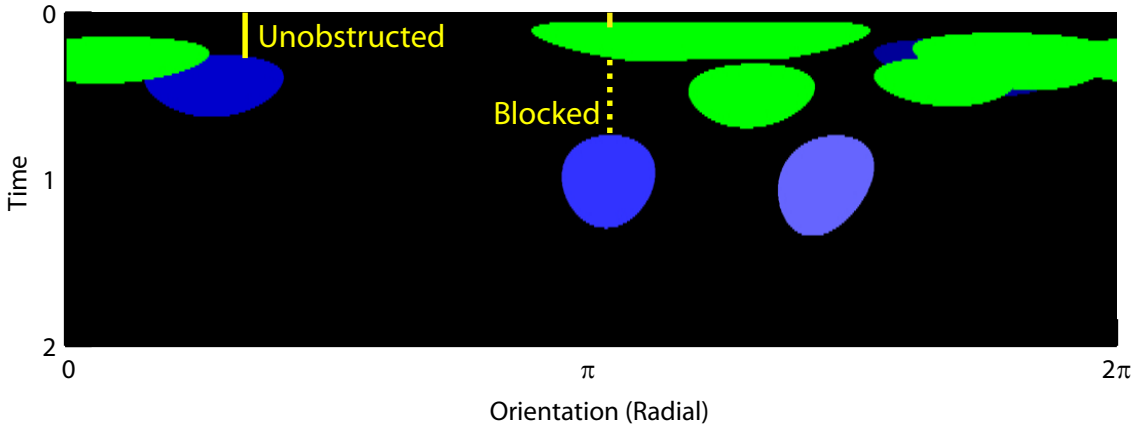


Figure 3.4: **Sample Kinetic Accessibility Map.** Think of this as wrapped around a conical surface where the entire top edge of the image contracts to the apex, the location of the player at the time in question (x-axis runs left to right from angle 0 to π, y-axis runs top to bottom from time 0 to 1 second). Teammates are shown in different shades of blue, opponents are all in green. In this particular map, two players are entirely blocked by opponents, the remaining two are partially occluded

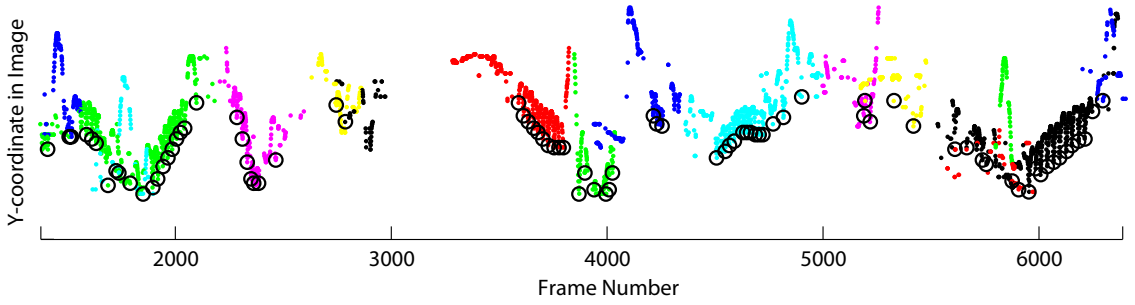
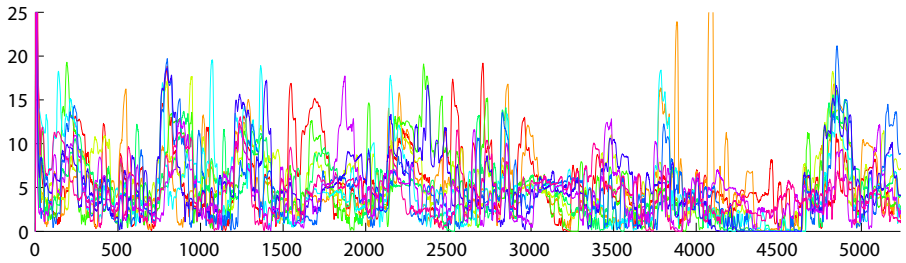
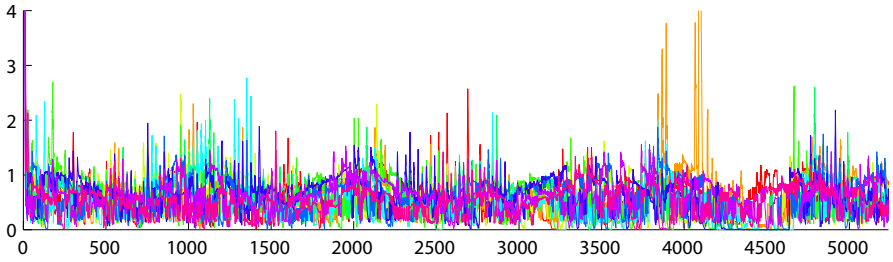


Figure 3.5: **Ball Detection/Tracking Results.** The ball is tracked using the detection approach described for more than 5000 frames. Detections within a contiguous track are in the same color, and detected bounces are indicated with black circles



(a) Velocity (ft/s)



(b) Acceleration (ft/s^2)

Figure 3.6: **Tracking Data used to Determine Parameters Limits.** Tracking data used to determine the parameter values for the maximum (a) velocity V_{max} and (b) maximum acceleration A_{max}

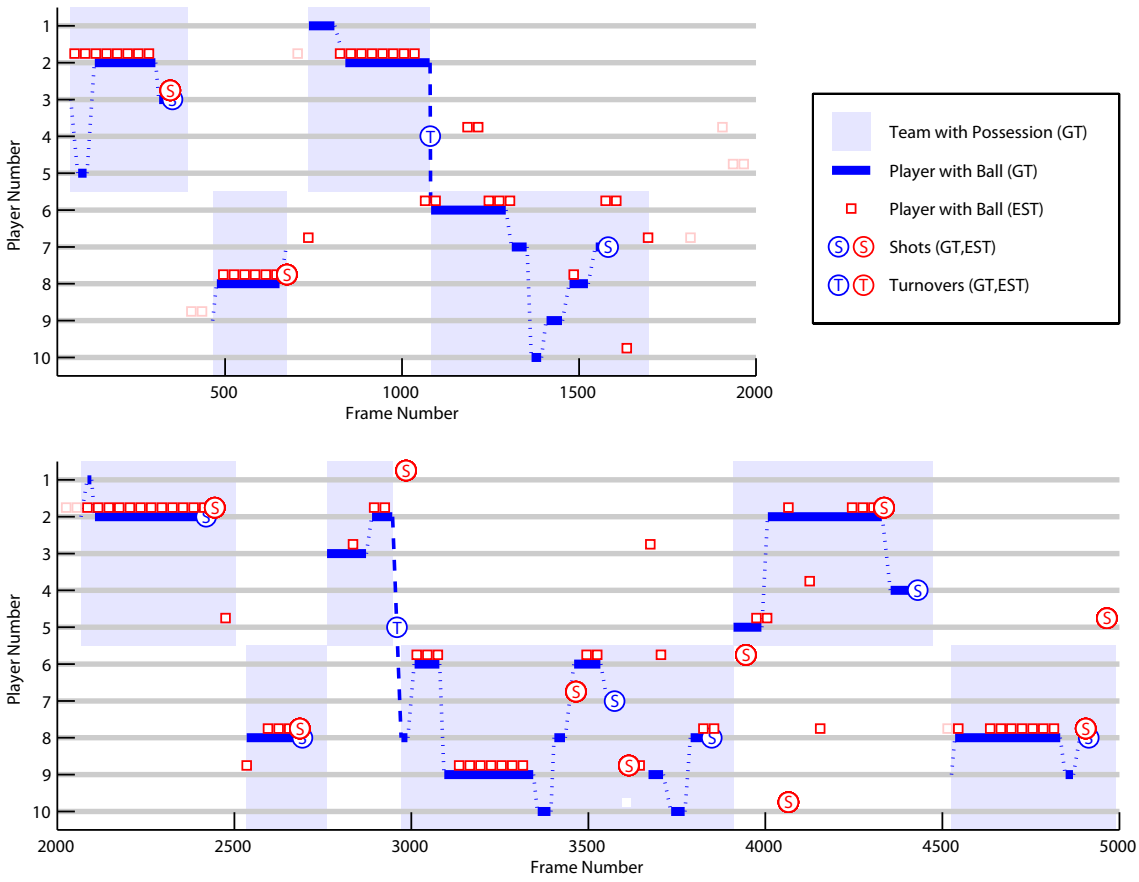


Figure 3.7: **Event Prediction Results.** Ground Truth (in BLUE) and Estimated (in RED) Events for 5000 frames of a Basketball Video

Chapter 4

Learning Higher-order Transition Models in Medium-scale Camera Networks

4.1 Introduction

While traditional CCTV surveillance systems are generally limited to archival and operator monitoring, the recent proliferation of Network Cameras and Smart Cameras [14] heralds a new generation of intelligent surveillance architectures. Future surveillance devices will be endowed with substantial computational and communication resources. The challenge is to provide them with commensurate algorithms to collectively interpret activity within the network.

A fundamental challenge in surveillance is tracking objects and individuals throughout the network in spite of occlusion and lapses in observation, changing illumination, *etc*. Stauffer and Tieu [118] provide an excellent description of the general tracking problem, suggesting that an ideal tracking system should produce "only as many tracking sequences as there were independently moving objects in an environment, regardless of the number of cameras or their overlap". Successful tracking requires the maintenance of object identity, typically relying both on an understanding of the camera network topology and the ability to match properties such as appearance and dynamics across observations [131].

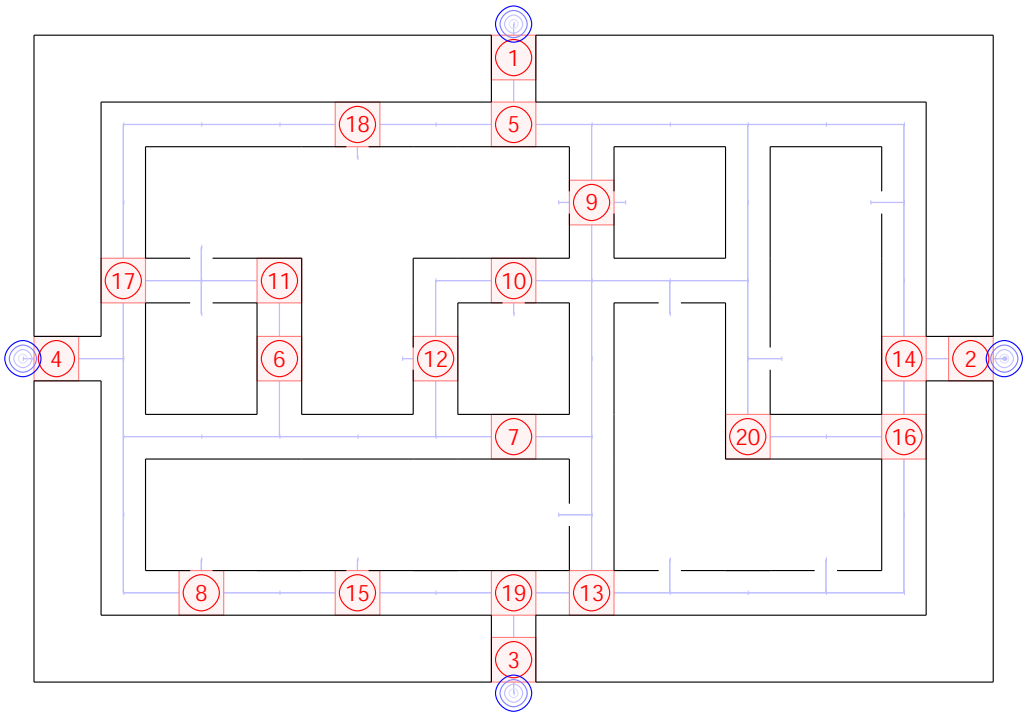


Figure 4.1: **Simulated Camera Network.** One of several camera networks used in our experiments. This one contains 20 cameras: four peripheral cameras (numbered 1-4), with the others (5-20) placed randomly within the halls of the office environment.

We focus on the problem of recovering topology in uncalibrated medium-scale (10-1000 camera) surveillance networks. While previous work has focused mainly on first-order relationships (*i.e.* adjacency), our focus is on higher-order topological relationships and transition models.

Consider the case of a newsstand located in an airport terminal. We might generally expect most of the newspapers and magazines sold to be purchased by departing passengers to read on their flight or while they wait for boarding. Suppose we know, however, that the few people who come from a gate and stop by the

newsstand almost always continue in the direction of the main terminal. Contrary to the general expectation that people go from the newsstand to a gate, we have a strong prior belief that people tracked from a gate to the newsstand will continue toward the main terminal when they leave.

The example illustrates that learning higher-order topological relationships can potentially improve tracking performance. Other benefits of recovering higher-order relationships include resilience to camera/node failure. If second-order topological data is available, then we can overcome loss of any given camera, or even multiple non-adjacent cameras.

4.2 Related Work

In the computer vision literature, previous work on uncalibrated camera network topologies has focused primarily on pairwise camera relationships. Stauffer and Tieu [118] illustrate the possible types of camera overlap: (*i*) non-overlapping views (no mutually observable volume); (*ii*) partially overlapping views (views are connected, adjacent cameras have regions of overlap); (*iii*) completely overlapping views (existence of a spatial volume mutually observable by all cameras); and (*iv*) the general case combining the other types. The problem they tackle is that of modelling regions of overlap for groups of cameras with at least partial overlap (type (*ii*)), where the overlapping regions lie on or near a ground plane. The solution they propose is to consider cameras pairwise and use temporally co-occurring observations sequences in the two views to estimate a homography between them and the

region of overlap or mutual observability.

Recently there has been an increased interest in non-overlapping camera networks. Makris *et al.* [88] attempted to recover the network topology to facilitate tracking between spatially adjacent cameras by estimating the transition delay between two cameras using cross-correlation on large numbers (thousands) of departure/arrival observations. Tieu *et al.* [121] suggested the use of mutual information as a measure of pairwise statistical dependence, using Markov Chain Monte-Carlo (MCMC) to simultaneously recover the correspondence between departures/arrivals and the transition delay distribution. In contrast to these entirely unsupervised techniques, Javed *et al.* [68] use labeled ground-truth trajectories to generate a non-parametric model (using Parzen windows) of transition probability between cameras. The parameters they use in building the model are exit location, entrance location, exit velocity and time delay. After generating this model, they employ an offline tracking algorithm based on bipartite graph matching and an online approach which updates the KDE in real-time.

A vast literature addresses the problem of data association. Among the earliest work in this field was Reid's algorithm [111], a Bayesian formulation for multiple target tracking in a single view (*e.g.* associating radar tracks). Where Reid outlined a multiple-hypothesis tracker (MHT) to deal with the intractable space, Cox and Hingorani [26] later provided an efficient implementation of Reid's algorithm based on bipartite graph matching. Huang and Russell [65] applied this approach to the problem of highway traffic monitoring, offering an improved matching algorithm which scores the quality of each association.

Recent research on multiple-camera surveillance continues to use the Bayesian formulation. Given certain constraints, Kettnaker and Zabih [75] are able to frame the multi-camera problem as a Linear Program. Pasula *et al.* [104] and Oh *et al.* [100] use MCMC-based approaches. Zajdel *et al.* [132, 133, 134] employ an approach similar to Pasula *et al.*, but use Dynamic Bayes Networks to evaluate hypothesis likelihoods and an EM-based algorithm for learning model parameters.

Another body of relevant work is found in the sensor networks literature. The ad hoc and inherently distributed nature of wireless sensor networks has led researchers to a focus on distributed inference techniques. Funiak *et al.* [52] presented an online approach for localizing a network of cameras, essentially employing distributed probabilistic inference to approximately calibrate a network of sensors based on observations of an object moving through the network. Distributed inference is also employed for multi-target tracking. Examples include Chen *et al.* [82] and Chu *et al.* [24] though both apply distributed inference to tracking in dense networks of calibrated, non-visual sensors.

4.3 Learning Camera Network Topology

Our objective is to learn as much as possible about the camera network, while assuming as little as possible. The primary purpose for learning network topology is to improve the models used by tracking algorithms. Understanding the probabilities of potential events that could follow an object's departure from a given camera provides information which should be helpful in tracking the object.

Particularly difficult scenarios arise when multiple objects with similar appearances are simultaneously present in the network. A network topology model can help discriminate between ambiguous objects when appearance alone cannot. For example, suppose we know that a particularly object x cannot get from camera 1, where it was last seen, to camera 3 without passing through camera 2. If an object with the appearance of x is then seen at camera 3 before one is seen at camera 2, we deduce that the object in camera 3 cannot be object x.

To recover the topological relationships, we focus not just on first-order "adjacency". While most previous work only considered where an object leaving camera i could appear next, we are interested in higher-order transition models which provide a richer description of object movement tendencies. As we recover complete trajectories, the range of queries that can be addressed are broader, *e.g.* "What fraction of objects passing through cameras 5 and 7 will, at some time later reach camera 4?".

We make only one assumption about the spatial distribution of the cameras, requiring non-overlapping fields of view. The only information we require a priori is a labelled set of *peripheral cameras*, the subset of cameras where objects may enter or exit the network. We also assume the network is initially empty. Without these two constraints, we would need to consider the possibility that each observation is of a unique, previously unseen object.

Our task is then to recover these underlying trajectories, using what little information we know. Given the peripheral-labeled cameras and the full set of observations, we aim to simultaneously determine how many objects have passed

through, learn their respective appearances and associate which observations belong to which objects. The probabilistic approach we use to partition the observations into individual object trajectories is described next, in Section 4.4.

4.4 Bayesian Observation Partitioning

Several Bayesian approaches for problems such as data association and tracking are described in Section 5.2. Our solution closely follows the Bayesian framework presented in Zajdel [132] for multi-camera tracking (similar approaches used in [100, 104]). This approach learns model parameters incrementally by accumulating observations into consideration incrementally and probabilistically evaluating proposed partitionings of these observations into objects.

4.4.1 Finding the Optimal Partition

In this approach, we first consider $\mathbf{O} = \{o_1, o_2, \ldots, o_N\}$, the entire set of observations. These observations represent the observable portions of the trajectories of K (value unknown) objects moving within the network. Each observation represents an object passing through a given camera at a particular time. The observations could have been generated by a single object ($K = 1$), N distinct objects with just one observation each ($K = N$), or, some number of objects in between ($1 < K < N$). Our goal is to select a partition $\omega \in \Omega_N$ of the observations $\mathbf{O}$

$$\mathbf{O}_\omega \stackrel{\omega}{=} \mathbf{O}_1 \cup \mathbf{O}_2 \cup \ldots \cup \mathbf{O}_{K_\omega} \tag{4.1}$$

such that each set $\mathbf{O}_k = \{o_1^{(k)}, o_2^{(k)}, \ldots, o_{n_k}^{(k)}\}$ contains all n_k observations of the k^{th} object, the temporal sequence $o_1^{(k)}, o_2^{(k)}, \ldots, o_{n_k}^{(k)}$ describing object k's trajectory or path through the network.

Since the true number of objects is unknown, we consider various partitionings (see Figure 4.2) and in estimating the most likely one, hopefully recover the correct set of objects with their respective trajectories. Formally, we consider the space Ω_N of all partitions of the N observations, evaluating each partition's likelihood in the context of established priors and the evidence (observations) collected. However, for any nontrivial observation size N, considering all such partitionings exhaustively is intractable [1]. We therefore use a procedure reminiscent of Reid's multiple hypothesis tracking approach [111], to prune the partition space.

We begin with a small initial observation set consisting of the first m observations, $\mathbf{O}_0 = \{o_1, o_2, \ldots, o_m\}$. We exhaustively enumerate all partitions in Ω_m and evaluate the likelihood of each partition ω using the inference method described below in Section 4.4.2. At this point we discard unlikely partitions, retaining only the B best (most probable) partitions, associating with each retained partition an updated model reflecting the properties of its respective trajectories. Formally, we denote this initial set of hypotheses as $\mathbf{H}_0 = \{h_1^{(0)}, h_2^{(0)}, \ldots, h_B^{(0)}\}$ where $h_i^{(0)}$ is comprised of its partition $\omega_i^{(0)}$ and its resulting transition model $T_i^{(0)}$ (the transition model is covered more fully in section 4.4.2.2).

[1] The number of ways to partition a set of n elements is given by the n^{th} Bell number, B_n. The first 10 Bell numbers are 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975 and, in general, $B_{n+1} = 1 + \sum_{k=1}^{n} \binom{n}{k} B_k$.

With our initial set of hypotheses $\mathbf{H}_0$ formed, we begin an incremental search process akin to Fox's beam search [50]. At each iteration we add a few, s, additional observations and again consider the resulting partitions and prune all but the best. For the τ^{th} iteration, to extend each hypothesis $h_i^{(\tau-1)} \in \mathbf{H}_{\tau-1}$ with s additional observations, we must evaluate $O(k^s)$ amended partitions[2], where k is the number of trajectories in $h_i^{(\tau-1)}$. Due to the exponential complexity $O(B \cdot k^s)$, small values of s are used in practice. After these amended partitions are evaluated, the unlikely partitions are again pruned and we form $\mathbf{H}_\tau$ by retaining the B most likely amended partitions, each with its updated model. This incremental process is continued until all of the observations have been brought into consideration and the most likely partition in the final hypothesis set is taken as the final MAP estimate (this is described in greater detail below).

4.4.2 Partition Likelihood

To determine which partitioning of the observations is the most likely, we wish to find the partitioning $\omega_{MAP} \in \Omega_N$ which maximizes the posterior $P(\omega|\mathbf{O})$. Assuming a uniform prior $P(\omega)$, we use Bayes' rule to express this posterior in terms

[2] For $s = \{1, 2, 3, \ldots\}$, adding s additional observations to a hypothesis of size k will produce $\{k+1, k^2+2k+2, k^3+3k^2+6k+5, \ldots\}$ amended partitions to evaluate. In essence, each observation added can go into any one of the existing trajectories or be considered as a new object. The combinatorial complexity of adding s observations to a partition with k trajectories is $O(k^s)$, independent of the total number of observations.

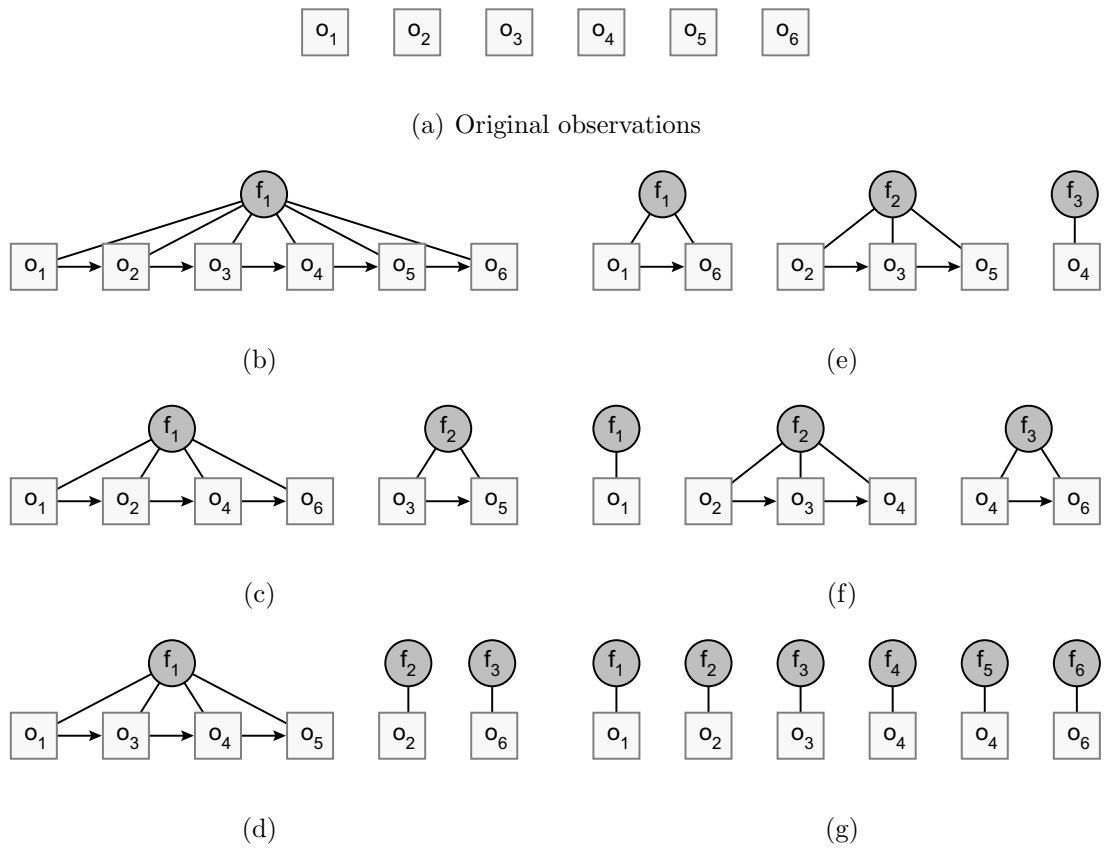


Figure 4.2: **The Space of Partitions**. In (b)-(g) we show a few of the 203 possible ways to partition the six observations shown in (a). Semantically, (b) refers to the hypothesis that a single object generated all six observations. Similarly, (g) depicts the scenario where each observations was generated by a unique object. The difference between (e) and (f) is the object to which observation o_6 is attributed. Note that in any given trajectory the observations must form a temporally-increasing sequence.

of the likelihood

$$P(\omega|\mathbf{O}) = \alpha P(\mathbf{O}|\omega)P(\omega) = \alpha P(\mathbf{O}|\omega). \tag{4.2}$$

where α represents normalization terms.

Recall that $\mathbf{O}_\omega$, defined in Eq (4.1), represents the division by ω of the complete

set of observations, $\mathbf{O}$, into K_w disjoint trajectories, $\mathbf{O}_1, \mathbf{O}_2, \ldots \mathbf{O}_K$. Assuming independence amongst the object trajectories, the likelihood $P(\mathbf{O}|\omega) = P(\mathbf{O}_\omega)$ can be factored as a product of the individual trajectory likelihoods

$$P(\mathbf{O}_\omega) = \prod_{k=1}^{K_\omega} P(\mathbf{O}_k) \tag{4.3}$$

The likelihood of a given trajectory is dependent on various parameters including the object's intrinsic appearance and the camera topology/transition model. As in Zajdel [132], we use a Dynamic Bayes Net (DBN) to evaluate the likelihood of each given trajectory.

We first describe the graphical model representing a single trajectory, illustrated in Figure 4.3. The intrinsic appearance of object k is described by the hidden variable f_k. Each observation o_i in the trajectory's observation set $\mathbf{O}_k = \{o_1, o_2, \ldots, o_{n_k}\}$ is represented by the observable variables a_i, c_i, t_i, e_i, and d_i, described on the left of Figure 4.3. This graphical model facilitates representing the joint distribution over all variables, by describing conditional dependencies (arrows) between them. The conditional dependencies, represented by PDFs, are described below together with priors for those variables that are not conditioned on others:

- $P(f_k)$ - the prior probability on the intrinsic appearance of an object. See Section 4.4.2.1 for details.

- $P(a_i|f_k, c_i)$ - the appearance model. The observed appearance depends on both the intrinsic object properties and camera-specific factors such as illumination and occlusion.

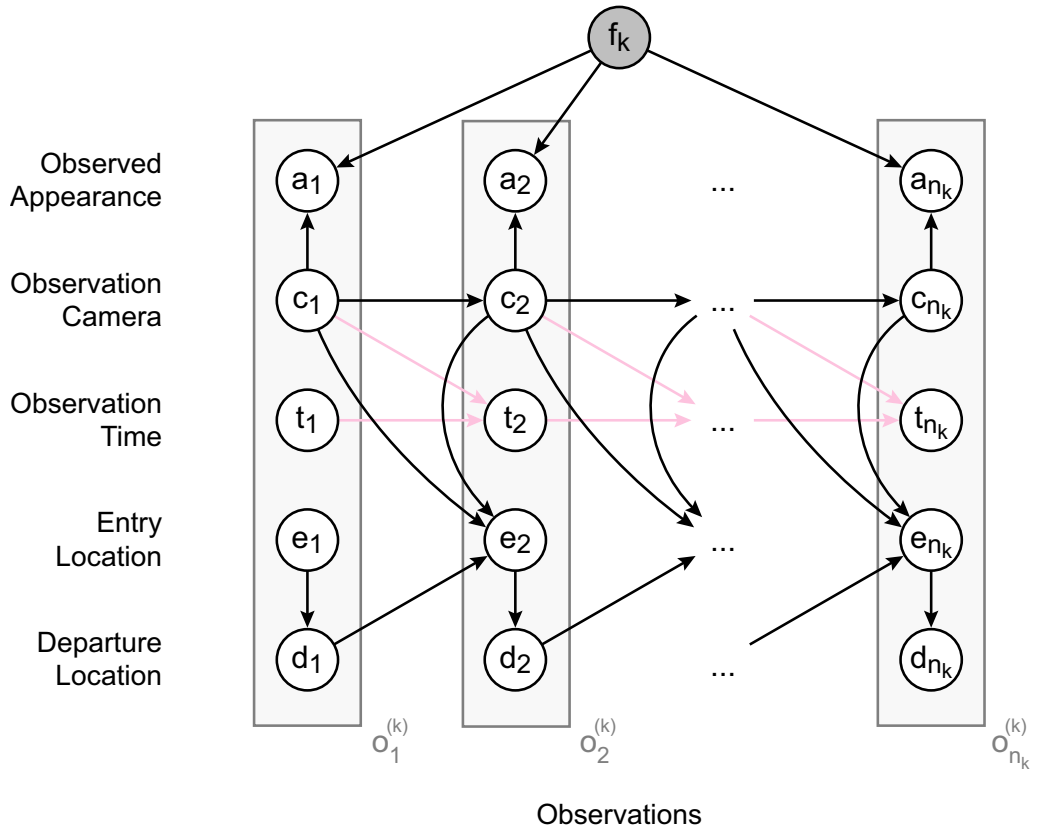


Figure 4.3: **Graphical Model (Dynamic Bayes Net).** This is the model used for computing trajectory likelihood and estimating object k's intrinsic appearance f_k.

- $P(e_i, c_i | d_{i-1}, c_{i-1})$ - the [first-order] transition model/topology. In practice, we approximate this distribution by the product $P(e_i|c_i, d_{i-1}, c_{i-1}) \cdot P(c_i|c_{i-1})$.

- $P(t_i|c_i, c_{i-1}, t_{i-1})$ - the distribution of transition times between cameras. While of great utility when object dynamics are highly predicable (see [88, 121]), in other applications where objects stop or disappear for long, uncorrelated periods of time, this term may be of lesser value. This term is presently neglected but is mentioned here for completeness.

- $P(d_i|c_i, e_i)$ - the typical paths within a camera's field of view. As this information is clearly observed within each camera, this term is computed directly from the data.

- $P(c_1)$ - the cameras where an object may enter (or exit) the network. The *peripheral cameras* are incorporated into the model in this manner.

- $P(e_1|c_1)$ - the entry points in the peripheral cameras where an object can first appear; this is learned from the data.

Using these priors and conditional distributions, the DBN allows computation of the trajectory likelihood as

$$P(\mathbf{O}_k) = P(c_1)\,P(e_1|c_1) \cdot \overbrace{\left(\prod_{i=2}^{n_K} P(e_i, c_i|d_{i-1}, c_{i-1})\right)}^{\text{Inter-Camera}} \cdot \underbrace{\left(\prod_{i=1}^{n_K} P(d_i|c_i, e_i)\right)}_{\text{Intra-Camera}} \cdot \underbrace{\left(\prod_{i=1}^{n_K} P(a_i|\hat{f}_k, c_i)\right)}_{\text{Appearance}} \tag{4.4}$$

where $\hat{f}_k$ is the estimated intrinsic appearance for object k.

4.4.2.1 Estimating Intrinsic Appearance

A given partitioning ω splits the set of observations $\mathbf{O}$ into K_ω individual trajectories $\{\mathbf{O}_k\}$. In Eq (4.4), the appearance term $P(a_i|\hat{f}_k, c_i)$, expresses the measurement likelihood that camera c_i measures the appearance a_i (color, etc.) from object k where object k's actual appearance is given by (estimated as) $\hat{f}_k$. At present, estimation of the camera-specific influence on measured appearance is not considered.

To facilitate parameter estimation, we model the intrinsic appearance as a Gaussian distribution, $\hat{f}_k = \mathcal{N}(\mu_k, \Sigma_k)$, though more descriptive models could be employed. The appearance of each observation is represented by a point in RGB color-space. We compute μ_k as the maximum likelihood estimate μ_{ML}, equal to the sample mean. We assume a known covariance Σ_k derived from the complete observation set.

4.4.2.2 Transition Model Parameters

The transition model consists of a known prior $P(c_1)$ and the conditional dependencies $P(e_1|c_1)$, $P(e_i, c_i|d_{i-1}, c_{i-1})$, and $P(d_i|c_i, e_i)$ which are learned. As we learn the transition model incrementally, starting with just a few observations (see section 4.4.1), we want to dynamically model the uncertainty, which begins high but gradually decreases as we consider additional observations. To model this uncertainty, we represent the transition model by combining a uniform prior T_{unif} with the model constructed from the current partitioning of the observations T_{data}

$$T(\tau) = \beta \cdot T_{unif} + (1 - \beta) \cdot T_{data} \tag{4.5}$$

where β is the exponentially decaying interpolation parameter defined as $\beta = e^{-4\frac{m+s\tau}{N}}$ and, as previously, τ represents the iteration number ($0 \leq \tau \leq \lceil \frac{N-m}{s} \rceil$). In a given iteration, the transition model used for the inter- and intra-camera conditional probabilities is $T^{(\tau-1)}$, the model resulting from the previous iteration. The updated model $T^{(\tau)}$ is computed after completing iteration τ, only on the partition hypotheses which are retained.

4.5 Experimental Results

We created randomly-generated medium-scale camera networks comprised of 20 cameras placed in the hallways of an indoor office environment (see example in Figure 4.1). The simulator, implemented in MATLAB, can control the number of objects (people) in the network as well as their behavior: whether they stay primarily in their own office, visit colleagues, how quickly they leave, etc. Each time an object passes through a camera's field of view an observation is recorded, noting the time and image location of the object's entrance and exit, and the measured appearance for the object. Ground truth appearance values are perturbed for each observation by additive Gaussian noise with parameters $\mathcal{N}(0.5, \sigma_a)$ in each color-space (RGB) dimension.

All observations made within the network are gathered into a single observation set $\mathbf{O}$, sorted by entrance time. We then follow the incremental estimation procedure outlined in section 4.4.1. After beginning with a small initial set of the first m observations, we iteratively add s observations, evaluating and keeping only the B best partitions at each iteration. The iteration continues until the entire set $\mathbf{O}$ has been considered yielding a final maximum a posteriori partition estimate ω_{MAP} and the corresponding transition model T_{MAP}.

4.5.1 Trajectory Reconstruction

It is critical to accurately reconstruct the original object trajectories. As we will show in sections 4.5.2 and 4.5.3, accurate reconstruction of the trajectories

ensures accurate estimation of both first- and higher-order topological relationships.

To quantitatively assess trajectory reconstruction, we use two measures: partition *accuracy* and partition *recall* (see [132]). Suppose the true (ground-truth) partition $\bar{\omega}$ divides the full observation set $\mathbf{O}$ into $K_{\bar{\omega}}$ trajectories $\bar{\mathbf{O}}_i$, $1 \leq i \leq K_{\bar{\omega}}$. Similarly, the partition estimated by our algorithm $\hat{\omega}$ produces $K_{\hat{\omega}}$ trajectories $\hat{\mathbf{O}}_k$, $1 \leq k \leq K_{\hat{\omega}}$. The partition *accuracy* denotes the [average] fraction of each recovered trajectory's observations that actually belong to some ground-truth trajectory

$$q_{\hat{\omega}} = \frac{1}{K_{\hat{\omega}}} \sum_{k=1}^{K_{\hat{\omega}}} \frac{\max_i |\bar{\mathbf{O}}_i \cap \hat{\mathbf{O}}_k|}{|\hat{\mathbf{O}}_k|} \cdot 100\% \tag{4.6}$$

Similarly, the partition *recall* indicates the fraction of each ground-truth trajectory's observations that are partitioned together in the estimated partition

$$\rho_{\hat{\omega}} = \frac{1}{K_{\bar{\omega}}} \sum_{i=1}^{K_{\bar{\omega}}} \frac{\max_k |\bar{\mathbf{O}}_i \cap \hat{\mathbf{O}}_k|}{|\bar{\mathbf{O}}_i|} \cdot 100\% \tag{4.7}$$

After using our algorithm to recover object trajectories in several simulated camera networks, we apply these two metrics to the results. Table 4.1 shows how performance varies with changes in B, the number of hypotheses retained at each iteration and with σ_a, the appearance noise parameter (see Figure 4.4 for a visual noise comparison). These results represent average performance across 20 randomly-generated camera networks. Each 20-camera network accumulated observations from ten objects moving through the network with a mean of 32.8 observations collected per object (per network).

B	$\sigma_a = 0.00$ acc./recall	$\sigma_a = 0.02$ acc./recall	$\sigma_a = 0.05$ acc./recall	$\sigma_a = 0.10$ acc./recall	$\sigma_a = 0.20$ acc./recall
1	70.0 / 98.4	68.2 / 97.6	63.9 / 94.4	52.7 / 83.0	40.3 / 59.6
2	69.4 / 97.9	69.0 / 97.0	65.5 / 94.1	54.1 / 82.9	39.0 / 60.0
5	69.5 / 98.5	67.7 / 96.8	65.5 / 94.2	54.4 / 84.0	38.8 / 60.3
10	69.7 / 97.5	69.2 / 97.0	65.9 / 94.1	54.2 / 84.6	40.9 / 59.5
25	- / -	69.7 / 97.2	- / -	55.4 / 84.2	- / -

Table 4.1: **Performance across Hypothesis and Appearance Noise Parameters.** The partition accuracy and recall vary as the number of retained hypotheses B and the appearance noise parameter, σ_a are changed. In these simulations $m = 8$ and $s = 2$ are fixed. The influence of appearance noise on accuracy and recall is substantial, while that of the hypotheses retained is negligible. We believe that accuracy values are lower than recall due to the recovery of too few trajectories. When a new object first appears, if all partitions which attribute it to a new trajectory are pruned then all of its observations will be assigned to existing trajectories.

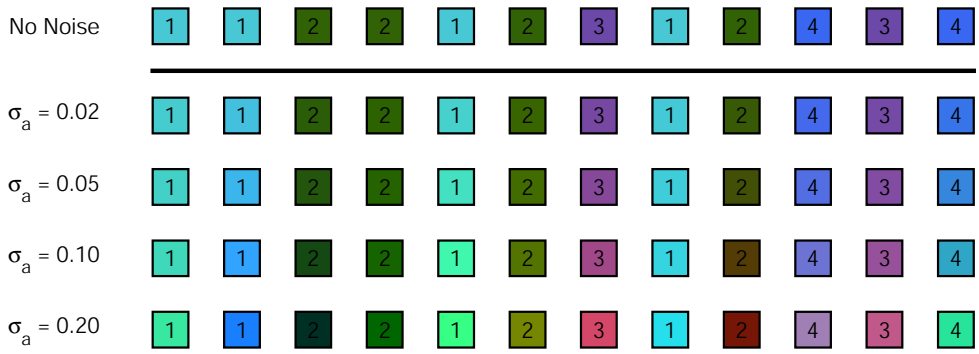


Figure 4.4: **Visual Noise Comparison.** Twelve initial observations labeled by object identity (the object number is displayed inside each observation's square). The top row shows the true appearance (color) of each observation as measured without any noise. The four lower rows show how noise of varying levels ($\sigma_a = \{0.02, 0.05, 0.10, 0.20\}$) can change the measured appearance. Note how some objects (*e.g.* the first and last of the twelve observations) can begin to appear similar when measured with high noise, making appearance a less effective discriminant between objects.

4.5.2 First-Order Topology

We next evaluate the algorithm's recovery of the first-order topology, as done in previous work on topology [88, 121]. Our comparison is based on a stochastic adjacency matrix we call the *topology matrix*. The entries of row i form a probability distribution, indicating the probability that an object last seen at camera i will next appear at a particular camera. In theory, the binary matrix formed by replacing the non-zero transition probabilities in the topology matrix with ones would be symmetric (if an object can move from camera a to b it should be able to return

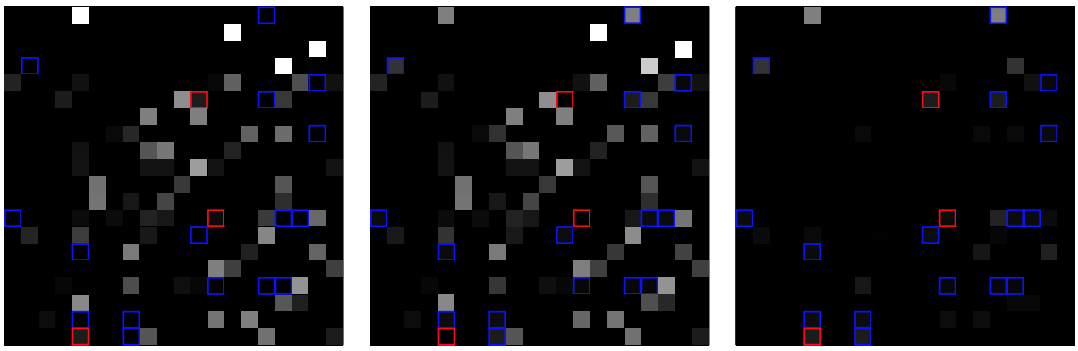
from b to a). However, while some domains might exhibit true "one-way" paths, in practice there simply may not be any objects taking the reverse path despite its availability.

Results from a simulated 20-camera network (that shown in Figure 4.1) are presented in Figure 4.5. This simulation has ten[3] objects collectively producing 314 observations. Both the ground-truth and recovered topology matrices are shown, together with an error matrix displaying discrepancies between the two. The results show the recovery of almost every camera-camera transition made, and with the correct probabilities in all but a few cases. With the exception of cameras 1 and 4 (the top and fourth rows), all of the spurious estimated transitions are of negligible probability.

4.5.3 Higher-Order Topology

Partitioning the observations into full object trajectories enables the extraction of higher-order topological relationships, simply by analyzing the estimated trajectories. With this additional information, we can answer queries such as, "If an object was first observed in camera a and next in camera b, what is the likelihood that it will next be seen in camera c?". As we are unaware of other work recovering higher-order transition models, we cannot provide a direct comparison with other algorithms. We therefore present results showing the extent to which our technique is able to accurately recover the second-order transition model. Example second-

[3]For real-world observations, one would, of course, need far more than ten tracks to construct a useful model.

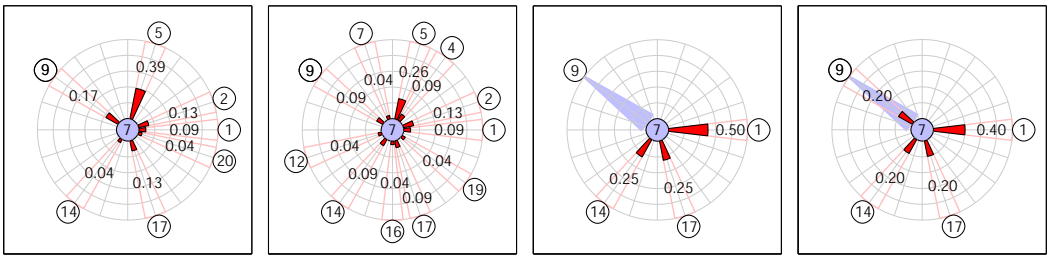


(a) Ground Truth (b) Estimated (c) Errors

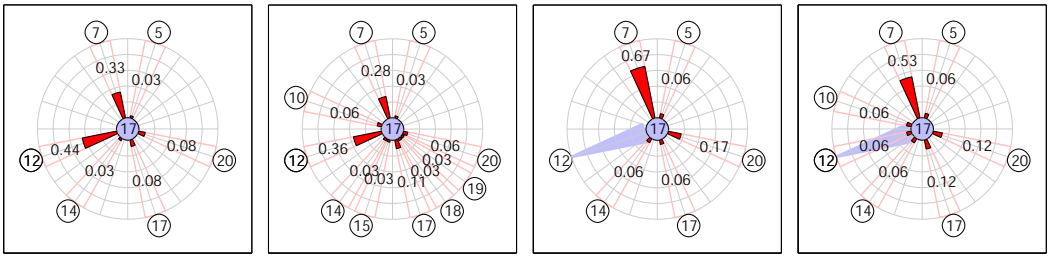
Figure 4.5: **First-order Topology.** Shown here are the topology matrices induced by (a) the ground truth partition, (b) the estimated best partition, and in (c) the error between the ground truth and estimated topology matrices. The observations used had an appearance noise of $\sigma_a = 0.05$, and the parameters used in estimation were $m = 8$,$s = 2$, and $B = 10$. Entries framed in red denote non-zero entries in the ground-truth which were entirely lost in the recovery process. Blue-framed entries denote spurious transitions due to estimation errors. Both are generally very low-probability errors.

order transition model estimation results are shown in Figure 4.6. The increased expressiveness of the second-order model over first-order adjacency can be seen.

Average second-order transition model errors are shown in Figure 4.7. Errors are computed using sum of squared errors between the ground-truth and the estimated distributions. The probability of an object passing through cameras a,b, then c in sequence is expressed as a distribution across camera c for the given camera pair (a, b). The presented results are the average errors over all pairs (a, b).



(a) First-order (GT,EST) (b) Second-order (GT,EST)



(c) First-order (GT,EST) (d) Second-order (GT,EST)

Figure 4.6: **Higher-order Transition Model Examples.** Two cameras, serve to illustrate the expressiveness of the second-order transition model. Each pair (a)-(d) shows the ground-truth (GT) model on the left and the estimated (EST) model on the right. In these plots, the light blue circle in the center is the camera where the object was last observed (with a blue path indicating where it came from in the second-order model). The red paths indicate probabilities of next appearing at other given cameras, with each visible radial bar proportional in length to its in respective non-zero probability. In (a) and (b) we see that while in general objects at camera 7 most often go to camera 5, the second-order model shows that objects arriving at camera 7 never go to camera 5, rather to cameras 1, 14, and 17. Similarly in (c) and (d), objects at camera 17 are more likely to go to camera 12 than camera 7, however, quite the opposite if they came from camera 12.

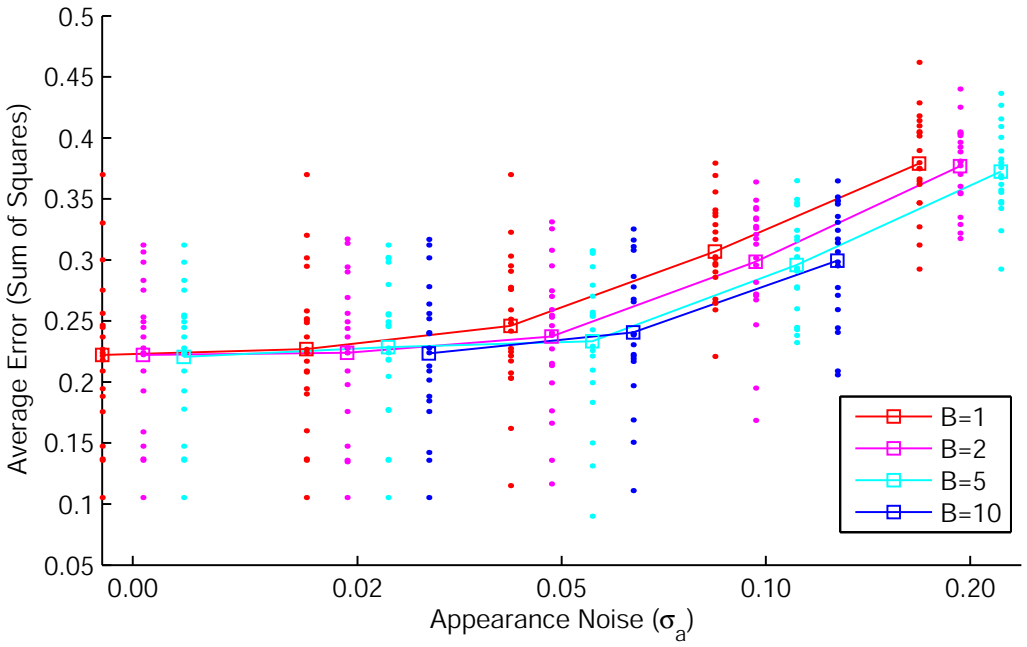


Figure 4.7: **Higher-order Transition Model Errors.** Second-order transition model errors computed using sum of squared errors between ground-truth and estimated distributions. The points represent the results for the 20 randomly-generated networks, boxes indicate the mean across these networks.

Chapter 5

Decentralized Discovery of Camera Network Topology

5.1 Introduction

One of the primary challenges in developing distributed and collaborative sensing systems is providing the constituent sensors with the means to interpret each other's observations and measurements. The absence of global or even pairwise reference information effectively isolates the individual sensors, leaving them unable to determine the meaning or relevance of other sensors' observations. While such reference information can be provided manually to systems comprised of a mere handful of sensors, systems deployed with hundreds and soon thousands of sensors necessitate the development of automated approaches.

The recovery of such reference information is often referred to as *localization* by the sensor network community and as *calibration* by computer vision researchers. Various innovative technologies and techniques have been suggested for both network localization [52, 95, 90, 40] and camera calibration [62, 15, 32, 109, 11]. As the need for localizing/calibrating large numbers of networked sensor is a relatively recent development, novel approaches and techniques continue to be developed.

One of the primary benefits of a camera network is the ability to monitor a larger area than is possible with a single camera. An important monitoring/surveillance task with myriad applications is the tracking of entities (people,

vehicles, animals, etc.) through the camera network. Passively tracking people and automobiles through a camera network can be challenging however, due to factors such as visual occlusion and limited camera field-of-view, resulting in unobserved regions "between" cameras.

In the computer vision literature, methods have been proposed for tracking objects through *non-overlapping* sensors. In this scenario, where objects pass in and out of the sensor-monitored regions, it is critical to understand where an object leaving one sensor's field-of-coverage could potentially next appear.

The probability distribution that governs where objects go when they leave one sensor and how long they take to arrive at the next is called the *transition model*. The graph comprised of a node for each sensor and edges between adjacent sensors shall be referred to as the network's *topology*. Here, two cameras are considered *adjacent* if there exists a path between them that an object can follow without crossing through any other cameras.

In Section 5.2, a discussion of previous work in recovering the topology of non-overlapping cameras is provided. Section 5.3 outlines the decentralized approach for estimating a camera network transition model (implicitly defining the topology). Section 5.4 provides the details of the inference mechanism used: a modified multinomial distribution, which accommodates observation uncertainty. Experimental results using this approach are presented in Section 5.5.

5.2 Related Work in Topology Estimation

Decades of work on data association and tracking have recently led to an investigation of camera network topology. Stauffer and Tieu [118] proposed a technique for identifying cameras with overlapping fields of view and estimating the homography (mapping transformation) between the views.

Mandel *et al.* [89] utilized the SPRT (Sequential Probability Ratio Test), a statistical technique for accumulating sequential evidence until a decision can be made about which of two hypotheses holds. The approach effectively identified overlapping cameras, but could not determine adjacency relationships among non-overlapping cameras. Detmold *et al.* [31] sought to recover "activity topology" by identifying overlapping fields of view in thousand camera networks using the principle of *exclusion*: when an object is present in one camera and not in another, then the object's location is not mutually observable. To extend the approach to identify adjacent non-overlapping views, a temporally padded spatial window was used.

Approaches designed for recovering non-overlapping camera topology have also been suggested. Javed *et al.* [68] use ground-truthed trajectory data to construct nonparametric transition models which capture not only the topology but also the pairwise illumination change between cameras. Makris *et al.* [88] attempt to identify adjacent cameras by using cross-correlation and covariance over thousands of observation departure and arrival times. Tieu *et al.* [121] use statistical dependency (mutual information) to estimate transition delay distributions using an MCMC

chain for sampling correspondences between observations. Gilbert and Bowden [55] use an incremental approach, recursively subdividing each camera's view into blocks and looking for temporal correlations between blocks in different cameras.

In previous work, Farrell *et al.* [39], used a Bayesian approach to partition the observations into the individual objects' trajectories. If trajectories can be estimated (essentially linking together the observations into chains, one per object), then not only the first-order topology, but also higher-order transitions can be recovered. One of the primary benefits of such higher-order models is resilience to camera failure. If a node fails, it can be bypassed since the higher-order model describes where moving objects could have gone after they passed through the failed node's field-of-view. Despite these advantages, the approach had at least two shortcomings of consequence: one, it was computationally intensive; and two, mistakes made early in the estimation process could not be corrected, instead incurring additional errors over time.

Nearly all of these methods for recovering camera network topology assume that computation is performed at a single location. Centralized approaches not only raise the concern of a single point of failure. They also generally exhibit poor scalability. To overcome these barriers, our approach utilizes both the sensing and processing capacities of the entire network, estimating the transition model (and with it the topology) in a decentralized fashion.

5.3 Our Approach

To effectively track objects within the camera network, it is necessary to understand the spatial relationships between cameras. If no *a priori* knowledge of camera locations is given, then, in theory, any camera could be adjacent to any other. While it is extremely unlikely that any single camera will be adjacent to all of the others, this possibility must at least be considered.

Consider a camera network consisting of N nodes. As no information is assumed about the location of one camera relative to any other, the only source of available evidence is the observations that each camera makes. The goal is to use these observations to determine, for each camera c_i, the set of neighboring cameras $N(c_i)$ which are adjacent to c_i.

As with most problems, a global or centralized solution is computationally expensive and scales poorly to larger networks. One of the fundamental principles motivating the approach presented here is that a distributed or decentralized solution provides superior scalability. In a camera network, computational resources increase with the number of cameras.

A semi-localized[1] algorithm has therefore been developed which allows each camera c_i to determine (estimate) which cameras c_j are adjacent to it. The result is a transition model describing not only the adjacency but also the relative probabilities that an entity leaving c_i will arrive at the various $c_j \in N(c_i)$. This algorithm

[1]The approach is "semi-localized" in that the processing is done locally, but collecting the other cameras' observations (which comprise part of the needed input) requires communication.

seeks to establish correspondences between the given camera's own observations and observations made at other cameras within a temporal window. The potential correspondences are weighted by two factors: an information-theoretic appearance measure (see Section 5.3.1); and the delay between the observations. While these weighted correspondences are being accumulated as evidence, the underlying topology and transition model are estimated using a modified multinomial distribution (described in Section 5.4).

5.3.1 Information-Theoretic Appearance Matching

This approach relies heavily on the premise that appearance distinctiveness is key to efficient learning. For a given deployment environment, some of the objects to be tracked will generally have ambiguous appearances. On a university campus, for example, jeans and a t-shirt are common attire. Differentiating such similarly dressed individuals can be difficult. On the same campus relatively few individuals will be observed wearing a suit and tie. When nothing is known about the camera network topology, a great deal more can be learned from individuals with more distinctive appearances than can be from those with more typical appearances.

When an individual leaves a given camera, the system ideally could determine which camera they appear in next. If the individual's appearance is ambiguous, it becomes very difficult to determine which of several subsequent observations (at various cameras) corresponds to this individual's next appearance. However, if the individual's appearance is distinctive, there is a higher likelihood of correctly

determining which observation corresponds to the individual's next appearance.

Suppose that an appearance model is available where the prior probability of observing an appearance f is given by the density $A(f)$. An information-theoretic model, inspired by AIC Weighting (Akaike's Information Criterion [2, 20]), is defined as follows

$$D_A(f) = \frac{e^{-\delta_A \cdot A(f)}}{\int e^{-\delta_A \cdot A(f)} df} \tag{5.1}$$

where $D_A(f)$ is the distinctiveness weight for appearance f and δ_A is a weighting coefficient which determines how much to emphasize distinctive appearances (see Figure 5.1) in matching different observations.

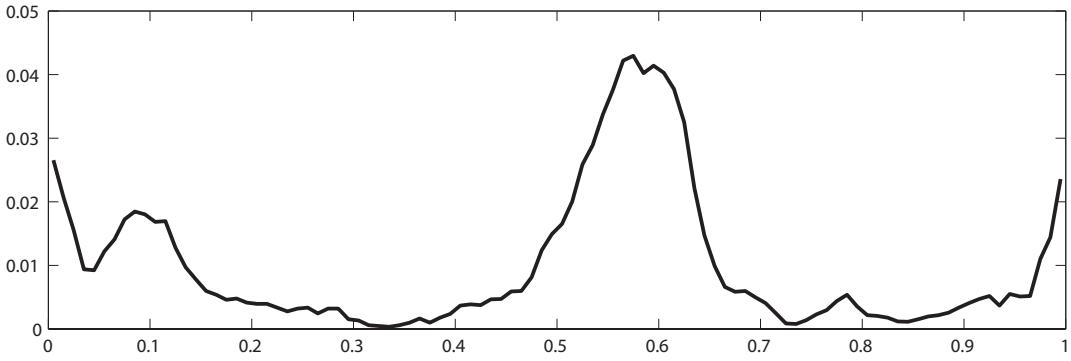
This criterion can be utilized to define a matching score between two appearances. Given two observed appearances f_1 and f_2, a weighted match score M is defined as follows:

$$M(f_1, f_2) = D_A(f_1) \cdot D_A(f_2) \cdot Pr(f_1 = f_2) \tag{5.2}$$

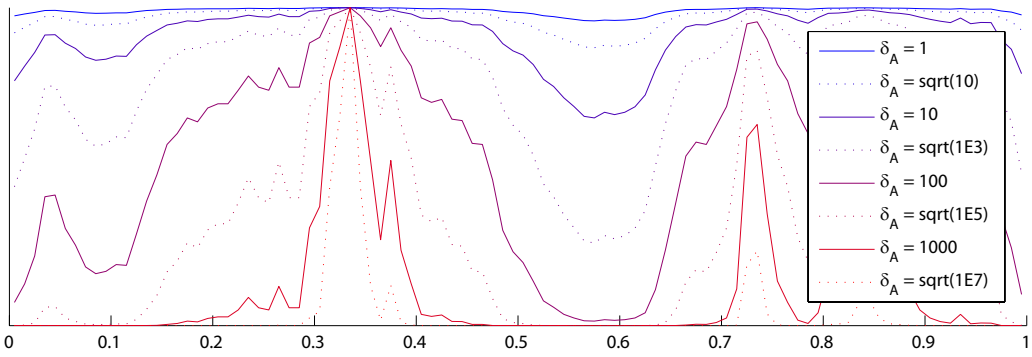
where $Pr(f_1 = f_2)$ denotes the probability that f_1 and f_2 were sampled from the same object (and thus share the same underlying appearance).

5.3.2 Modelling and Estimation Phases

The approach consists of two phases. First, the Modelling Phase, which is conducted offline, consists of 1) acquiring a nonparametric appearance model and 2) determining a coarse distribution over transition delay times. The subsequent online Estimation Phase uses these models to perform sequential (incremental) estimation of the transition model, and hence the topology.



(a) Appearance Density $A(f)$



(b) Distinctiveness Weights with different δ_A

Figure 5.1: **Appearance Distinctiveness.** The above plots demonstrate the distinctiveness weighting described in Section 5.3.1. Given the appearance density $A(f)$ in (a) above, the distinctiveness curves corresponding to different values of δ_A are shown below in (b). While a one-dimensional appearance density has been used for demonstration, the approach can handle more general appearance models.

During the modelling phase, an appearance model can be learned by accumulating observations over a substantial length of time. Given a large sample of appearance observations, it is straightforward to determine $A(f)$. Given the appearance density, constructing the initial transition distributions is done as follows. For two cameras, c_i and c_j, observations are collected over some period of time (these

can be the same observations used to determine $A(f)$). These observations, $\mathbf{o}_i$ and $\mathbf{o}_j$, are used to construct inter-camera time delay densities $T_{i,j}$ as follows (similar to the temporal binning of [55]).

$$T_{i,j}(\hat{t}) \propto \sum_{o_i \in \mathbf{o}_i} \sum_{o_j \in \mathbf{o}_j} K(\hat{t}, t_j - t_i) \cdot \Psi_\tau(t_i, t_j) \cdot M(f_i, f_j) \tag{5.3}$$

where $\Psi_\tau(t_1, t_2)$ is a binary compatibility function defined on the size of the temporal correlation window τ

$$\Psi_\tau(t_1, t_2) = \begin{cases} 1 & \text{if } 0 < t_2 - t_1 \leq \tau \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

$K(\hat{t}, \Delta t)$ is a smoothing or weighting kernel[2], and $M(f_1, f_2)$ is defined in (5.2). This is reminiscent of a weighted cross-correlation.

During the estimation phase, the constructed appearance distinctiveness weights $D_A(f)$ and inter-camera time delay densities $T_{i,j}$ are used as a basis for weighting observations. The modified multinomial distribution uses these weighted observations to estimate the underlying transition model. After describing the observation weighting, we will introduce the modified multinomial distribution.

Suppose camera c_i makes an observation $o = (t_o, f_o)$, indicating that an object with observed appearance f_o exited camera c_i at time t_o. Now consider the set S of observations made at other cameras for objects arriving within a temporal window of length $\tau > 0$. In theory, each of these other observations in S must be considered

[2] $K(\hat{t}, \Delta t)$ allows construction of a nonparametric Kernel Density Estimate (KDE), a technique also referred to as Parzen Windows. The kernel K could be a truncated Gaussian, the Epanechnikov kernel, a triangle filter, etc.

as the potential next appearance of the object observed leaving camera c_i at time t_o. The objective is to estimate:

1. A normalized contribution vector $\mathbf{w}_o$, the j-th component expressing an estimate of the probability that the object originally observed at t_o next appeared at camera c_j.

2. A mixing weight κ_o for this observation based on the distinctiveness of the object, $D_A(f_o)$. The inference procedure should place greater confidence in what is "learned" from distinctive objects.

   For simplicity, define an indicator vector for each camera c_j as

$$\mathbf{e}_j = [\underbrace{0 \ldots 0}_{j-1 \text{ zeros}} \, 1 \, \underbrace{0 \ldots 0}_{N-j \text{ zeros}}] \tag{5.5}$$

hence, $\mathbf{e}_1 = [1\,0 \ldots 0]$, $\mathbf{e}_2 = [0\,1\,0 \ldots 0]$ and $\mathbf{e}_N = [0 \ldots 0\,1]$. Now, the contribution vector for the observation o is defined as

$$\mathbf{w}_o = \sum_{s \in S} M(f_o, f_s) \cdot T_{i,cam(s)}(t_s - t_o) \cdot \mathbf{e}_{cam(s)} \tag{5.6}$$

After processing all S observations, the N-dimensional vector $\mathbf{w}_o$ is normalized, providing a probability density for which camera the object is next observed in.

Over time, a node-specific evidence vector $\boldsymbol{\alpha}$ is accumulated using both the contribution vectors $\mathbf{w}_o$ and the mixing weights $\kappa_o = D_A(f_o)$. After m observations at camera c_i, the evidence vector is given by

$$\boldsymbol{\alpha}^{(m)} = \sum_{o=1}^{m} \kappa_o \mathbf{w}_o \tag{5.7}$$

The mixing weights κ_o cause the observations to be weighted unequally, higher weight being given to more distinctive appearances.

One might question the suggested "decentralized" nature of this approach, given that each node requires the observations of all other cameras. A centralized algorithm performs computation at a single node. Distributed algorithms divide the computation amongst many nodes, often with one node determining the division of labor. The approach presented is decentralized in that all nodes perform processing without any coordination. Observations are broadcast globally and each node gathers the observations to compute its own evidence vector. While the global observations are the same, each node's evidence vector is unique, its own observations determine the weights locally.

In some sense, this accumulated evidence vector $\boldsymbol{\alpha}$ provides us with an estimate of the transition model. However, it is not necessarily the most accurate one. If a coin were flipped and landed face up ("heads"), one wouldn't necessarily assume that the coin would always land face up, although this is what the limited evidence would suggest. This is equivalent to Maximum Likelihood Estimation (MLE). Alternately, one could instead consider all possible values of $Pr(heads)$, and how likely each is given the observations. This approach would compute the expected value of $Pr(heads)$.

Section 5.4 explains how a camera's *expected* transition mo-del is derived from the evidence vector $\boldsymbol{\alpha}$. A modified multinomial distribution is used to represent the probability density over all possible transition models. While a standard Multinomial distribution uses discrete evidence (a die yields either $1, 2, 3, 4, 5$ or 6, an M&M

is either red, or orange, or yellow, etc.), our modified multinomial distribution accommodates uncertainty in the observations.

5.4 The Modified Multinomial Distribution

The mechanism proposed for inferring the transition model from the accumulated evidence is the modified Multinomial distribution. This model provides convenient expressions for both the posterior probability of all possible transition models and most importantly the *expected* transition model.

To motivate the usage of the modified Multinomial distribution, we begin by considering a simple urn problem. A large urn contains m marbles; each marble is either black or white in color. Suppose that b of the m marbles are black. If a marble is drawn from the urn, the probability that it will be black is simply $Pr(B) = b/m$. Similarly, the probability of a white marble being drawn is $Pr(W) = (m-b)/m$. It is assumed that marbles are returned to the urn after they are drawn, thus making the trials independent and keeping $Pr(B)$ and $Pr(W)$ constant.

Suppose now, that instead of a discrete urn model defined by parameters b and m, a continuous model is used where the probabilities of drawing black or white are respectively $Pr(B) = p$ and $Pr(W) = (1-p)$.[3] Drawing a marble from this continuous model (parameterized by p) is a Bernoulli trial. If n marbles are drawn from the urn (n Bernoulli trials performed), the probability of drawing exactly k

[3]This is generally equivalent to selecting b and m such that $b/m = p$, however, we need not restrict ourselves to selecting $p \in \mathbb{Q}$.

black marbles (and $n-k$ white marbles) is given by

$$Pr(|B| = k) = \binom{n}{k} p^k (1-p)^{n-k}, \tag{5.8}$$

known as the binomial distribution.

Suppose n marbles have been drawn, of which k are black and $n-k$ are white. Is it possible to determine what the underlying distribution in the urn (i.e p) is? The value of p cannot be determined exactly, but one can estimate a most likely value (See Appendix A.1.1 for derivation of the Maximum Likelihood Estimate) or even determine the probability distribution over all possible values of p.

Hereafter, the observations will be represented using the vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$. If k of the n marbles drawn were black, then $\boldsymbol{\alpha} = (k, n-k)$. The binomial distribution with parameter p can be rewritten to express the likelihood of $\boldsymbol{\alpha}$ as

$$Pr(\boldsymbol{\alpha}|p) = \frac{(\alpha_1 + \alpha_2)!}{\alpha_1! \alpha_2!} p^{\alpha_1} (1-p)^{\alpha_2} \tag{5.9}$$

since

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{(\alpha_1 + \alpha_2)!}{\alpha_1! \alpha_2!} \tag{5.10}$$

Figure 5.2 shows several plots for various values of $\boldsymbol{\alpha}$ and the corresponding most likely value p_{MLE} and expected value $\mathbb{E}(p|\boldsymbol{\alpha})$, derived below.

From a Bayesian perspective, the posterior distribution over p is proportional to the product of the observation likelihood with the prior

$$Pr(p|\boldsymbol{\alpha}) \propto Pr(\boldsymbol{\alpha}|p) \cdot Pr(p) \tag{5.11}$$

Assuming a uniform prior,

$$Pr(p|\boldsymbol{\alpha}) \propto \frac{(\alpha_1 + \alpha_2)!}{\alpha_1! \alpha_2!} p^{\alpha_1} (1-p)^{\alpha_2} \tag{5.12}$$

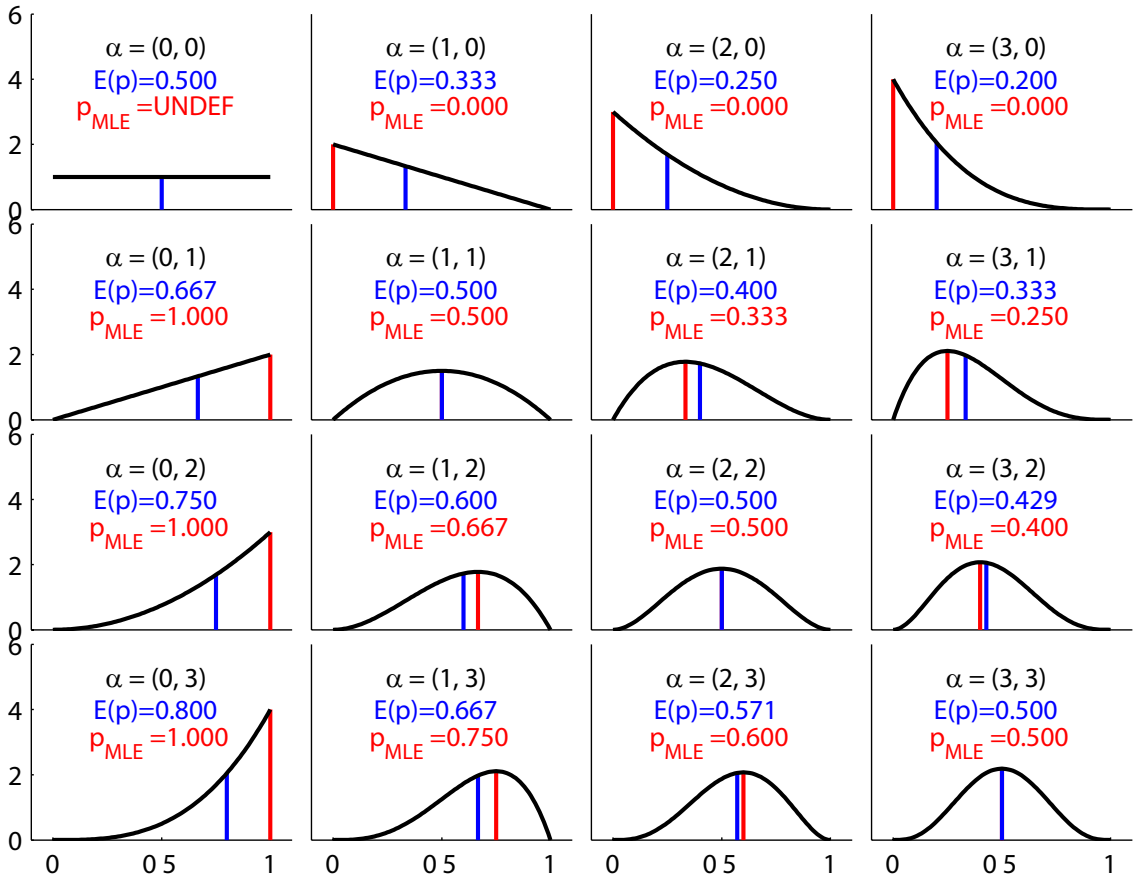


Figure 5.2: **Probability Densities for Binomial Distribution Parameter.** These plots show the probability densities for the binomial distribution parameter p, given the observations $\boldsymbol{\alpha}$ as shown. The Maximum Likelihood Estimate, p_{MLE} (derived in Appendix A.1.1), is shown in red while the expected value, $\mathbb{E}(p|\boldsymbol{\alpha})$, is shown in blue.

To facilitate notation, introduce the Gamma and Beta functions. The Gamma function, defined as

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t}\, dt \tag{5.13}$$

interpolates the factorial function, $\Gamma(n) = (n-1)!$ for positive integers n. The

(binomial) Beta function is defined as

$$\beta(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1}\, dt = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)} \tag{5.14}$$

Thus, the posterior (5.12) can be simplified as

$$Pr(p|\boldsymbol{\alpha}) \propto \frac{1}{\beta(\alpha_1, \alpha_2)} p^{\alpha_1}(1-p)^{\alpha_2} \tag{5.15}$$

For the posterior to be a probability density, the normalization constant C must be found such that

$$\int_0^1 \frac{C}{\beta(\alpha_1, \alpha_2)} p^{\alpha_1}(1-p)^{\alpha_2}\, dp = 1 \tag{5.16}$$

From the definition of the Beta function (5.14), we see that $\int_0^1 t^x(1-t)^y\, dt = \beta(x+1, y+1)$. Applying this equality, along with the identity $\Gamma(x+1) = x\Gamma(x)$, we find that

$$C = \frac{\beta(\alpha_1, \alpha_2)}{\int_0^1 p^{\alpha_1}(1-p)^{\alpha_2} dp} = \frac{\beta(\alpha_1, \alpha_2)}{\beta(\alpha_1+1, \alpha_2+1)} \tag{5.17}$$

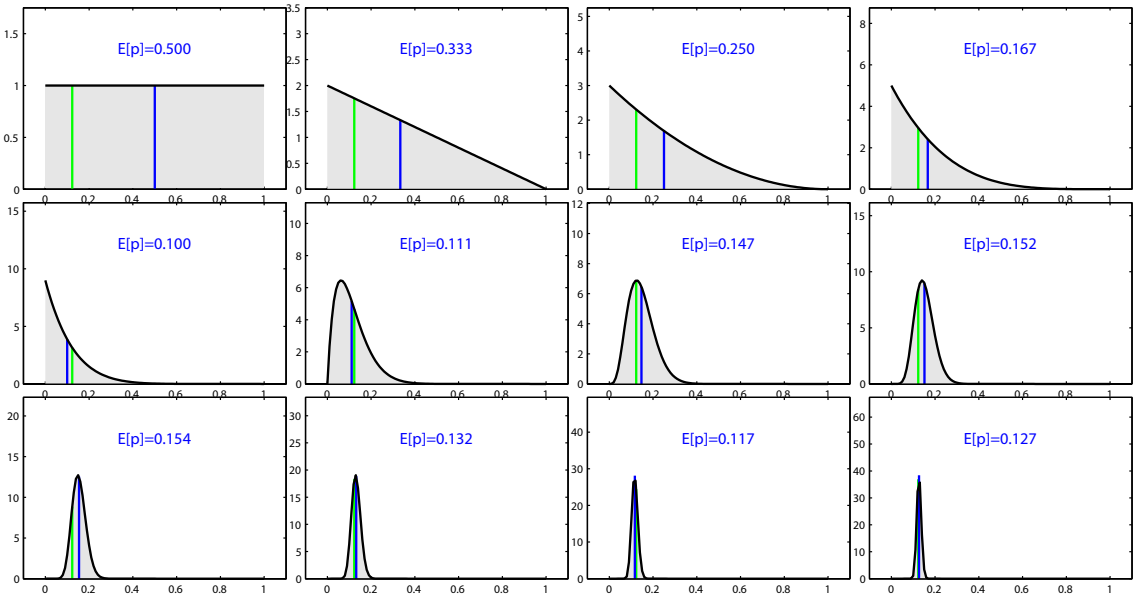
And thus, the *posterior density* is

$$Pr(p|\boldsymbol{\alpha}) = \frac{1}{\beta(\alpha_1+1, \alpha_2+1)} p^{\alpha_1}(1-p)^{\alpha_2} \tag{5.18}$$

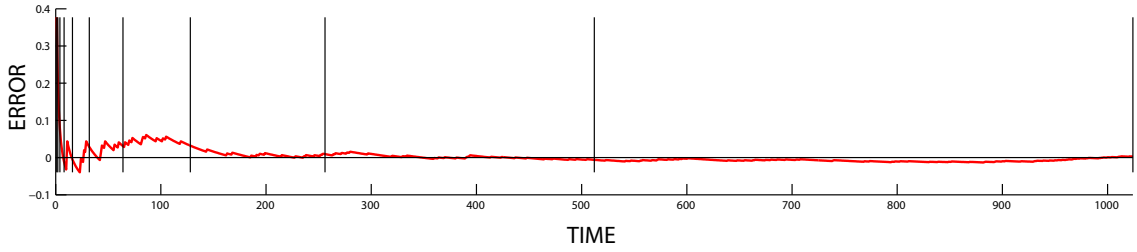
Given this posterior density, the expected value of p is given by (see derivation in A.1.2)

$$\mathbb{E}[p|\boldsymbol{\alpha}] = \frac{(\alpha_1+1)}{(\alpha_1+\alpha_2+2)} \tag{5.19}$$

The sequential estimation process for a binomial distribution is illustrated in Figure 5.3.



(a) Estimation of p at Exponential Intervals



(b) Estimation Error over Time

Figure 5.3: **Binomial Distribution Parameter Estimation.** (a) shows the sequential estimates of the underlying parameter, p, of a binomial distribution. In each plot, the ground truth value of $p = 0.123$ is shown in green. The posterior $P(p|\boldsymbol{\alpha})$ given the observations/evidence accumulated $\boldsymbol{\alpha}$ is shown by the curve. The current expectation $\mathbb{E}(p|\boldsymbol{\alpha})$ is depicted in blue. Note that the plots are shown at exponentially increasing intervals: after $0, 1, 2, 4, \ldots, 1024$ observations. (b) shows the error over time, computed as the difference between the ground truth value of $p = 0.123$ and the current expectation $\mathbb{E}(p|\boldsymbol{\alpha})$.

5.4.1 Extending to the Multinomial Distribution

We now revisit the initial assumption that the urn contains only black and white marbles. The two-color problem, modelled by a binomial distribution, can be extended to a multi-color problem, modelled by the multinomial distribution. In the multivariate case, a vector of probabilities denoted by $\mathbf{p} = (p_1, p_2, \ldots p_d)$, $\sum_i p_i = 1$, is used to describe the underlying distribution across the d colors. The probability of drawing color i is p_i. Utilizing the same approach that was used for the binomial distribution, the d-dimensional posterior density is given by

$$Pr(\mathbf{p}|\boldsymbol{\alpha}) = \frac{1}{\beta(\boldsymbol{\alpha}+1)} \prod_{i=1}^{d} {p_i}^{\alpha_i} \tag{5.20}$$

with the Beta function extended to d dimensions as

$$\beta(\boldsymbol{\alpha}+1) = \frac{\Gamma(\alpha_1+1)\Gamma(\alpha_2+1)\cdots\Gamma(\alpha_d+1)}{\Gamma(\alpha_1+\alpha_2+\ldots+\alpha_d+d)} \tag{5.21}$$

The expectation $\mathbb{E}[\mathbf{p}|\alpha]$ can be written as

$$\mathbb{E}[\mathbf{p}|\boldsymbol{\alpha}] = (\mathbb{E}[p_1|\boldsymbol{\alpha}], \mathbb{E}[p_2|\boldsymbol{\alpha}], \ldots, \mathbb{E}[p_d|\boldsymbol{\alpha}]) \tag{5.22}$$

where (see derivation in Appendix A.2)

$$\mathbb{E}[p_i|\boldsymbol{\alpha}] = \frac{(\alpha_i+1)}{(\alpha_1+\alpha_2+\cdots+\alpha_d+d)} \tag{5.23}$$

This produces the following expectation over $\mathbf{p}$:

$$\mathbb{E}[\mathbf{p}|\boldsymbol{\alpha}] = \frac{(\alpha_1+1, \alpha_2+1, \ldots, \alpha_d+1)}{\alpha_1+\alpha_2+\cdots+\alpha_d+d} \tag{5.24}$$

5.4.2 Uncertain Observations

Thus far, each marble has been considered to have a single deterministic color (once drawn). However, the mathematical derivations have made no assumption

requiring $\alpha_i \in \mathbb{Z}$. In fact, when we replaced the combinatorial expressions with $\Gamma(\cdot)$ and $\beta(\cdot)$ functions, we equipped our model to handle any observation vector $\boldsymbol{\alpha}$, not just $\boldsymbol{\alpha} \in \mathbb{Z}^d$.

The multi-camera tracking problem, in fact, requires one to allow uncertainty in the observations. Under this paradigm, marbles represent objects and colors represent cameras. A camera observing an object leaving its field-of-view is an event analogous to drawing a marble. With no prior knowledge of the network's topology, we cannot know which camera it will appear in next, hence the color of the marble is uncertain.

As the other cameras subsequently report objects they observe arriving, those cameras (colors) receive increased weight according to the observed object similarity. Thus, each observation (think marble) produces a contribution vector (w_o from Section 5.3.2) which can be any convex combination of the cameras (colors) as described in (5.6) above. These contributions are aggregated into the evidence vector $\boldsymbol{\alpha}$ as described in (5.7).

5.5 Experimental Results

The experiments use a JAVA simulation environment (see Figure 5.4). The environment provides a network of sensor nodes, each node coupling a processing unit with one or more sensors. The network may be modelled as either wired, where any node can communicate with any other, or wireless, where only a subset of the network is in a given node's neighborhood. The sensor interface has been designed

such that various modalities could be simulated, but at present only cameras are used.

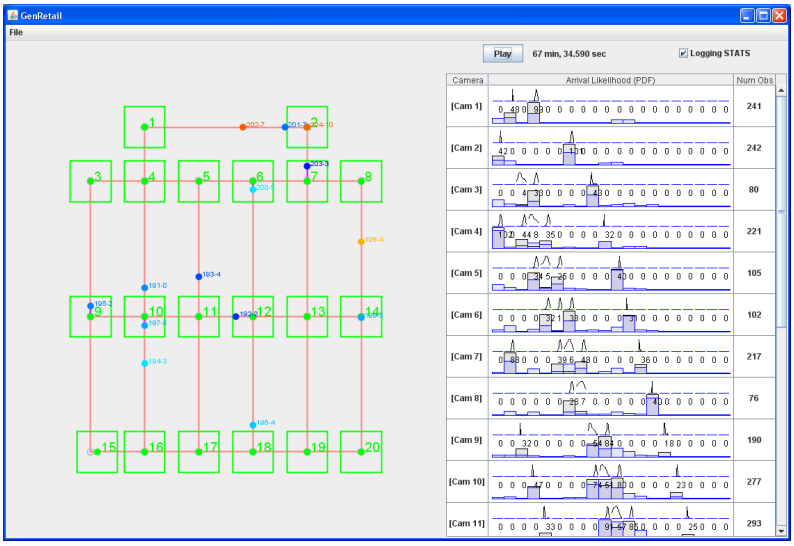


Figure 5.4: **Simulation Environment.** Written in Java, the experimental frameworks simulates a distributed camera network. It currently implements the decentralized approach to topology estimation, but is extensible for other distributed sensing tasks.

The simulation environment provides a synchronized time model which effectively allows the nodes, the sensors, and the physical entities moving around the scene, to perform their respective functions in parallel. The nodes are currently implemented to process and communicate the camera observations, applying the approach for decentralized estimation of the transition model and topology.

5.5.1 Experimental Application: A Supermarket

The application used in the experiments is that of a retail environment. A supermarket is a prototypical scenario for the following reasons:

- In a supermarket, each customer/shopper generally follows a different route. Simulating this makes the experiments both realistic and non-trivial.
- While some aisles of a supermarket are visited more frequently, most parts of the store receive a fair amount of traffic. This should make it possible to recover the topology of all of the cameras.
- The variety of appearances observed over time enables the learning approach, and at the same time makes observation correspondence a nontrivial challenge.

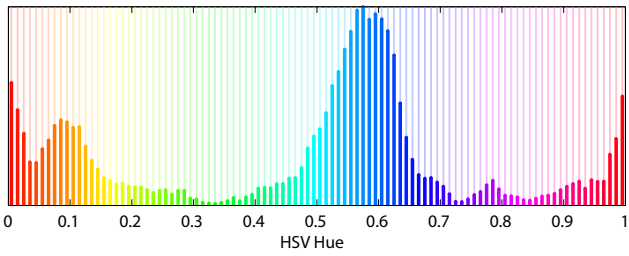
In this retail environment, there are various aisles which we'll imagine contain the items which shoppers will collect and purchase. The timing of shoppers entering the supermarket is governed by a Poisson departure process (parameters $\lambda = 1$ and the same τ used for the temporal correlation window). Each shopper has a list of items to pick up (actually a list of locations within the store to visit) and follows an optimal (shortest) route to collect these items. When finished, they depart through the same entrance they arrived at.

5.5.2 Appearance Model

The simulation environment can be outfitted with whatever appearance model is desired. For simplicity, a single parameter (HSV hue) empirically-derived appear-

ance model is used (see Figure 5.5).

Each shopper is assigned an intrinsic appearance, sampled from this nonparametric density. When an individual shopper passes through a camera, the intrinsic appearance is perturbed by $\mathcal{N}(0, \sigma_A)$, yielding noisy observations.



(a) Empirically-Derived Appearance Model A_0

Figure 5.5: **Nonparametric Appearance Model A_0.** The appearance model A_0 in (a), which is used in our retail simulation, is derived empirically from the HSV hue of the vehicles in a parking lot image (the photographer did not respond to our request for permission to publish the image, we therefore cannot include the image here).

5.5.3 Simulation Parameters

There are a number of factors that influence the simulation and consequently any tracking efforts. Before examining the effect of these various factors, baseline results are presented (see Figure 5.6) on a 20-camera network, using a temporal correlation window of $\tau = 20$ seconds. These results show the performance, given by the mean error per camera (using $L2$-norm on the transition model) over time. The Modelling and Estimation phases are each run for 5 hours of simulation time.

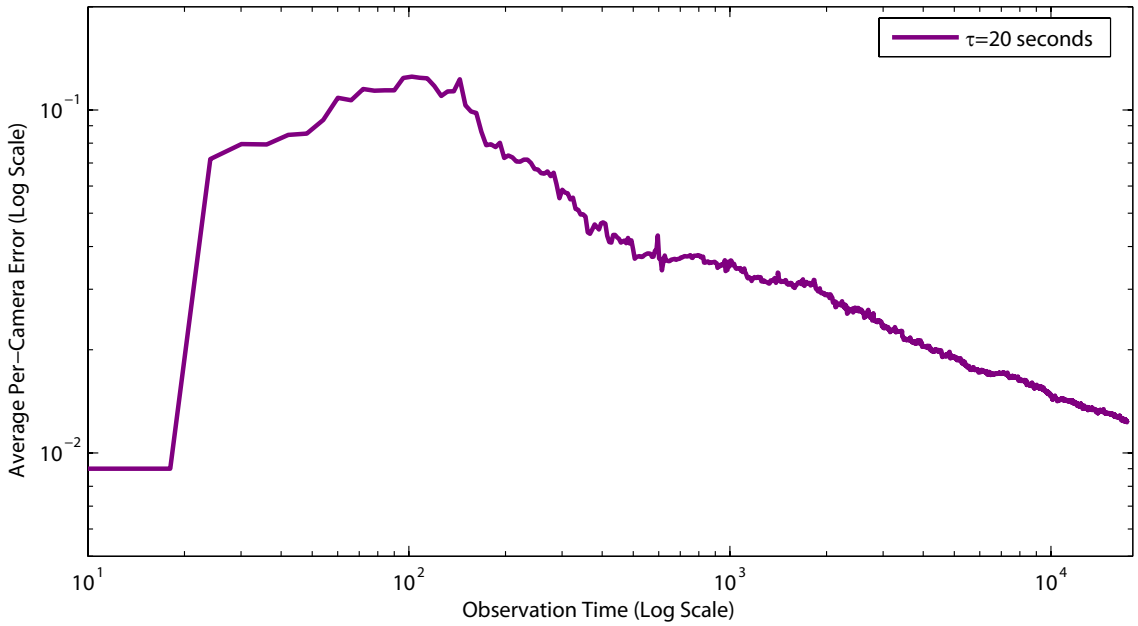


Figure 5.6: **Estimation Results.** This plot depicts the error between the estimated transition model and the ground truth transition model over the course of the estimation process. This simulation contained 20 cameras, used a correlation window of $\tau = 20$ seconds with the appearance model A_0 shown in Figure 5.5(a). The errors are computed using the $L2$-norm on a per-camera basis.

The impact that various simulation and model parameters have on the algorithm's ability to recover the topology and tracking model have been studied and are presented below.

- **Number of Cameras/Nodes** - While the initial results were generated on networks of 20 camera nodes, simulations have been conducted for networks ranging in size from 10 cameras to 100 cameras. These results are presented in Figure 5.7. The size of the area being monitored increases in proportion to the number of cameras, keeping the sensor density approximately fixed.

- **Appearance Entropy** - From the empirical nonparametric appearance model A (see Section 5.5.2 above), k additional appearance densities are generated with varying entropy. Using these new densities in the simulation enables exploring how increasing or decreasing the entropy of the appearance model affects recovery of the topology/tracking model. These results are presented in Figure 5.8.

- **Distinctiveness Weights** - For the appearance model A, we described in (5.1) an information-theoretic model for weighting appearance distinctiveness. By varying the distinctiveness parameter δ_A, we see the influence of the weighting on the estimation results in Figure 5.9.

- **Correspondence Window** - The correspondence window implicitly defines the maximum separation of two cameras (in terms of inter-camera transit time) which allows recovery of their adjacency. The size of the temporal correspondence interval or window τ is varied (however, the poisson departure distribution parameter $\tau = 20$ is not). These results are presented in Figure 5.10.

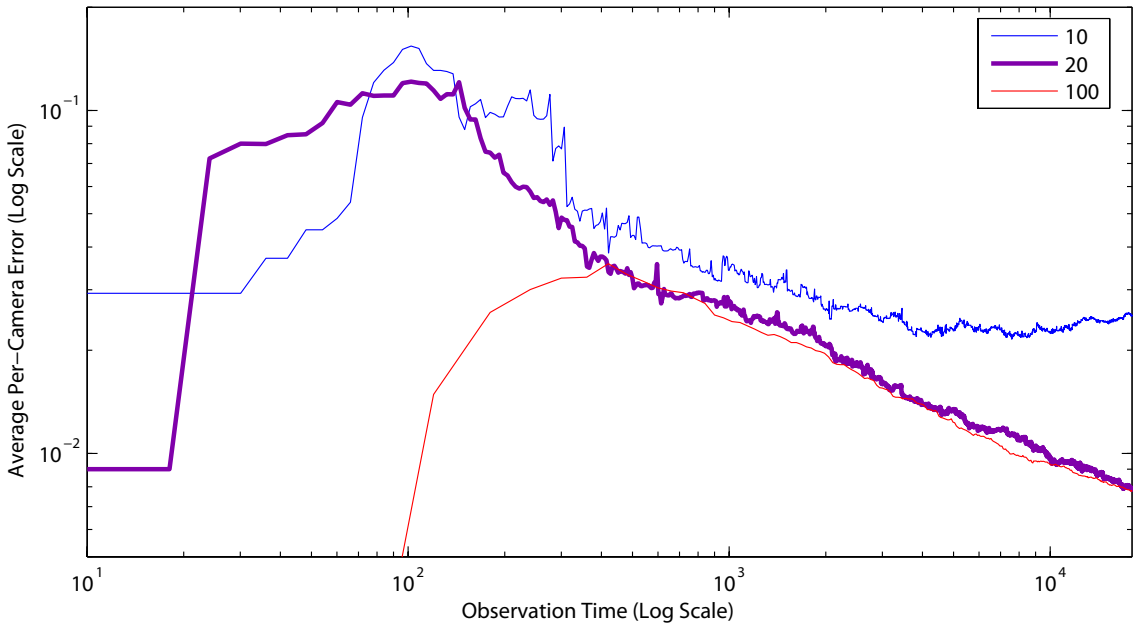
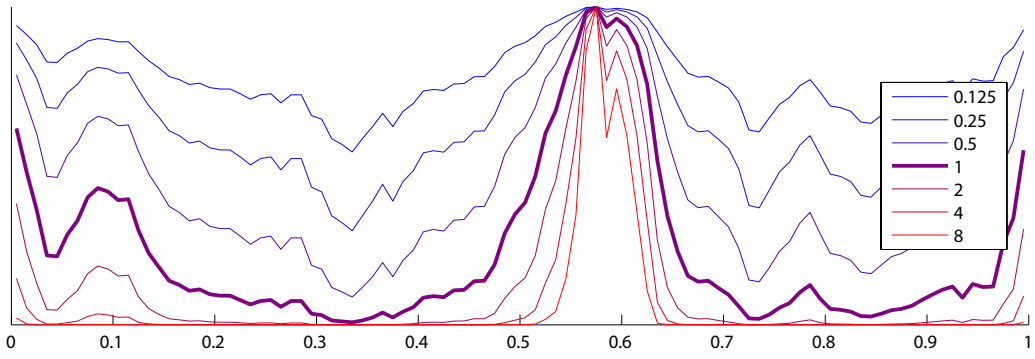
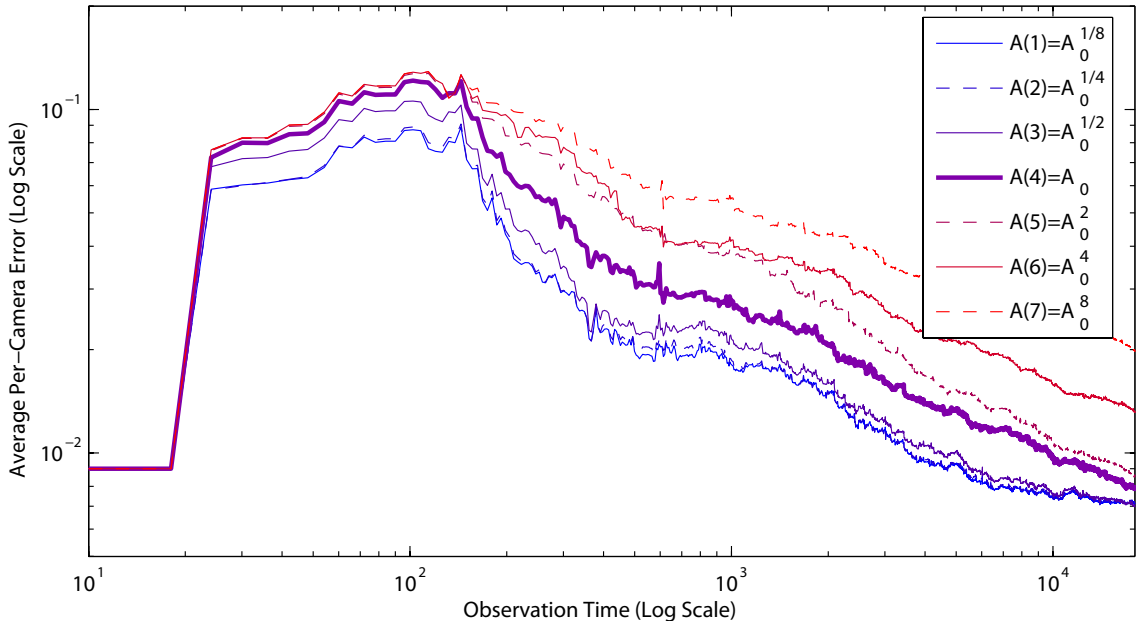


Figure 5.7: **Scalability - Varying the Camera Network Size.** Here the effects of changing the camera network size (number of nodes) are shown using the same metric as in the baseline, Figure 5.6. The number of cameras is varied from 10 to 100.



(a) Changing the Appearance Entropy



(b) Comparison of Estimation Results

Figure 5.8: **Varying the Entropy of the Appearance Density** A**.** A family of appearance densities, with varying entropy, is derived from the original density A_0. In (a), the original density A_0 (in bold purple) is shown along with the derived densities $A(i) \propto {A_0}^{2^a}$ for $a = \pm 1, 2, 3$. The influence of changing the appearance density entropy is shown in (b), comparing the error curves for the derived densities with that of the original, A_0.

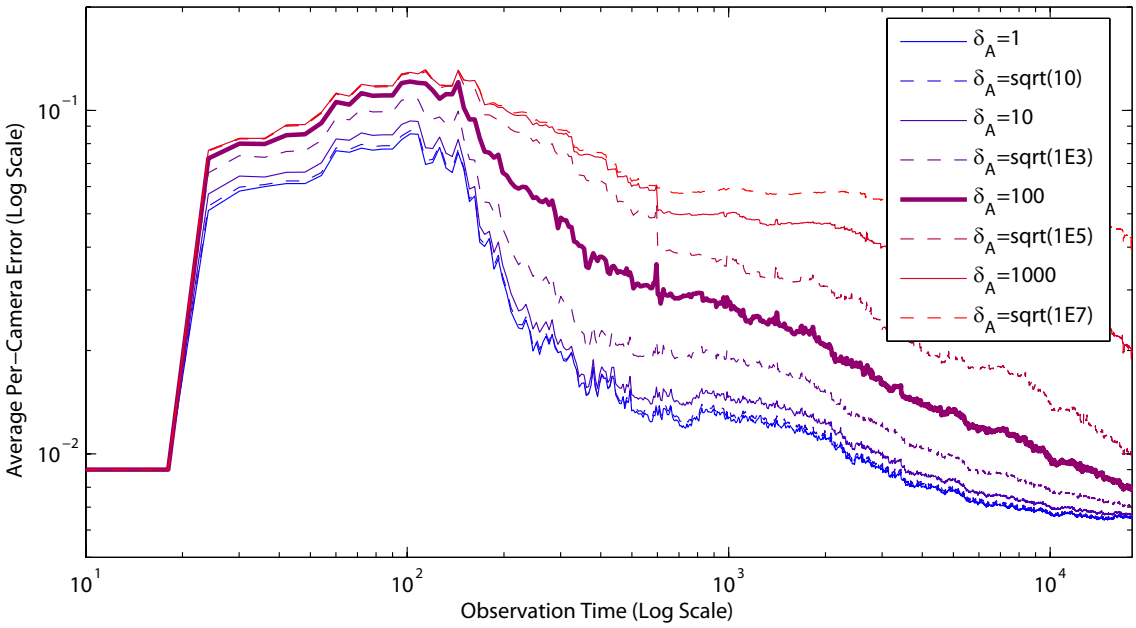


Figure 5.9: **Varying the Distinctiveness Weights.** Varying the parameter δ_A changes the Distinctiveness Weighting Function $D_A(f)$ (See Figure 5.1). Adjusting δ_A affects the estimation convergence rate. Results are shown for $\delta_A \in 10^{\frac{1}{2}(0,1,2,3,4,5,6,7)}$.

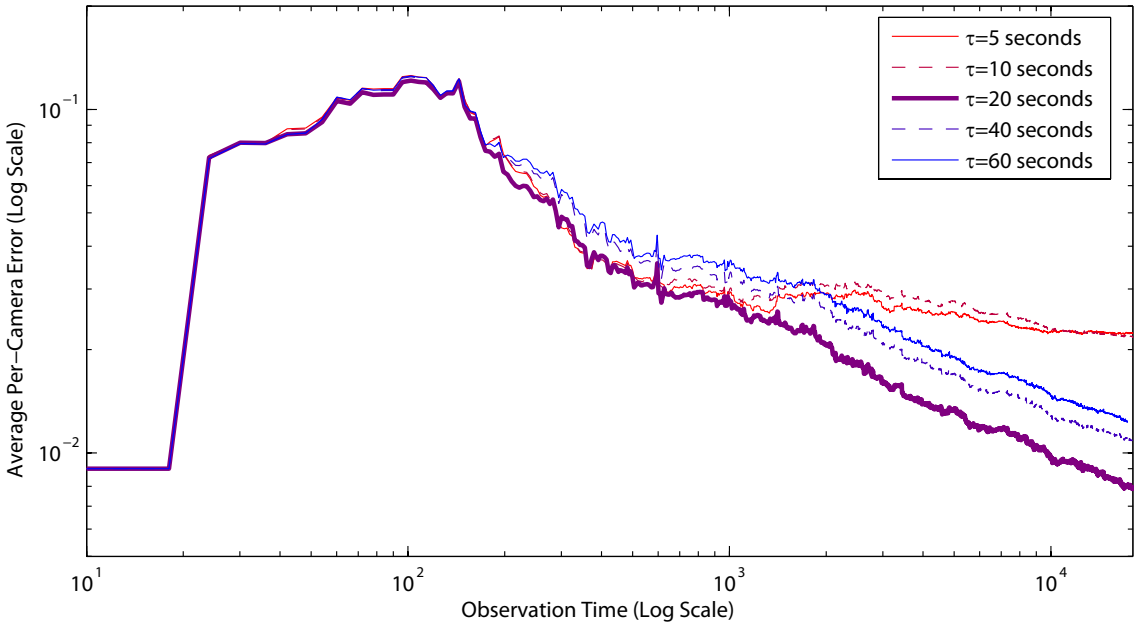


Figure 5.10: **Varying the Temporal Correlation Window.** Here the effects of changing the size of the temporal correlation window, τ, can be observed in comparison with the baseline value ($\tau = 20$s). The value of τ varies from from 5 to 60 seconds.

Chapter 6

Conclusion

6.1 Concluding Remarks and Future Work

As outlined in the introduction, this work addresses three different subtasks under the broader heading of learning visual patterns. We now provide some concluding remarks for each topic, together with directions for future research.

6.2 Subordinate Categorization

We have presented an approach for subordinate categorization using a pose-normalized appearance model founded upon a volumetric part model. Whereas basic-level categories are represented by a part model (a set of volumetric primitives,) the variation in shape and appearance properties of these parts across a taxonomy provides the cues needed for subordinate categorization.

Our model associates the underlying image pattern parameters used for detection with corresponding volumetric part location, scale and orientation parameters. This parameters implicitly define a mapping from the image pixels into a pose-normalized appearance space, which removes view and pose dependency, facilitating subordinate categorization.

Some additional directions we are currently investigating include

Broader Annotated Dataset: First, the subset of the CUB-200 dataset used in our experiments is but a fraction of what we would like to utilize. We have therefore undertaken the task of annotating additional families within the CUB-200 dataset. The CUB-200 dataset was not collected with taxonomy as a primary goal, leaving out not just families, but entire orders of the avian taxonomy. We are therefore also working to collect a broader dataset, increasing the number of species to more than 500, with a greater number of examples per category.

Improving Detection Accuracy: Part of the reason that our bottom-line subordinate categorization results were not better was the difficult of accurately detecting the birds. While we believe that additional annotations will help, we continue to work on improvements to the detection process, perhaps investing alternatives to the poselet framework.

Classification Model: In this work, we combined the pose-normalized appearance descriptor (PNAD) with a Random Forest classifier framework. This combination holds potential, however, a more thorough evaluation and comparison using other classifiers and/or variants of the current PNAD descriptor should be conducted.

6.3 Kinetic Accessibility

While these results are very promising, we strongly believe that the full capability of the kinetic accessibility paradigm has yet to be realized. It can and should be applied to domains other than basketball, but we feel that even within this con-

text, the logic rules can be enhanced and extended to effectively provide recognition of more perceptually subtle events such as turnovers, baskets made or missed (based on what happens subsequently), *etc*.

One insight of great interest was discovered while comparing the pairwise `CanPass(`P_1`,`P_2`)` values for a given frame. Players on the defensive team typically have most or all of their values close to 1.0 (indicating unimpeded passing opportunities if they had the ball). On the offensive team however, the values vary greatly due to defensive players obstructing passing lanes. Rarely is an offensive player free to pass to all four teammates. This is a rather intuitive result — the offense isn't concerned with obstructing the defense's passing lanes. As implemented, the team assignments (which players are on which teams) were provided. However, the assignments could likely be learned using this approach, particularly in conjunction with other information such as who was observed passing to whom, who seems to be covering whom, *etc*.

Additional directions for investigation include

Better Handling in the Temporal Domain: At present, we have treated the temporal dimension only superficially, we would suggest a more robust model of time such as that presented by Morariu and Davis [96].

Capture the Flag: Capture the flag is an ideal domain for our logic-based kinetic accessibility framework. Sadilek and Kautz [115] already investigated using Markov Logic Networks to model capture the flag, but the coarse 1m resolution in their dataset would not be amenable to kinetic accessibility.

Multi-Target Tracking: Another application that we are investigating is that of multi-target tracking and data association. Often data association in tracking relies more on appearance similarity than on dynamics. Kinetic accessibility provides an ideal framework in which to jointly consider the dynamics and identities of multiple objects.

6.4 Estimation of Camera Network Topology

6.4.1 Centralized Estimation

We have presented a technique for constructing higher-order statistical transition models. The approach is based on recovering object trajectories by partitioning the observation set in a Bayesian Framework. We described the Bayesian framework for determining partition likelihood by evaluation of a probabilistic graphical model. We adopt an incremental approach, adding observations and pruning unlikely partitions, retaining only the most probable partitions after each iteration. Having recovered the trajectories we are able to extract not only camera adjacency but also higher-order topological relationships which can improve tracking accuracy and offers topological redundancy, fortifying against camera failures.

We feel that this technique holds promise for recovering the topology information for camera networks. To more fully realize this potential, we propose further work on the following areas

Overlapping Field-of-View: At present we make the assumption that all fields of view are non-overlapping. While it facilitates our present approach, this con-

straint inhibits the analysis of more general camera networks where cameras may or may not overlap.

Scalability: While the approach we present is described as a serial algorithm, it is inherently parallel and could be implemented on a medium-scale network of "smart cameras", each possessing the computational resources to process its own video and also collaborate in distributed topology estimation. (see Bramberger, *et al.* [14] for such a platform). At each iteration, a large number of partitions are evaluated to determine the partition likelihood. The overhead required to divvy up the partitions amongst the camera nodes and gather/prune the results would be minimal (constitutes only 3.80% of the present serial implementation run-time).

Real Camera Network/Tracking: We also want to test our data on an actual deployment of 20 or more cameras. We initially plan to apply our algorithm to the 9-camera Terrascope dataset [69] from the U. of Kentucky. We further wish to verify that our higher-order transition model will improve tracking performance.

6.4.2 Decentralized Estimation

To successfully track entities within a camera network, an understanding of camera topology is essential. In this paper, a decentralized technique has been presented for each node in the network to recover its own topological neighbors. Each node estimates adjacent cameras by combining its own observations with those

of the other cameras in the network. The vehicle for this distributed inference is a modified multinomial distribution. The observations are correlated using an information-theoretic weighting model, assessing appearance distinctiveness.

Through this work, the authors have identified a few areas for further study as outlined below

Per-camera Observation Frequency: Cameras with higher observation frequencies have a higher prior for finding matches in the correlation window due simply to the fact that more objects pass through within the correspondence window, independent of the underlying topology.

Multiple Cameras during Correlation Interval: Another difficulty introduced by the semi-local approach is that when an object passes through multiple cameras during the correlation interval, there is no way to explicitly know whether the resulting observations were all due the same object or to multiple different objects. An approach that balances weighting the earlier observations in the interval (to somewhat suppress subsequent observations along the same trajectory) with those that seem "best" might further improve results.

Convergence: It is clearly evident that the estimation convergence time is dependent on parameters such as: the number of adjacent cameras; the total number of cameras; and the per-camera observation frequency. It may be possible to use a confidence measure such as covariance to control convergence of the estimation process.

Appendix A

Appendix - Expectation Derivations

A.1 Derivations for the Binomial Distribution

A.1.1 Maximum Likelihood Estimate (MLE)

For a binomial distribution of unknown parameter p and observations $\boldsymbol{\alpha}$, the Maximum Likelihood Estimate (MLE) of p is determined as follows:

$$\frac{d}{dp} Pr(p|\boldsymbol{\alpha}) = 0 \tag{A.1}$$

Which, drawing from Equation (5.18) means

$$\frac{1}{\beta(\alpha_1 + 1, \alpha_2 + 1)} \frac{d}{dp} p^{\alpha_1}(1-p)^{\alpha_2} = 0 \tag{A.2}$$

Differentiating, we find that

$$\left(\alpha_1 \cdot p^{\alpha_1 - 1}(1-p)^{\alpha_2} - \alpha_2 \cdot p^{\alpha_1}(1-p)^{\alpha_2 - 1}\right) = 0 \tag{A.3}$$

or, simplifying,

$$\alpha_1 \cdot (1-p) = \alpha_2 \cdot p \tag{A.4}$$

which at last yields the Maximum Likelihood Estimate

$$p_{MLE} = \frac{\alpha_1}{\alpha_1 + \alpha_2} \tag{A.5}$$

A.1.2 Expectation (for p)

For a binomial distribution of unknown parameter p, the expected value of p is derived based on the observations $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ as follows

$$\begin{aligned}
\mathbb{E}[p|\boldsymbol{\alpha}] &= \int_0^1 p \cdot Pr(p|\boldsymbol{\alpha})\, dp \\
&= \frac{1}{\beta(\alpha_1+1, \alpha_2+1)} \int_0^1 p^{\alpha_1+1}(1-p)^{\alpha_2}\, dp \\
&= \frac{\beta(\alpha_1+2, \alpha_2+1)}{\beta(\alpha_1+1, \alpha_2+1)} \\
&= \frac{\Gamma(\alpha_1+2)\Gamma(\alpha_2+1)}{\Gamma(\alpha_1+\alpha_2+3)} \cdot \frac{\Gamma(\alpha_1+\alpha_2+2)}{\Gamma(\alpha_1+1)\Gamma(\alpha_2+1)} \\
&= \frac{(\alpha_1+1)\Gamma(\alpha_1+1)\Gamma(\alpha_2+1)}{(\alpha_1+\alpha_2+2)\Gamma(\alpha_1+\alpha_2+2)} \cdot \\
&\quad \frac{\Gamma(\alpha_1+\alpha_2+2)}{\Gamma(\alpha_1+1)\Gamma(\alpha_2+1)} \\
&= \frac{(\alpha_1+1)}{(\alpha_1+\alpha_2+2)}
\end{aligned} \tag{A.6}$$

A.2 Expectation of a Multinomial Distribution

The expectation $\mathbb{E}[\mathbf{p}|\boldsymbol{\alpha}]$ of the multinomial distribution can be derived as it was in the binomial case. First note that this is a vector valued expectation $\mathbb{E}[\mathbf{p}|\boldsymbol{\alpha}] = (\mathbb{E}[p_1|\boldsymbol{\alpha}], \mathbb{E}[p_2|\boldsymbol{\alpha}], \ldots, \mathbb{E}[p_d|\boldsymbol{\alpha}])$.

The following integral [117] will be helpful:

$$\int_0^a x^m \left(a^n - x^n\right)^p dx = \frac{a^{m+1+np}\Gamma[(m+1)/n]\Gamma(p+1)}{n\Gamma[(m+1)/n+p+1]} \tag{A.7}$$

which, when $n = 1$, can conveniently be written as

$$\int_0^a x^m (a-x)^p \, dx = \frac{a^{m+p+1}\Gamma(m+1)\Gamma(p+1)}{\Gamma(m+p+2])}$$
$$= a^{m+p+1}\beta(m+1, p+1) \tag{A.8}$$

Computing $\mathbb{E}[p_i|\boldsymbol{\alpha}]$ is a little trickier in the d-dimensional (multivariate) case. Without loss of generality, let us compute $\mathbb{E}[p_1|\boldsymbol{\alpha}]$. Furthermore, the value $d = 4$ will be used, as the generalization (and notation) for large d is cleaner in an example with a small value for d. The general form will be given, just derived using a small value of d for clarity. Note that since $\sum_0^d p_i = 1$, then p_d is a function of $p_1, p_2, \ldots, p_{d-1}$. For example, when $d = 4$, $p_4 = 1 - p_1 - p_2 - p_3$.

$$\mathbb{E}[p_1|\boldsymbol{\alpha}] = \int_0^1 p_1 \int_0^{1-p_1} \int_0^{1-p_1-p_2} Pr(p_1|\boldsymbol{\alpha}) \, dp_3 \, dp_2 \, dp_1 \tag{A.9}$$
$$\propto \int_0^1 p_1^{\alpha_1+1} \int_0^{1-p_1} p_2^{\alpha_2} \int_0^{1-p_1-p_2} p_3^{\alpha_3} (1-p_1-p_2-p_3)^{\alpha_4} \, dp_3 \, dp_2 \, dp_1 \tag{A.10}$$

By application of (A.8), using $a = (1 - p_1 - p_2)$, one can repeatedly evaluate the innermost integral, beginning with (A.10), accumulating the $\beta(\cdot)$ terms as follows

$$\int_0^1 p_1^{\alpha_1+1} \int_0^{1-p_1} p_2^{\alpha_2} \underbrace{\int_0^{(1-p_1-p_2)} p_3^{\alpha_3} ((1-p_1-p_2)-p_3)^{\alpha_4} \, dp_3}_{=(1-p_1-p_2)^{\alpha_3+\alpha_4+1}\beta(\alpha_3+1,\alpha_4+1)} \, dp_2 \, dp_1 \tag{A.11}$$
$$\propto \int_0^1 p_1^{\alpha_1+1} \underbrace{\int_0^{(1-p_1)} p_2^{\alpha_2} ((1-p_1)-p_2)^{\alpha_3+\alpha_4+1} dp_2}_{=(1-p_1)^{\alpha_2+\alpha_3+\alpha_4+2}\beta(\alpha_2+1,\alpha_3+\alpha_4+2)} \, dp_1 \tag{A.12}$$
$$\propto \underbrace{\int_0^1 p_1^{\alpha_1+1} (1-p_1)^{\alpha_2+\alpha_3+\alpha_4+2} dp_1}_{=\beta(\alpha_1+2,\alpha_2+\alpha_3+\alpha_4+3)} \tag{A.13}$$

Then, accumulating this chain of $\beta(\cdot)$ terms (including the original $1/\beta(\boldsymbol{\alpha}+1)$ from

$Pr(p_1|\boldsymbol{\alpha})$), yields

$$\frac{1}{\beta(\boldsymbol{\alpha}+1)} \cdot \beta(\alpha_1+2, \alpha_2+\alpha_3+\alpha_4+3) \cdot \beta(\alpha_2+1, \alpha_3+\alpha_4+2) \cdot$$

$$\beta(\alpha_3+1, \alpha_4+1) \tag{A.14}$$

$$= \frac{\Gamma(\alpha_1+\alpha_2+\alpha_3+\alpha_4+4)}{\Gamma(\alpha_1+1)\Gamma(\alpha_2+1)\Gamma(\alpha_3+1)\Gamma(\alpha_4+1)} \cdot$$

$$\frac{\Gamma(\alpha_1+2)\Gamma(\alpha_2+\alpha_3+\alpha_4+3)}{\Gamma(\alpha_1+\alpha_2+\alpha_3+\alpha_4+5)} \cdot$$

$$\frac{\Gamma(\alpha_2+1)\Gamma(\alpha_3+\alpha_4+2)}{\Gamma(\alpha_2+\alpha_3+\alpha_4+3)} \cdot \frac{\Gamma(\alpha_3+1)\Gamma(\alpha_4+1)}{\Gamma(\alpha_3+\alpha_4+2)} \tag{A.15}$$

$$= \frac{\Gamma(\alpha_1+2) \cdot \Gamma(\alpha_1+\alpha_2+\alpha_3+\alpha_4+4)}{\Gamma(\alpha_1+1) \cdot \Gamma(\alpha_1+\alpha_2+\alpha_3+\alpha_4+5)} \tag{A.16}$$

$$= \frac{[(\alpha_1+1)\Gamma(\alpha_1+1)] \cdot \Gamma(\alpha_1+\alpha_2+\alpha_3+\alpha_4+4)}{\Gamma(\alpha_1+1) \cdot [(\alpha_1+\alpha_2+\alpha_3+\alpha_4+4)\Gamma(\alpha_1+\alpha_2+\alpha_3+\alpha_4+4)]}$$

$$= \frac{(\alpha_1+1)}{(\alpha_1+\alpha_2+\alpha_3+\alpha_4+4)} \tag{A.17}$$

This result generalizes to higher dimensions. The expectation for p_i in the d-dimensional case is

$$\mathbb{E}[p_i|\boldsymbol{\alpha}] = \frac{(\alpha_i+1)}{(\alpha_1+\alpha_2+\cdots+\alpha_d+d)} \tag{A.18}$$

Bibliography

[1] J. K. Aggarwal and Qin Cai. Human Motion Analysis: A Review. *CVIU*, 73(3):428–440, March 1999. 37

[2] Hirotugu Akaike. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974. 86

[3] Anonymous. New basketball dataset. available online: `http://anonymous/`. 37, 52

[4] Boris Babenko, Steve Branson, and Serge Belongie. Similarity Metrics for Categorization: From Monolithic to Category Specific. In *ICCV*, 2009. 13

[5] Aharon Bar-Hillel and Daphna Weinshall. Subordinate Class Recognition Using Relational Object Models. In *NIPS*, 2007. 12

[6] Evgeniy Bart, Ian Porteous, Pietro Perona, and Max Welling. Unsupervised Learning of Visual Taxonomies. In *CVPR*, 2008. 13

[7] Tamara Berg, Alexander Berg, and Jonathan Shih. Automatic Attribute Discovery and Characterization from Noisy Web Data. In *ECCV*, 2010. 14

[8] Irving Biederman. Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, 94(2):115 – 147, April 1987. 4, 11, 14, 16, 21

[9] Irving Biederman, Suresh Subramaniam, Moshe Bar, Peter Kalocsai, and József Fiser. Subordinate-level object classification reexamined. *Psychological Research*, 62(2-3):131–153, July 1999. 11, 15

[10] Aaron F. Bobick and James W. Davis. The Recognition of Human Movement Using Temporal Templates. *PAMI*, 23(3):257–267, March 2001. 37

[11] Jean-Yves Bouget. *Camera Calibration Toolbox for MATLAB*, 2010. Software available at: `http://www.vision.caltech.edu/bouguetj/calib_doc/`. 80

[12] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting People Using Mutually Consistent Poselet Activations. In *ECCV*, 2010. 4, 11, 20, 29

[13] Lubomir Bourdev and Jitendra Malik. Poselets: Body Part Detectors Trained Using 3D Human Pose Annotations. In *ICCV*, 2009. 4, 11, 20, 29

[14] Michael Bramberger, Andreas Doblander, Arnold Maier, Bernhard Rinner, and Helmut Schwabach. Distributed Embedded Smart Cameras for Surveillance Applications. *IEEE Computer*, 39(2):68–75, February 2006. 58, 111

[15] Matthew Brand, Matthew E. Antone, and Seth J. Teller. Spectral Solution of Large-Scale Extrinsic Camera Calibration as a Graph Embedding Problem. In *ECCV*, 2004. 80

[16] Matthew Brand and Vera Kettnaker. Discovery and Segmentation of Activities in Video. *PAMI*, 22(8):844–851, August 2000. 38

[17] Steve Branson, Catherine Wah, Boris Babenko, Florian Schroff, Peter Welinder, Pietro Perona, and Serge Belongie. Visual Recognition with Humans in the Loop. In *ECCV*, 2010. 14, 28, 32

[18] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001. 25

[19] Michael C. Burl, Thomas K. Leung, and Pietro Perona. Face Localization via Shape Statistics. In *International Workshop on Automatic Face and Gesture Recognition*, 1995. 12

[20] Kenneth P. Burnham and David R. Anderson. *Model Selection and Multi-Model Inference*. Springer-Verlag, Second edition, 2002. Chapter 2. 86

[21] Michael T. Chan, Anthony Hoogs, Rahul Bhotika, Amitha Perera, John Schmiederer, and Gianfranco Doretto. Joint Recognition of Complex Events and Track Matching. *CVPR*, 2006. 39

[22] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at: `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. 29

[23] Christopher James Needham. *Tracking and Modelling of Team Game Interactions*. PhD dissertation, The University of Leeds, School of Computing, October 2003. 40

[24] Maurice Chu, Sanjoy Mitter, and Feng Zhao. Distributed Multiple Target Tracking and Data Association in Ad Hoc Sensor Networks. In *International Conference on Information Fusion*, 2003. 62

[25] Timothy F. Cootes, Gareth J. Edwards, and Christopher J Taylor. Active Appearance Models. *PAMI*, 23(6):681–685, June 2001. 11

[26] Ingemar J. Cox and Sunita L. Hingorani. An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking. *PAMI*, 18(2):138–150, February 1996. 61

[27] David Crandall, Pedro Felzenszwalb, and Daniel Huttenlocher. Spatial Priors for Part-Based Recognition using Statistical Models. In *CVPR*, 2005. 12

[28] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005. 23

[29] Jia Deng, Alexander Berg, Kai Li, and Li Fei-Fei. What Does Classifying More Than 10,000 Image Categories Tell Us? In *ECCV*, 2010. 13

[30] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 13

[31] Henry Detmold, Anton van den Hengel, Anthony Dick, Alex Cichowski, Rhys Hill, Ekim Kocadag, Katrina Falkner, and David S. Munro. Topology Estimation for Thousand-Camera Surveillance Networks. *ICDSC*, 2007. 82

[32] Dhanya Devarajan, Richard J. Radke, and Haeyong Chung. Distributed Metric Calibration of Ad Hoc Camera Networks. *ACM Transactions on Sensor Networks*, 2(3):380–403, August 2006. 80

[33] Piotr Dollar, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior Recognition via Sparse Spatio-Temporal Features. In *VS-PETS Workshop*, 2005. 38

[34] Alexei A. Efros, Alexander C. Berg, Greg Mori, and Jitendra Malik. Recognizing Action at a Distance. In *ICCV*, 2003. 38

[35] Tim Ellis. *Pinnawela Elephant Orphanage.* Flickr photo available under Creative Commons license at: `http://www.flickr.com/photos/tim_ellis/4639998087/`. 3

[36] Ali Farhadi, Ian Endres, and Derek Hoiem. Attribute-Centric Recognition for Cross-category Generalization. In *CVPR*, 2010. 13

[37] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing Objects by their Attributes. In *CVPR*, 2009. 13

[38] Ryan Farrell and Larry S. Davis. Decentralized Discovery of Camera Network Topology. In *ICDSC*, 2008. 7

[39] Ryan Farrell, David Doermann, and Larry S. Davis. Learning Higher-order Transition Models in Medium-scale Camera Networks. In *OMNIVIS Workshop, ICCV*, 2007. 7, 83

[40] Ryan Farrell, Roberto Garcia, Dennis Lucarelli, Andreas Terzis, and I-Jeng Wang. Localization in Multi-Modal Sensor Networks. In *ISSNIP*, 2007. 80

[41] Li Fei-Fei, Rob Fergus, and Pietro Perona. A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories. In *ICCV*, 2003. 14

[42] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-Shot Learning of Object Categories. *PAMI*, 28(4):594–611, April 2006. 14

[43] Pedro Felzenszwalb, Ross Girshick, David McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part Based Models. *PAMI*, 32(9):1627–1645, September 2010. 12

[44] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial Structures for Object Recognition. *International Journal of Computer Vision*, 61(1):55–79, January 2005. 12

[45] Andras Ferencz, Erik G. Learned-Miller, and Jitendra Malik. Building a Classification Cascade for Visual Identification from One Example. In *ICCV*, 2005. 14

[46] Rob Fergus, Hector Bernal, Yair Weiss, and Antonio Torralba. Semantic Label Sharing for Learning with Many Categories. In *ECCV*, 2010. 13

[47] Rob Fergus, Pietro Perona, and Andrew Zisserman. A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition. In *CVPR*, 2005. 12

[48] Vittorio Ferrari, Frédéric Jurie, and Cordelia Schmid. From Images to Shape Models for Object Detection. *IJCV*, 87(3):284–303, May 2010. 12

[49] Martin A. Fischler and Robert A. Elschlager. The Representation and Matching of Pictorial Structures. *IEEE Transactions on Computers*, C-22(1):67 – 92, Jan 1973. 12

[50] Mark S. Fox. *Constraint-directed Search: A Case Study of Job-Shop Scheduling*. Morgan Kaufmann Publishers, 1987. 66

[51] FP7 APADIS Project. *Apidis Dataset*, 2009. Dataset available at: `http://www.apidis.org/Dataset/`. 51

[52] Stanislav Funiak, Carlos Guestrin, Mark A. Paskin, and Rahul Sukthankar. Distributed localization of networked cameras. In *IPSN*, 2006. 62, 80

[53] James J. Gibson. *The Ecological Approach To Visual Perception*. Houghton Mifflin, Boston, 1979. 15

[54] James J. Gibson and Laurence E. Crooks. A Theoretical Field-Analysis of Automobile-Driving. *The American Journal of Psychology*, 51(3):453–471, July 1938. 35

[55] Andrew Gilbert and Richard Bowden. Tracking Objects Across Cameras by Incrementally Learning Inter-camera Colour Calibration and Patterns of Activity. In *ECCV*, 2006. 83, 88

[56] Shaogang Gong and Tao Xiang. Recognition of Group Activities using Dynamic Probabilistic Networks. In *ICCV*, 2003. 38

[57] Yihong Gong, Mei Han, Wei Hua, and Wei Xu. Maximum Entropy Model-based Baseball Highlight Detection and Classification. *CVIU*, 96(2):181–199, November 2004. 40

[58] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as Space-Time Shapes. *PAMI*, 29(12):2247–2253, December 2007. 38

[59] Gregory Griffin and Pietro Perona. Learning and Using Taxonomies for Fast Visual Categorization. In *CVPR*, 2008. 13

[60] Abhinav Gupta, Praveen Srinivasan, Jianbo Shi, and Larry S. Davis. Understanding Videos, Constructing Plots, Learning a Visually Grounded Storyline Model from Annotated Videos. In *CVPR*, 2009. 40

[61] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1), 2009. 29

[62] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. 80

[63] Kenneth E. Hoff, III, Tim Culver, John Keyser, Ming Lin, and Dinesh Manocha. Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware. In *SIGGRAPH*, 1999. 52

[64] Somboon Hongeng and Ramakant Nevatia. Multi-Agent Event Recognition. In *ICCV*, 2001. 39

[65] Timothy Huang and Stuart J. Russell. Object Identification in a Bayesian Context. In *IJCAI*, 1997. 61

[66] Stephen S. Intille and Aaron F. Bobick. Recognizing Planned, Multiperson Action. *CVIU*, 81(3):414–445, March 2001. 40

[67] Yuri A. Ivanov and Aaron F. Bobick. Recognition of Multi-agent Interaction in Video Surveillance. In *ICCV*, 1999. 39

[68] Omar Javed, Zeeshan Rasheed, Khurram Shafique, and Mubarak Shah. Tracking Across Multiple Cameras With Disjoint Views. In *ICCV*, 2003. 61, 82

[69] Christopher Jaynes, Amit Kale, Nathaniel Sanders, and Etienne Grossmann. The Terrascope Dataset: Scripted Multi-Camera Indoor Video Surveillance with Ground-truth. In *VS-PETS05*, 2005. 111

[70] Michael J. Jones and Tomaso Poggio. Multidimensional Morphable Models: A Framework for Representing and Matching Object Classes. *IJCV*, 29(2):107–131, August 1998. 11

[71] Imran Junejo, Emilie Dexter, Ivan Laptev, and Patrick Perez. Cross-View Action Recognition from Temporal Self-similarities. In *ECCV*, 2008. 38

[72] Chan-Hyun Kang, Jung-Rae Hwang, and Ki-Joune Li. Trajectory Analysis for Soccer Players. In *International Conference on Data Mining Workshops*, 2006. 40

[73] Yan Ke, Rahul Sukthankar, and Martial Hebert. Efficient Visual Event Detection Using Volumetric Features. In *ICCV*, 2005. 38

[74] Yan Ke, Rahul Sukthankar, and Martial Hebert. Event Detection in Crowded Videos. In *ICCV*, 2007. 38

[75] Vera Kettnaker and Ramin Zabih. Bayesian Multi-Camera Surveillance. In *CVPR*, 1999. 62

[76] Stanley Kok, Marc Sumner, Matthew Richardson, Parag Singla, Hoifung Poon, Daniel Lowd, Jue Wang, and Pedro Domingos. The Alchemy System for Statistical Relational AI. Technical Report, Department of Computer Science and Engineering, University of Washington, Seattle, WA. `http://alchemy.cs.washington.edu`, 2009. 50

[77] Matej Kristan, Janez Pers, Matej Perse, and Stanislav Kovacic. Closed-world Tracking of Multiple Interacting Targets for Indoor-Sports Applications. *CVIU*, 113(5):598–611, May 2009. 40

[78] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and Simile Classifiers for Face Verification. In *ICCV*, 2009. 13

[79] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to Detect Unseen Object Classes by Between-Class Attribute Transfer. In *CVPR*, 2009. 13

[80] Ivan Laptev and Tony Lindeberg. Space-time Interest Points. In *ICCV*, 2003. 38

[81] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning Realistic Human Actions from Movies. In *CVPR*, 2008. 38

[82] Lei Chen and Müjdat Çetin and Alan S. Willsky. Distributed Data Association for Multi-Target Tracking in Sensor Networks. In *International Conference on Information Fusion*, 2005. 62

[83] Riccardo Leonardi, Pierangelo Migliorati, and Maria Prandini. Semantic Indexing of Soccer Audio-Visual Sequences: A Multimodal Approach Based on Controlled Markov Chains. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):634–643, May 2004. 40

[84] Kurt Lewin. *Principles of Topological Psychology.* McGraw-Hill, New York, 1936. 35

[85] Ruonan Li, Rama Chellappa, and Shaohua Kevin Zhou. Learning Multi-modal Densities on Discriminative Temporal Interaction Manifold for Group Activity Recognition. In *CVPR*, 2009. 39

[86] Hsuan-Tien Lin, Chih-Jen Lin, and Ruby Weng. A Note on Platt's Probabilistic Outputs for Support Vector Machines. *Machine Learning*, 68(3):267–276, October 2007. 29

[87] Chunxi Liu, Qingming Huang, Shuqiang Jiang, Liyuan Xing, Qixiang Ye, and Wen Gao. A Framework for Flexible Summarization of Racquet Sports Video Using Multiple Modalities. *CVIU*, 113(3):415–424, March 2009. 40

[88] Dimitrios Makris, Tim Ellis, and James Black. Bridging the Gaps between Cameras. In *CVPR*, 2004. 61, 69, 75, 82

[89] Zehavit Mandel, Ilan Shimshoni, and Daniel Keren. Multi-camera Topology Recovery from Coherent Motion. In *ICDSC*, 2007. 82

[90] Miklós Maróti, Péter Völgyesi, Sebestyén Dóra, Branislav Kusý, András Nádas, Ákos Lédeczi, György Balogh, and Károly Molnár. Radio Interferometric Geolocation. In *SENSYS*, 2005. 80

[91] David Marr and H. K. Nishihara. Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 200(1140):269–294, February 1978. 14

[92] Marcin Marszałek and Cordelia Schmid. Constructing Category Hierarchies for Visual Recognition. In *ECCV*, 2008. 13

[93] Gonzalo Martínez-Muñoz, Natalia Larios, Eric Mortensen, Wei Zhang, Asako Yamamuro, Robert Paasch, Nadia Payet, David Lytle, Linda Shapiro, Sinisa Todorovic, Andrew Moldenke, and Thomas G. Dietterich. Dictionary-Free Categorization of Very Similar Objects via Stacked Evidence Trees. In *CVPR*, 2009. 11, 12, 25

[94] Erik Miller, Nicholas Matsakis, and Paul Viola. Learning from One Example Through Shared Densities on Transforms. In *CVPR*, 2000. 14

[95] David Moore, John Leonard, Daniela Rus, and Seth J. Teller. Robust Distributed Network Localization with Noisy Range Measurements. In *SENSYS*, 2004. 80

[96] Vlad Morariu and Larry S. Davis. Multi-agent Event Recognition in Structured Scenarios. In *CVPR*, 2011. 109

[97] Surya Nepal, Uma Srinivasan, and Graham Reynolds. Automatic Detection of 'Goal' Segments in Basketball Videos. In *ACM Multimedia*, 2001. 40

[98] Maria-Elena Nilsback and Andrew Zisserman. A Visual Vocabulary for Flower Classification. In *CVPR*, 2006. 12

[99] Maria-Elena Nilsback and Andrew Zisserman. Automated Flower Classification over a Large Number of Classes. In *ICVGIP*, 2008. 12

[100] Songhwai Oh, Stuart Russell, and Shankar Sastry. Markov Chain Monte Carlo Data Association for Multiple-Target Tracking. *IEEE Transactions on Automatic Control*, 54(3):481–497, March 2009. 62, 64

[101] Nuria M. Oliver, Barbara Rosario, and Alex P. Pentland. A Bayesian Computer Vision System for Modeling Human Interactions. *PAMI*, 22(8):831–843, August 2000. 38

[102] Vasu Parameswaran and Rama Chellappa. View Invariants for Human Action Recognition. In *CVPR*, 2003. 38

[103] Sangho Park and J. K. Aggarwal. Recognition of Two-person Interactions Using a Hierarchical Bayesian Network. In *First ACM SIGMM International Workshop on Video Surveillance*. ACM, 2003. 38

[104] Hanna Pasula, Stuart J. Russell, Michael Ostland, and Yaacov Ritov. Tracking Many Objects with Many Sensors. In *IJCAI*, 1999. 62, 64

[105] Kadir A. Peker. Rapid Generation of Sports Video Highlights using the MPEG-7 Motion Activity Descriptor. In *Proceedings of SPIE*, 2001. 40

[106] Janez Pers, Marta Bon, and Goran Vuckovic. *CVBASE 06 Dataset, ECCV 2006 Workshop*, 2006. Dataset available at: `http://vision.fe.uni-lj.si/cvbase06/dataset.html`. 51

[107] Matej Perse, Matej Kristan, Stanislav Kovacic, Goran Vuckovic, and Janez Pers. A Trajectory-based Analysis of Coordinated Team Activity in a Basketball Game. *CVIU*, 113(5):612–621, May 2009. 40

[108] Ronald Poppe. A Survey on Vision-based Human Action Recognition. *Image and Vision Computing*, 28(6):976–990, June 2010. 37

[109] Ali Rahimi, Brian Dunagan, and Trevor Darrell. Simultaneous Calibration and Tracking with a Network of Non-Overlapping Sensors. In *CVPR*, 2004. 80

[110] Clive Reid. *Elephant, Hluhluwe-iMfolozi Game Reserve*. Flickr photo available under Creative Commons license at: `http://www.flickr.com/photos/kleinz/3552012856/`. 3

[111] Donald B. Reid. An Algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, December 1979. 61, 65

[112] Matthew Richardson and Pedro Domingos. Markov Logic Networks. *Machine Learning*, 62(1):107–136, February 2006. 5, 37, 49

[113] Eleanor Rosch, Carolyn B. Mervis, Wayne D. Gray, David M. Johnson, and Penny Boyes-Braem. Basic Objects in Natural Categories. *Cognitive Psychology*, 8(3):382–439, July 1976. 10, 15

[114] Yong Rui, Anoop Gupta, and Alex Acero. Automatically extracting highlights for TV baseball programs. In *ACM Multimedia*, 2000. 40

[115] Adam Sadilek and Henry Kautz. Recognizing Multi-Agent activities from GPS data. In *AAAI*, 2010. 39, 109

[116] William Robson Schwartz, Aniruddha Kembhavi, David Harwood, and Larry S. Davis. Human Detection using Partial Least Squares Analysis. In *ICCV*, 2009. 46

[117] Murray R. Spiegel. *Mathematical Handbook of Formulas and Tables*. McGraw-Hill, 1968. (Equation 15.24, Page 95). 114

[118] Chris Stauffer and Kinh Tieu. Automated Multi-camera Planar Tracking Correspondence Modeling. In *CVPR*, 2003. 58, 60, 82

[119] Eran Swears and Anthony Hoogs. Learning and Recognizing American Football Plays. In *Learning Workshop*, 2010. 40

[120] Michael J. Tarr, Pepper Williams, William G. Hayward, and Isabel Gauthier. Three-dimensional Object Recognition is Viewpoint Dependent. *Nature Neuroscience*, 1(4):275–277, August 1998. 14

[121] Kinh Tieu, Gerald Dalley, and W. Eric L. Grimson. Inference of Non-Overlapping Camera Network Topology by Measuring Statistical Dependence. In *ICCV*, 2005. 61, 69, 75, 82

[122] Sinisa Todorovic and Narendra Ahuja. Learning Subcategory Relevances for Category Recognition. In *CVPR*, 2008. 13

[123] Son D. Tran and Larry S. Davis. Event Modeling and Recognition Using Markov Logic Networks. In *ECCV*, 2008. 49

[124] Pavan Turaga, Rama Chellappa, V.S. Subrahmanian, and Octavian Udrea. Machine Recognition of Human Activities: A Survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, November 2008. 37

[125] Namrata Vaswani, Amit Roy Chowdhury, and Rama Chellappa. Activity Recognition Using the Dynamics of the Configuration of Interacting Objects. In *CVPR*, 2003. 39

[126] Andrea Vedaldi and Brian Fulkerson. *VLFeat: An Open and Portable Library of Computer Vision Algorithms*, 2010. Software available at: `http://www.vlfeat.org/`. 31

[127] Gang Wang and David Forsyth. Joint Learning of Visual Attributes, Object Classes and Visual Saliency. In *ICCV*, 2009. 13

[128] Markus Weber, Max Welling, and Pietro Perona. Unsupervised Learning of Models for Recognition. In *ECCV*, 2000. 12

[129] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 14, 17, 28

[130] Alper Yilmaz and Mubarak Shah. Actions Sketch: a Novel Action Representation. In *CVPR*, 2005. 38

[131] Yang Yu. *Human Appearance Modeling in Visual Surveillance*. Master's thesis, University of Maryland, College Park, USA, August 2007. 58

[132] Wojciech Zajdel. *Bayesian Visual Surveillance*. PhD thesis, Universiteit van Amsterdam, Amsterdam, Nederlands, January 2006. 62, 64, 68, 73

[133] Wojciech Zajdel, Ali Taylan Cemgil, and Ben J. A. Kröse. Dynamic Bayesian Networks for Visual Surveillance with Distributed Cameras. In *EuroSSC*, 2006. 62

[134] Wojciech Zajdel and Ben J. A. Kröse. A Sequential Bayesian Algorithm for Surveillance with Nonoverlapping Cameras. *IJPRAI*, 19(8):977–996, December 2005. 62

[135] Wensheng Zhou, Asha Vellaikal, and C. C. Jay Kuo. Rule-based Video Classification System for Basketball Video Indexing. In *ACM Workshops on Multimedia*, 2000. 40

[136] Yue Zhou, Shuicheng Yan, and Thomas S. Huang. Pair-Activity Classification by Bi-Trajectories Analysis. In *CVPR*, 2008. 39

[137] Alon Zweig and Daphna Weinshall. Exploiting Object Hierarchy: Combining Models from Different Category Levels. In *ICCV*, 2007. 13