

VISUAL HUMAN TRACKING AND GROUP ACTIVITY ANALYSIS:  
A VIDEO MINING SYSTEM FOR RETAIL MARKETING

Alex Leykin

Submitted to the faculty of the Graduate School  
in partial fulfillment of the requirements  
for the degree  
Doctor of Philosophy  
in Computer Science and Cognitive Science  
Indiana University

December 2007

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements of the degree of Doctor of Philosophy.

Doctoral  
Committee

---

Mihran Tuceryan, Ph.D.  
(Principal Advisor)

---

Andrew Hanson, Ph.D.

---

Steven Johnson, Ph.D.

---

Edward Robertson, Ph.D.

November 30, 2007

---

Raymond R. Burke, Ph.D.

Copyright © 2008

Alex Leykin

ALL RIGHTS RESERVED



# Acknowledgements

I would like to give thanks to the members of my committee for the indispensable advice they provided me with. To my advisor Mihran Tuceryan for taking on a challenge of advising a student from another campus. Special thanks also goes to Ray Burke, for recognizing unique collaboration opportunities in bridging business and computer science and providing me with his expert advice in the domain of retail marketing.

In a separate paragraph, I would like to sincerely thank my wonderful wife and partner, Inna Kouper. She has provided me with the best collegial experience and our daily discussions help me broaden my views on science, life and everything. Thanks to her, I no longer think that maths and hacking is all there is to it.

My friends and colleagues were of great help with their key contribution being: never doubting my mental abilities. My big bro' Anton Leykin on occasion helped me in sorting out some involved mathematical equations over the phone or in Skype. The members of my extended family, including my parents, helped me a great deal in creating the sense of urgency and making me believe that the impending defense day is unavoidable.

Last but not least, I would like to thank many little things that helped me keep my spirits up throughout this process. This includes but is not limited to: a good sense of humor, The Flying Spaghetti Monster, fine Belgian Wheat Ale, get-togethers with friends and beautiful sunny days in Bloomington, IN.

# Abstract

In this thesis we present a system for automatic human tracking and activity recognition from video sequences. The problem of automated analysis of visual information in order to derive descriptors of high level human activities has intrigued computer vision community for decades and is considered to be largely unsolved. A part of this interest is derived from the vast range of applications in which such a solution may be useful. We attempt to find efficient formulations of these tasks as applied to the extracting customer behavior information in a retail marketing context. Based on these formulations, we present a system that visually tracks customers in a retail store and performs a number of activity analysis tasks based on the output from the tracker.

In tracking we introduce new techniques for pedestrian detection, initialization of the body model and a formulation of the temporal tracking as a global trans-dimensional optimization problem. Initial human detection is addressed by a novel method for head detection, which incorporates the knowledge of the camera projection model. The initialization of the human body model is addressed by newly developed shape and appearance descriptors. Temporal tracking of customer trajectories is performed by employing a human body tracking system designed as a Bayesian jump-diffusion filter. This approach demonstrates the ability to overcome model dimensionality ambiguities as people are leaving and entering the scene.

Following the tracking, we developed a two-stage group activity formulation based upon the ideas from swarming research. For modelling purposes, all moving actors in the scene are viewed

here as simplistic agents in the swarm. This allows to effectively define a set of inter-agent interactions, which combine to derive a distance metric used in further swarm clustering. This way, in the first stage the shoppers that belong to the same group are identified by deterministically clustering bodies to detect short term events and in the second stage events are post-processed to form clusters of group activities with fuzzy memberships.

Quantitative analysis of the tracking subsystem shows an improvement over the state of the art methods, if used under similar conditions. Finally, based on the output from the tracker, the activity recognition procedure achieves over 80% correct shopper group detection, as validated by the human generated ground truth results.

**Keywords:** Human Tracking, Human Activity Modeling and Recognition, Swarming, Background Subtraction, Camera Calibration

# Contents

<b>Acknowledgements</b>	<b>v</b>
<b>Abstract</b>	<b>vi</b>
<b>1 Motivation and Problem Statement</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Goals . . . . .	3
1.3 Contributions . . . . .	5
<b>2 Related Work</b>	<b>8</b>
2.1 Human Tracking . . . . .	8
2.2 Activity Recognition . . . . .	11
<b>3 System Overview</b>	<b>15</b>
3.1 Experimental Setup . . . . .	15
3.2 Overview of the Computational Framework . . . . .	16

<b>4</b>	<b>Detection and Tracking</b>	<b>20</b>
4.1	Detection . . . . .	21
4.1.1	Camera Modelling . . . . .	21
4.1.1.1	Perspective Projection Camera . . . . .	21
4.1.1.2	Stitched Panoramic Camera . . . . .	24
4.1.2	Background Modeling and Subtraction . . . . .	26
4.1.3	Finding Head Candidates . . . . .	29
4.2	Tracking . . . . .	33
4.2.1	Bayesian Model: Observations and States . . . . .	33
4.2.2	Computing the Posterior Probability . . . . .	34
4.2.3	Body Shape Modelling . . . . .	35
4.2.4	Body Appearance Modelling . . . . .	36
4.2.5	Priors . . . . .	37
4.2.6	Likelihoods . . . . .	39
4.2.7	Jump-diffusion Transformations . . . . .	43
<b>5</b>	<b>Activity Analysis</b>	<b>47</b>
5.1	Modeling Group Activities . . . . .	47
5.2	Obtaining Shopper Trajectories . . . . .	51
5.2.1	Path Trajectory Smoothing . . . . .	51
5.3	Event Detection . . . . .	52

5.3.1	Deterministic Agglomerative Clustering of Bodies . . . . .	53
5.4	Activity Detection . . . . .	55
5.4.1	Fuzzy Agglomerative Clustering of Events . . . . .	57
5.5	Benchmarking Against Single Step Activity Detectors . . . . .	61
<b>6</b>	<b>Experimental Results</b>	<b>64</b>
6.1	Visual Tracking Results . . . . .	64
6.2	Activity Recognition Results . . . . .	70
<b>7</b>	<b>Conclusion</b>	<b>77</b>
7.1	Contributions . . . . .	77
7.2	Future Work . . . . .	79
	<b>Bibliography</b>	<b>83</b>
	<b>Appendices</b>	<b>92</b>
<b>A</b>	<b>Spheroid Mapping</b>	<b>92</b>
A.1	Perspective Projection . . . . .	92
A.2	Equirectangular Projection . . . . .	92
<b>B</b>	<b>Sample Detection and Tracking Frames</b>	<b>95</b>

# List of Tables

6.1	Tracking results for projection camera model . . . . .	68
6.2	Activity detection validation results. Event sampling frequency: every 30 frames. .	70
6.3	Activity detection validation results. Event sampling frequency: every 10 frames. .	71

# List of Figures

3.1	Major processing components of our system . . . . .	17
4.1	Finding vertical vanishing point $VZ$ . . . . .	24
4.2	Determining projection scale . . . . .	24
4.3	Output from six monoscopic sensors . . . . .	25
4.4	Stitched panoramic output from Ladybug . . . . .	25
4.5	Color codebook . . . . .	28
4.6	Finding extremes in VPP histogram . . . . .	30
4.7	Vanishing point projection (VPP) histogram . . . . .	30
4.8	Foreground segmentation results . . . . .	32
4.9	Gaussian kernels for weighted histogram . . . . .	38
4.10	Floor plan . . . . .	40
4.11	Hue-saturation 2D histograms . . . . .	41
4.12	Z-buffer . . . . .	42
4.13	Distance weight map . . . . .	43

4.14	Mean-shift illustration . . . . .	46
5.1	Major processing components of activity detection algorithm . . . . .	50
5.2	Clustering validity. <i>Left:</i> isolation $I_i$ and compactness $I_c$ <i>Right:</i> Combined validity index $I$ . . . . .	54
5.3	Function $\sigma_1(t)$ . . . . .	56
5.4	Function $\sigma_2(t)$ . . . . .	56
5.5	Two-dimensional profile of distance function $D_e^2$ . . . . .	56
5.6	Profile of distance function $D_e^2$ . . . . .	58
6.1	Head candidates from test frames . . . . .	66
6.2	Head detection algorithm performance evaluation . . . . .	67
6.3	Incorrect head detection from test frames . . . . .	67
6.4	Customer paths marked on the floor map . . . . .	69
6.5	Select frames showing the detection of shopping groups . . . . .	69
6.6	Swarming events in space-time . . . . .	72
6.7	Shopping groups graph . . . . .	73
6.8	Group detection accuracy as a function of path duration . . . . .	74
6.9	Shopper group detection illustration . . . . .	75
A.1	Finding spheroid range in horizontal scanlines . . . . .	94
A.2	Finding spheroid limits in horizontal scanlines . . . . .	94
B.1	Head candidates from test frames . . . . .	96

B.2	Sample tracking frames from CAVIAR dataset . . . . .	97
B.3	Sample tracking frames from OTCBVS dataset . . . . .	98
B.4	Sample tracking frames from apparel retail store dataset . . . . .	99
B.5	Sample tracking frames from electronics store dataset . . . . .	100

# Motivation and Problem Statement

## 1.1 Motivation

There is an increasing amount of research interest in the area of video analytics and video mining, in application to automated scene surveillance and subsequent behavior analysis. This, in part, is motivated by the increased awareness of potential security applications, but also by a growing interest toward such research in the industrial sector, in particular, by marketing departments of retail companies. Another moving force is the increased accessibility and speed of current computer hardware and a growing base of public domain pattern recognition, vision, and neural network processing software made available by the international research community.

Visual surveillance is entering a new more intelligent phase. Surveillance systems are no longer simply recording the observed visual information, but are attempting to extract low-level motion information and, more recently, to analyze complex behaviors in the scene. The novelty of this work is that it brings the marketing applications perspective to the computer vision sphere and implements one such application — detection and tracking of shopper groups based on the paths traveled by the customers. Of particular interest for marketing intelligence are moving customers, the products or fixtures they interact with, as well as how the customers interact with each other. Detecting shopper

groups can provide several useful statistics to be subsequently used by the marketing research community and implemented in practice by the retailers. This is particularly so as marketing intelligence is switching to a new paradigm of managing customer experience, where such indicators as store traffic, shopping path, aisle penetration, dwell time, product interaction and conversion rate become of essence. Statistics extracted from the tracking data are now becoming used to highlight social aspects of shopping habits. This is the area where activity analysis of shopper groups can contribute the most.

As marketing researchers in academia and industry are seeking tools to aid their decision making, their interest is increasingly involved with computer vision and in particular human tracking systems. Unlike other types of sensors, vision presents an ability to observe customer experience without separating it from the environment and without the intrusiveness of other, more active observation methods using other sensor modalities. By tracking the path traveled by the customer inside the store, important pieces of information, such as customer dwell time, aisle penetration and product interaction statistics can be collected [4, 35]. In our work we have concentrated on extracting from video, one of the most important customer statistics: information about the *shopper groups*.

Humans engage in various types of activities that can be analyzed for different purposes. For example, in the field of marketing, customer activities are analyzed to improve quality of service or increase sales. Marketing and retail researchers analyze customer behavior in videos by manual coding. However, manual identification of individuals and their activities can be time and resource consuming. Visual observation by human coders, physically present in the store, has proven itself to be an extremely intrusive methodology, which often interferes with the daily operation of businesses. Automating the recognition of human behavior by analyzing video material becomes an important task as it helps to overcome these limitations. Because many retailers are already gathering video data, methods of computer vision and video mining can be applied to the problem.

[4]

The task of automated activity recognition for marketing purposes can include various subtasks such as customer-product interaction, unusual event detection, customer group behavior analysis and others. Activity recognition is an ordinary task for the human observer. To have an automated system discriminate between various types of activities is, in large, an attempt to simulate the results of high level visual information processing in a human brain by means of computer vision. It is a combination of the state-of-art methods and the knowledge about how humans process this type of information, such as creating higher level abstractions, from the visual observations. Apart from productivity gains and resource savings, this is why automated customer activity analysis is an interesting and challenging problem.

## 1.2 Goals

One of the most important goals in the video analytics domain is to extract semantic information about a scene. Humans and their interactions are some of the most significant components in the scene, actively affecting the environment and the actors surrounding them. Studying human activities in retail contexts can be considered an attempt to formalize these interactions as applied to marketing, to be further used in developing formal indicators of the retail store performance. Despite a large array of research activities in short term tracking there have been few attempts to produce a consistent track throughout time spans substantial enough to attempt human behavior interpretation. Typically, in machine vision publications of the past, visual tracking is performed on sample video datasets of not more than several minutes in duration. In some scenarios this length is enough to detect simple actions involving a single human (sitting down or picking up an object) or even short interactions involving multiple humans (a handshake or meet-then-split sequence). Alas, the information contained in such short video sequences is usually insufficient to derive any reliable conclusions about high level group behaviors. The reason for this is because such behaviors have multiple manifestations that may be separated in time. For example imagine a scenario where a group of people interact as they enter the store, later people in the group split to do their individual

shopping, but at the end the group re-assembles to proceed to the checkout.

The ultimate goal of this research is to achieve automated information collection from video sequences observed in retail stores. This translates into extracting semantic, high-level information about human behaviors from long tracking sequences with the aim to pin-point specific types of activities that are interesting from the marketing research viewpoint. One particular type of information of interest is the location of customers at each instant of time and their interactions with products, peers, and employees.

The aim of this study in automated human tracking and activity recognition is the marketing driven analysis of retail store environments. The purpose of building a computer vision system for customer activity recognition then becomes twofold: to develop a set of methods for automated recognition of human activities in crowded environments and to recognize specific patterns of customer movements in order to facilitate marketing analysis

To identify the customers who are shopping as a group we have designed a distance metric, measured on the traveled trajectories. This metric, which incorporates space and time deviations between two paths is then used in a clustering system to label shopper groups in the input video. We further perform histogram analysis to detect store employees and motion dynamics analysis to detect dwelling customers.

To summarize, in this work we will address the following problems:

- Detecting and tracking human bodies in complex environments using a single stationary camera
- Creating formalized definitions of grouping events and activities
- Developing methods to detect grouping events and activities in tracking sequences
- Collecting quantitative statistics about grouping activities

### 1.3 Contributions

Visual tracking is a difficult problem for computer vision because human body is a highly articulated, non-rigid and often a self-occluding object. Each human body is unique due to the differences in shape, clothing color and texture, as well as gait and other dynamically changing characteristics. The case of a retail store, which is a crowded environment where multiple, partial and full occlusions may occur, makes this task even more challenging. Moreover, computer science has little understanding of how to formally define human activity or which manifestations of particular activities can be considered indicative. The problem then can be stated as follows: as manual analysis of customer activities in the retail environment becomes time- and resource consuming, it is necessary to develop an automated computer vision system to detect patterns of customer activities as well as to record time, place, duration and other relevant characteristics of their activities.

In its foundation, the system presented here segments foreground regions in each frame by using our novel, adaptive background model. Because each foreground region may contain multiple people, we further hypothesize the number of human bodies within each such region by using the head-candidate selection algorithm. The head is chosen as the most distinguishable, visible and pronounced part of the human body, especially when the observation is made with a multitude of objects occluding the lower parts of the body. Here we present a new method for head candidate detection, which uses the knowledge of camera model to achieve more discriminating estimates of head locations. For human tracking, we construct a Bayesian inference model operating on a Markov chain, where the states of the chain represent the parameters of the tracked customers. The inference is based on the a priori knowledge of the human body parameters, the store layout and geometry, observations of the body appearances at each frame and the temporal link to the previous state. To make the inference computationally efficient, we introduce a number of new reversible transformations with respect to the system state and apply them as part of the Monte-Carlo stochastic optimization approach.

The activity recognition work in this thesis was inspired by the studies of swarming behaviors in living organisms and their consequent implementations for modeling systems with complex intra-connectivities. We present a generalized extensible framework for automated recognition of swarming activities in video sequences. As one instance of this framework we present a method to detect *shopper groups* in tracked video sequences of retail stores. Our system uses tracked coordinates of customers to detect a series of *swarming events*, *i.e.*, the scenarios where several people behave with intrinsic group characteristics. There can be multiple events for a single group as people who enter the store as a group may repeatedly split apart and reconvene. Therefore, swarming events serve as short-term manifestations of a longer-term group behavior, which we call a *swarming activity*. In Chapter 5 we describe how two stages of agglomerative clustering can be used to detect shopper groups. At the first stage, to detect swarming events we employ a deterministic clustering of inter-actor discrepancies in location, orientation and dwelling status. The number of clusters and termination criteria is determined automatically by optimizing the clustering validity indexes. At the second stage our system integrates large quantities of swarming events to obtain a shorter list of more meaningful clusters, corresponding to shopper groups. Considering several clustering methodologies, we found that the fuzzy agglomerative techniques, proposed in [25], achieve the best segmentation and are robust to noise in the form of outliers.

The results obtained in this work demonstrate the ability of our method to detect such activities in congested surveillance videos. In particular, in three hours of indoor retail store videos, our method has correctly identified around 80% of valid “shopper-groups” with a  $< 2\%$  level of false positives, validated against human coded ground truth.

The structure of this thesis is as follows. Chapter 1 defines the scope, goals and key contributions of this thesis. In Chapter 2, we provide an overview of the existing research in human tracking and activity recognition with focus on retail applications. Chapter 3 describes the experimental setup, hardware, and computational modules involved in the system. Chapter 4 describes the algorithms and underlying model for human tracking and Chapter 5 presents the conceptual groundwork and a

method for detecting group activities. We present quantitative results for both tracking and activity recognition, and highlight several interesting findings in Chapter 6. We conclude by summarizing main achievements of this thesis, analyzing imminent shortcomings and highlighting some future work in Chapter 7. A comprehensive bibliography and index are included at the end of this thesis for the convenience of the reader. Appendices include in-depth formal descriptions of algorithms and illustrations too large to appear in the main part of the thesis.

## Related Work

The area of my concentration is human tracking in complex environments and human activity recognition. In this chapter we review existing approaches in tracking people in videos from stationary cameras and the methods for detecting activities of groups of people in these video sequences.

### 2.1 Human Tracking

The problem of automatic, real-time human detection and tracking has received a lot of attention in the machine vision community and is now identified as one of the key issues in numerous applications ranging from surveillance, autonomous navigation, and robotic systems [13, 51] through crowd behavior modeling and human activity recognition [29, 53, 56].

Significant progress has been made in detection and tracking of people. The majority of the studies address tracking of isolated people in well controlled environments (typically the office space), however, there is increasing effort in tracking people specifically in *crowded environments* [11, 52, 29, 28, 31, 23]. It is worth noting that many works assume the luxury of multiple well-calibrated cameras or stereo vision, which are to a large extent not yet present in most retail establishments and/or do not have the desired overlapping fields of view. In contrast, cheap low-resolution digital monocular color cameras are becoming more and more readily available as well

as the hardware for capturing compressed real-time streams provided by these cameras.

In videos taken with a stationary camera, background subtraction is a primary technique used to segment out foreground pixels. Statistical background modeling based on color distortion has been presented in [34], but a single mean for each pixel is unlikely to account for the noisiness of the background in the changing environment of the store. We have also given consideration to the methods that use a mixture of Gaussians (MoG) to model each pixel [63]. These methods are superior to the single-modality Gaussian based approaches, yet they operate based on the assumption of a fixed number of modalities which fail to robustly and comprehensively accommodate the noise and artifacts created by video compression algorithms. We have developed an adaptive background model based on the dynamic codebook approach which compensates for these problems. We have built upon the methods utilizing a variable number of modalities [43] to have the model adapt to changing background conditions while performing background subtraction. Various color spaces have been investigated for foreground segmentation. The initial success of using HSV, LAB or YUV spaces [66, 17] to remove the luminance component, therefore reducing lighting artifacts, was not confirmed by this work. We found that in the tight space of the store environment the interplay of colors and cast shadows contributes as much to the changes in hue as to the changes in the brightness itself.

Most visual tracking systems use some model of human body shape and appearance. Modelling a body at a joint level, by estimating corresponding angles between limbs and imposing constraints from human physiology has been done by several researchers [69, 71]. For the video dataset in this study, given the quality and the resolution, *i.e.* the number of pixels typically making up a single body, as well as the complexity of the scenes (with all external and self occlusions) we found this approach impractical. Single shape primitives (*e.g.* cylinders, ellipsoids or cones) were shown to somewhat address these shortcomings by simplifying the model and are the closest to our work. Appearance can be modeled based on color or texture features for the body as a whole [62, 45] or for separate parts [60]. There was only a limited focus on modeling dynamic changes in the

appearance and shape of the person and updating the state of changing environment. Here we show how the models can be extended by dynamically updating color representations. Overall, robust initialization and adaptation of shape and appearance models remains an open problem in the machine vision community.

To create the initial estimates for any tracking algorithm, some form of head position estimation has been used in related studies. In [29, 71] the vertical projection histogram was computed to reliably establish the location of head-candidates. Although the aforementioned approach shows promising results with the horizontally looking camera (*i.e.* the optical axis parallel to the ground plane), in this paper we make the argument that such a techniques will be prone to significant distortion in the case of ceiling mounted camera if the camera pose is not accounted for. As a result we are using the projection histogram that accounts for the camera and 3D scene parameters.

Once body candidates have been established, one of the primary goals of the tracking system becomes finding correspondences in time for each such body. The problem is made more complex by the fact that the number of the people in the scene at any time can either increase (when new people are entering the scene) or decrease (when people are exiting the scene). Temporary appearance and disappearance can also be produced by people fully or partially occluded by rigid objects in the scene as well as by other moving humans. Even if the number of tracked objects remains constants this could be due to the same number of people entering and leaving simultaneously.

To deal with these complexities we developed a generative tracking framework based on the newly found implementations for Markov Chain probability density sampling. In recent research, random sampling was shown not only to successfully overcome singularities in articulated motion [21, 55], but particle filtering approach applied to human tracking has also demonstrated potential in resolving ambiguities while dealing with crowded environments [38, 64, 42]. Working within the Bayesian framework it has been shown that particle filters can efficiently infer both the number of objects and their parameters. Another advantage is that in dealing with probability distributions of mostly unknown nature, particle filters do not make Gaussian assumptions, unlike Kalman filters

[40, 61]. The canonical *conditional density propagation* [37] incorporates the complexity of the scene but is computationally costly and prone to getting stuck in local minima. In these papers each system state is a multi-body object, which is allowed to change in size (as people enter and exit). Additionally, the authors introduce some *informed* particle transitions that can significantly reduce the computational complexity and help overcome local optima by utilizing knowledge of the underlying process.

## 2.2 Activity Recognition

Tracking followed by the analysis of customer behavior in stores is becoming an increasingly active subject in computer vision publications [29, 28, 31, 71]. In the computer vision community, detection of shopper groups in checkout lines has been attempted by Haritaoglu [28]. For grouping, authors use inter-body distances as well as such specific environmental clues as the cashier’s activities to determine the start and end of shopping transactions. Several approaches exist based on Discrete Fourier Transform and Dynamic Time Warping exist for comparing time series and have been used to measure similarity between motion trajectories. Most recently a method based on the *longest common subsequence* for comparing trajectories has been implemented by Buzan et al. in [5]. The authors perform trajectory-based clustering and retrieval, using a modified version of edit distance, called the *longest common subsequence*. Similarities are computed between projections of trajectories on coordinate axes. Trajectories are grouped based on this distance, using an agglomerative clustering algorithm. The specificity of our task is that it requires a relative time component — as opposed to comparing just the shapes of the trajectories, yet it must not account for time warps — as is done in speech recognition.

Rosario et al. [56, 59] have developed a framework based on coupled hidden Markov models (HMMs) to recognize pedestrian interactions in visual surveillance videos. In this work, simple behaviors such as walking or changing direction are grouped into higher level interaction scenarios,

for instance "approach, meet and walk together." This publication is, to our knowledge, the closest to our work, with the key difference being that the time span that we consider to find shopper groups is in the order of tens of minutes (*i.e.*, it is much longer). Additionally, we formulate our model as a recognition of the fittest swarming behavior, which gives us a freedom to not establish explicit ties between events present in Markov modeling.

Another approach is to consider single- or multi-threaded events [33] with consequent events satisfying a predefined decision tree. Here activity is considered to be composed of action threads, each thread performed by a single actor. A thread is modeled by a stochastic finite automaton of event states, which are derived from the trajectory and shape of moving blobs via Bayesian inference. However this method is more suitable to address single actor behaviors with a well defined time-sequential structure. The interactions between two or more actors are bracketed into a limited number of mutually exclusive behavior scenarios, which usually span less than one minute in time. While this approach has shown some good results on further dissecting human interactions into shorter pieces, its nature is not suited to perform well with the long-term behavior dynamics.

Some attention has been given to person-to-person interactions in the context of security applications. In [2] the authors detect hand gestures by using a context-free grammar parsing mechanism. The grammar and the parser provide longer range temporal constraints, disambiguate uncertain low level detections, and allow the inclusion of a priori knowledge about the structure of temporal events in a given domain. Again the attention here is given to the events happening sequentially in time. We will show that such an approach fails when applied to long-term group detection.

Several attempts were made at group activity recognition in the context of intelligent rooms. In [70] a two-layer HMM framework is presented to handle a bi-modal input from audio and video. The activities of each actor in the meeting, such as speaking or taking notes are hierarchically combined into activities like "presentation" or "white board" that depict the character of group behavior. It is worth noting that this study and the research in [56, 68] are some of the few works that introduce a two-level event/activity hierarchy. As with all HMM methods, a significant data

set and human resources are required to train the transition probabilities in Markov models in a supervised fashion.

An unsupervised statistical solution for detecting play/break activities in soccer games using hierarchical HMMs is presented in [68]. A Monte Carlo optimization is performed using a set of prior prototypes derived from the soccer playing rules, with dominant color ratio and motion intensity used as low level features. This is the case when the application domain knowledge is built into the system, therefore removing the training stage altogether.

Most recently, an unsupervised bottom-up activity recognition approach was presented in [67]. Their method was based on low-level descriptors, in particular location-specific detection of significant pixel changes. The metric called *Pixel Change History (PCH)* copes well with noise and clutter, present in complex scenes, where conventional tracking approaches would produce highly fragmented "tracklets". For our purposes, however, it is be suitable, since the specific number of actors, is of primary interest, therefore an explicit tracking of interacting objects is desired.

The works reviewed in this chapter only present a fraction of the whole body of research in human motion capture and activity recognition. We purposefully did not elaborate on human action detection, as it is quite different from activity recognition, in that it looks at the detailed actions performed by a single actor, such as running or sitting down. Activities, as defined in [53]: "...are larger scale events that typically depend on the context of the environment, objects, or interacting humans". Throughout this thesis, whenever need be, we will refer to such "actions" as *individual activities*, but the emphasis of this work remains on *group activities*.

Our novel paradigm for modelling human activities as multiple instances of swarming behavior is presented in Section 5. Swarming behavior in biology, is an activity of a decentralized group of multiple agents, such as schools of fish and flocks of birds. It was observed that the agents act based on a set of simple spatial rules to interact with their immediately neighboring agents. This observation has lead to an idea that this type of process can be used to model other seemingly complex systems, consisting of multiple simpler parts and has given rise to a whole new research

area of *swarm intelligence* in artificial intelligence [39]. Several studies in this topic deserve a special mention.

One of the early implementations of swarming intelligence was presented by Reynolds [58] for animating the flock of birds in computer graphics. Each bird/actor there following a set of simple rules, such as steering toward the center of the flock and maintaining distance with other flock members. Intelligent swarms and in particular a technique called *Particle Swarm Optimization or PSO* are currently applied for simulation of complex processes involving multiple locally-interacting agents [10]. In this approach the problem is modeled by particles in multidimensional space. These particles are flying through hyperspace  $R^n$  and have two essential reasoning capabilities: their memory of their own best position and knowledge of the swarm's best, *i.e.* the particle with the smallest objective value. Members of a swarm communicate good positions to each other and adjust their own position and velocity based on these good positions. Our method is different from PSO in that the target value is already given to us by the tracker. From the generative approach we transition to a recognition problem with the goal being: find a set of particles that, given their tracking data, best behave as a group.

Despite many successful applications of swarm intelligence models, to our knowledge there have yet been no attempts at using this paradigm to solve the problem of group activities in human behavior.

## System Overview

### 3.1 Experimental Setup

In this work two architectures for testing the accuracy of human tracking and precision of shopper group detection were developed. These designs are based on two types of cameras located in retail stores: a perspective projection digital security camera and a panoramic camera.

The perspective projection camera provided the video data for developing and testing of the tracking algorithm. The data were collected from several camera positions and at various times of the day. This included a number of sequences from security cameras placed in an electronics retail store as well as a number of indoor and outdoor publicly available sequences from CAVIAR [6] and OTCBVS [19] datasets. The resolution of the cameras was 320x240 color pixels.

For the panoramic camera, data were collected over a period of several weeks and recorded to a portable storage media. We used a panoramic camera system from PointGrey [36], consisting of six CCD cameras with 1024x768 resolution. Six images are post-processed to give a single radial panoramic image of 2048x512 in unprojected-map coordinates (see Section 4.1.1 for details). Overall we analyzed 3 hours of panoramic video and base our evaluation of activity recognition method on these experiments.

Sample frames with tracking results for all datasets described above are presented in Appendix B.

Some of the conditions that are similar for both testbeds are described below. In all sequences, cameras were mounted on the ceiling or the structures located close to the ceiling. Lighting conditions varied from natural daylight conditions with rapidly changing cloud cover, to a typical retail store lighting — a combination of incandescent and fluorescent light sources. Data analysis was conducted offline. The most time consuming stage is tracking, with speeds ranging from 0.05 to 5 seconds per frame, depending on the complexity (number of tracked bodies) of the scene. The resolution of the input scene image also influences the speed of tracking at the background segmentation stage and, implicitly, at the body-model construction phase, since the higher resolution camera covered more square footage, therefore including more bodies to track.

During the tracking stage the number of errors of type I (false negatives) and type II (false positives) tend to accumulate, especially when tracking is performed on hours of video data. While testing the accuracy of our activity recognition methods it was important to prevent the tracking errors from influencing the activity analysis. Therefore, to validate our activity detection method a ground truth was created by manually connecting broken tracks, removing the redundant ones and adding the ones missed by the system. The correction process was performed by several domain experts. This data was recorded in the format identical to the tracking performed completely automatically. The reader is referred to Chapter 4 for a detailed description of the tracking data format.

## 3.2 Overview of the Computational Framework

The methods presented in this paper are aimed at developing a tool for retailers to analyze patterns of behaviors by shoppers in stores and using the results of this analysis to make various marketing decisions. The system consists of processing layers ranging from low-level image processing operations, to tracking the positions of individuals in the store videos, to the higher level analysis of their movements and activities in the store. Figure 3.1 shows the various layers of the

system and the individual modules that make up these layers:

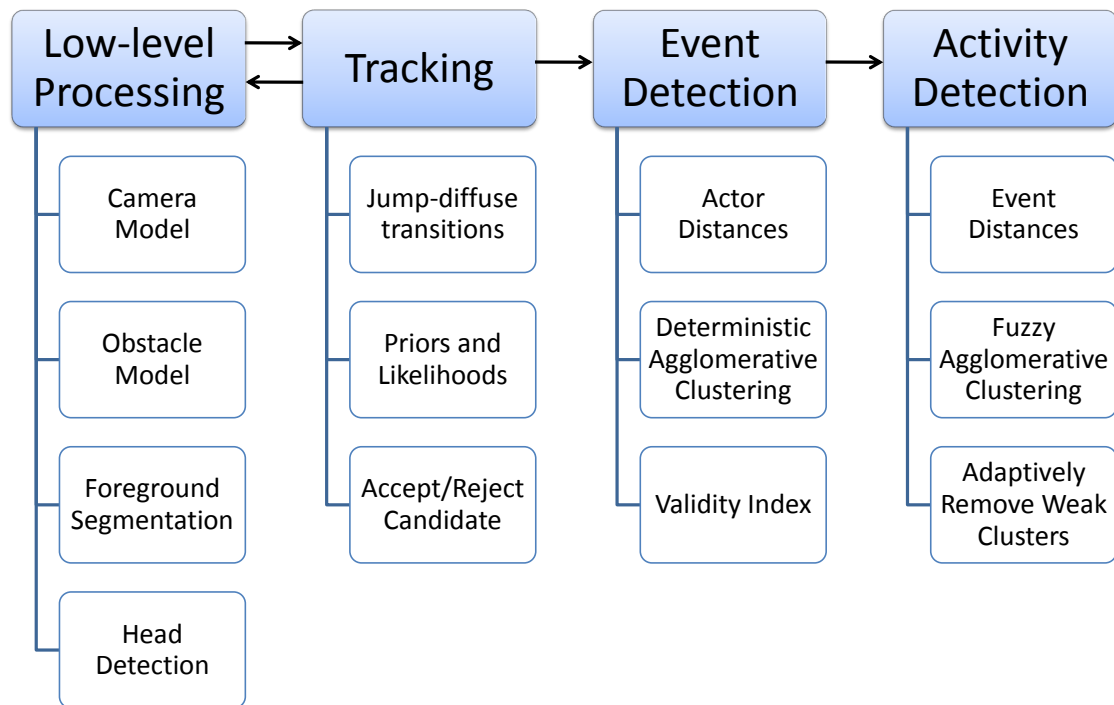


Figure 3.1: Major processing components of our system

The first layer consists of the pre-processing steps required by the tracker. Background subtraction is accomplished by learning a background model which incorporates knowledge about the changing lighting conditions of the scene. The output from the background subtraction is the binary foreground map as well as an array of foreground blobs, each represented as a 2D contour. The process of background learning and the algorithm for separating it from the foreground is presented in Section 4.1.2 of Chapter 4. Construction of a camera model at this step provides the next stage of the system with the locations of the vanishing points as well as the scale factor. In addition,

knowledge about the camera model and the approximate shape of the shoppers is also used to map the position of the tracked person on a floor map of the store and to obtain rough estimates of the positions of their heads in order to resolve multiple shoppers possibly occluding each other. In addition, the model of the static obstacles, adds information to deal with the occlusions by the store fixtures. Here, we also run an algorithm to detect head candidates for each human body. The details of this layer are described in Section 4.1 of Chapter 4.

The next layer implements the tracker. The goal of the tracker is twofold. First it attempts to model each human body in the static scene, so that it best describes the observation. It relies on the information about the foreground (*i.e.* the moving regions of the image) as well as the head candidates and obstacles, created in the previous step. The second goal of the tracker is to introduce temporal continuity, by building correspondences between the bodies in the current frame and their counterparts in the previous frame. We show how both of these goals can be achieved by modelling the system as a Markov Chain, where each state corresponds to one moment in time. A state is a combination of the individual parameters of each tracked body, such as position, dimensions and color histogram. Our tracker then uses a probabilistic sampling method to modify the current state of the Markov Chain. The fitness of the new state is assessed based on how it matches the new observation and how well it ties to the previous state. The fittest candidate becomes the new state of the chain. This way, the tracker maintains the identity of people in the store, dynamically assigns identities when new shoppers enter the scene and removes the identities of the customers leaving the scene. The results of the tracking are also channelled back to the background subtraction layer to help exclude human bodies from the background model. The details of this layer are also described in Chapter 4.

The third layer uses the tracking results from the first two layers to detect short-time manifestations of a grouping behavior. As part of this analysis, the paths of the shoppers on the store floor are extracted. This information is then used to determine if some of the shoppers exhibit characteristics of group behavior to identify *swarming events*, based on the pattern of their coordinated movements

in the store. By examining the co-location, co-orientation and co-dwelling of the people in a static frame, we are using agglomerative clustering to detect the customers that exhibit the most of the above group characteristics. The details of this layer are described in Chapter 5.

Finally, our activity detection method uses distances between the events detected in the previous step to form groups of events, indicative of the behavior of shoppers (*i.e. swarming activities*). This is done as part of a probabilistic clustering framework. Our algorithm assigns one or more events to one or more activities with a certain probability. For instance, if enough events relate strongly to a single cluster, then this supports the hypothesis in favor of group behavior. For this purpose we define an event proximity measure which can be computed for each pair of events. Based on this measure, the clustering is performed, and the effects of outliers are alleviated by the use of robust estimators. We describe our method and related data analyses in Chapter 5.

## Detection and Tracking

Detecting and tracking human body position is a necessary intermediate step in the process of developing automated customer activity recognition system. Tracking in crowded environments is a particularly complex case of human tracking primarily because of the high level of entropy generated by multiple body occlusions and also because the complexity of the model rises exponentially with an increasing number of people. For this task we developed a Monte Carlo based generative model within the Bayesian inference framework, which we use in a time-efficient manner to speculate about future system configurations based on two pieces of information: current model state and current observation. The novel algorithm for adaptive background subtraction we developed specifically addresses the problems of low-quality, highly-compressed videos, where the backgrounds can be better represented by a dynamically growing codeword of pixels. Initialization of the tracking model is done by detecting human head candidates using vanishing point projection histograms. Individual human body position of each customer in our system is estimated by our mean-shift tracker for ellipses with a weighted anisotropic Gaussian kernel, that tracks based both on the color histogram difference as well as on the background/foreground mask consistency [49].

## 4.1 Detection

To identify the position of each subject in the scene we first employ a range of detection techniques on a static video frame. These techniques do not consider any temporal dependencies and serve merely as an initialization step for the tracker. The accuracy of this step, however, is of utmost importance, since any initialization errors tend to quickly propagate throughout the later stages of processing.

### 4.1.1 Camera Modelling

While building realistic human body models during the higher-level tracking stages of the system, one cannot rely only on the projected image for several reasons. Primarily, the reason is the ambiguity that arises from the possibility of projecting several distinct world objects into the same location in the image. Additionally, we would like to approximate each human body with the 3D shape (an ellipsoid) and thus the image coordinates are not suitable. This is why it is important to work in 3D scene space. The blob-level observations only provide us with the location in the two dimensional image plane and therefore there is a need to create mechanisms of converting the location of the objects to the three dimensional world. To accomplish this, intrinsic and extrinsic camera parameters must be estimated in order to relate the image space to the scene space. Throughout the paper we will use the following notation. Let  $\{X, Y, Z\}$  be a system of world coordinates and  $O_{cam} = \{X_c, Y_c, Z_c\}$  camera coordinates where  $X_c = 0$ ,  $Y_c = 0$  and  $Z_c$  is the elevation of the camera in *cm*. Let  $O_{sph} = \{X_s, Y_s, Z_s\}$  denote the center of the spheroidal body model and  $O = \{X_o = 0, Y_o = 0, Z_o = 0\}$  be the origin of the world coordinate system.

#### 4.1.1.1 Perspective Projection Camera

A monoscopic camera can be modelled by the perspective projection or a pinhole camera model. The model consists of the projection matrix  $P$  which converts the coordinates of a point  $\vec{X}$  in the

real world to the homogeneous coordinates  $\vec{x}$  in the image plane  $\vec{x} = P\vec{X}$ .  $P$  is a  $3 \times 4$  matrix as presented in Equation (4.1):

$$P = C[R|T]$$

$$C = \begin{pmatrix} fk_x & 0 & x_0 \\ 0 & -fk_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.1)$$

The matrix  $C$  contains intrinsic parameters of the camera:  $f$  — the focal length,  $k_x$  and  $k_y$  — dimensions of the image pixel, and  $x_0, y_0$  — the principal point (intersection of image plane with the optical axis in image coordinates).  $R$  is the camera rotation matrix and  $T$  is the camera translation vector. For the implementation details and advanced properties of pinhole camera model see [24].

The parameters of a perspective projection camera model are usually found by placing a recognizable object (such as the checker board), with well known 3D coordinates, into the scene and then finding point-to-point correspondences between the image and the 3D points (*i.e.* the calibration process). This method is, however, too intrusive to be applied in the operating store environment. Our solution is to use the knowledge about the geometric properties of the in-store fixtures for camera calibration. Many man-made environments contain rectilinear structures in the scene. We have used algorithms that extract vanishing points from the images of parallel lines in such rectilinear scene structures [15, 9]. The projection of parallel lines in the image converges in the so-called vanishing point. We are interested in finding the vertical vanishing point  $\mathbf{V}_Z$  as the center of intersection of the lines which point in the vertical direction. Two lines are sufficient to find  $\mathbf{V}_Z$ , but in a noisy environment it is beneficial to consider more lines to achieve higher accuracy in the location of the vertical vanishing point  $\mathbf{V}_Z$ . This is computed as the centroid of the intersection points of the images of all the 3D vertical lines. In our application environment there is an abundance of

man-made rectilinear structures with vertical lines that can be used for that purpose (aisles, boxes, markings on the floor, doors and windows).

In the calibration phase, a number of lines, parallel in space, are designated manually with the help of a simple point and click interface (Figure 4.1). Each line is represented as two endpoints  $\mathbf{e}_1 = [x_1, y_1]$  and  $\mathbf{e}_2 = [x_2, y_2]$

Prior to computing the vanishing point all line endpoints are converted into the homogeneous coordinates with the origin in the center of the image  $[\frac{w}{2}, \frac{h}{2}]$ , where  $w$  and  $h$  are the width and height of the image in pixels, respectively. The scaling factor is set to the average of image half-width and half-height  $(w + h)/4$  for better numerical conditioning (*i.e.* to prevent floating point precision loss; see [15]).

$$\begin{aligned}\mathbf{e}'_1 &= [x_1 \times \frac{w}{2}, y_1 \times \frac{w}{2}, (w + h)/2] \\ \mathbf{e}'_2 &= [x_2 \times \frac{w}{2}, y_2 \times \frac{w}{2}, (w + h)/2]\end{aligned}$$

Then in homogeneous coordinates each line can be computed as a cross-product of its endpoints  $l = \mathbf{e}'_1 \times \mathbf{e}'_2$ .

The  $3 \times 3$  “second moment” matrix  $M$  is built from an array of lines  $\mathbf{l}_i$  and  $\mathbf{V}_Z$  is computed from the solution of  $M$  by singular value decomposition as the eigenvector that corresponds to the smallest eigenvalue [12].

The conversion from 3D world coordinates  $\vec{X} = \{X, Y, Z, 1\}$  to 2D homogeneous image coordinates  $\vec{x} = \{x, y, 1\}$  is done by left multiplying by  $3 \times 4$  projection matrix:  $\vec{x} = P \cdot \vec{X}$ . Consequently homogeneous coordinates are converted to image coordinates. The conversion from image coordinates, given the  $\hat{Z}$  coordinate in the world is done by solving the system  $[sx, sy, s] = P \cdot [X, Y, \hat{Z}, 1]$  using singular value decomposition.



Figure 4.1: Vanishing point  $V_Z$  can be found by manually marking two or more vertical straight lines

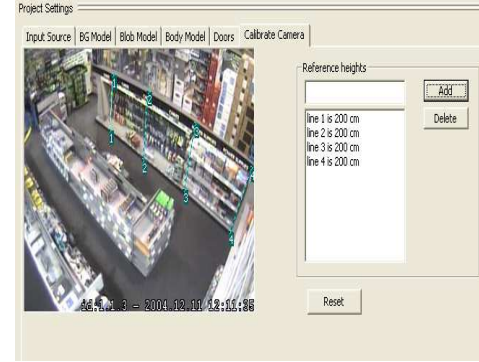


Figure 4.2: Marking the objects of known height to determine the scale

#### 4.1.1.2 Stitched Panoramic Camera

To have a surveillance system fully cover the scene in the retail store, several applications with multiple perspective projection cameras have been proposed [8, 54, 41]. This setup requires tracking across different cameras and is associated with the significant increase in system complexity, both in hardware and in software. Instead, we used a more easily deployable tracking system, which consisted of a single LadyBug panoramic camera [36].

The output from such a camera, used in our tracking experiments is a 1024 by 256 panoramic image (see Figure 4.4), stitched from the outputs of 6 monoscopic perspective cameras (see Figure 4.3). The device is located in the middle of the store, at an elevation of 3 meters and consists of one camera looking straight down and 5 additional cameras located around the horizontal circumference at even intervals. The store area is approximately  $30 \times 30$  meters, with the most accurate tracking achieved within a 20 meter radius and gradually degrading towards the periphery, because the occlusions become more pronounced and the resolution per tracked body is lower.

The panoramic image in Figure 4.4 is presented in the form of an *unprojected map* also referred to as *equirectangular projection*. In an unprojected map the horizontal coordinate is the longitude and the vertical coordinate is the latitude. In our case, for each pixel  $p_{x,y}$ ,  $lat(p) = x * 360^\circ / w$  and



Figure 4.3: Output from six monoscopic sensors within the Ladybug (*bottom-right*: output from downward facing camera [36])



Figure 4.4: Stitched panoramic output from Ladybug

$lon(p) = 90^\circ + y * 90^\circ / h$ , where  $w$  and  $h$  are image width and height in pixels. This makes  $x$  range from  $0^\circ$  to  $360^\circ$  and  $y$  range from  $90^\circ$  to  $180^\circ$ , effectively covering the southern hemisphere with the south pole located directly below the camera.

The conversion from image to world Euclidean coordinates is quite straightforward in this model, with the south pole point on the floor (the center pixel in image 6 in Figure 4.3) designated as the origin  $[x = 0, y = 0, z = 0]$ . Note that the vanishing point histogram now reduces to its special case of vertical projection histogram, since the vertical projection lines leading to  $V_z$  are now parallel. The conversion from image coordinates, given the  $\hat{Z}$  coordinate in the world is done by first converting to spherical coordinates  $\{x, y\} \rightarrow \{\phi, \theta\}$  and then finding the world coordinates from similar triangle geometry

$$\begin{cases} X = \cos(\theta) \tan(\pi - \phi)(Z_c - \hat{Z}) \\ Y = \sin(\theta) \tan(\pi - \phi)(Z_c - \hat{Z}) \\ Z = \hat{Z} \end{cases}$$

For each of the two camera models outlined above the tracked coordinates are eventually converted into floor coordinate, *i.e.*  $X_f, Y_f$  in the floor plane with  $Z = 0$  in the world coordinates. This way each track can be viewed as a poly-line or a spline spreading across the floor map of the store.

#### 4.1.2 Background Modeling and Subtraction

The ability to extract moving regions in the video data is crucial in visual tracking of humans. The process of the background subtraction is a class of methods for separating the moving pixels in the image, from the static background. The basic assumption underlying these approaches is that the visual appearance of the static parts in the scene (as appearing in the image) remains constant, while the position of moving objects changes from frame to frame. Thus, if observed for some time, one can accumulate the image of the stationary parts of the scene and further subtract it from the

future video data.

Video sequences from the in-store surveillance cameras are frequently compressed with MPEG-like algorithms, which usually create a periodic noise on the level of a single pixel. Changes in lighting and dynamic changes in the scene layout, such as new fixtures being introduced, all add to the complexity of background separation task. We have incorporated a multi-modal statistical background model based on the codebook approach based on the research by Kim et al. [43] with a number of improvements, to achieve a stable and sustainable foreground segmentation.

In order to capture multi-distributed light variation on a pixel level and to account for the periodic video compression noise, we model each pixel in the image as a dynamically growing vector of codewords, a so-called codebook (Figure 4.5). A codeword is represented by: the average pixel  $RGB$  value and by the luminance range  $I_{low}$  and  $I_{hi}$  allowed for this particular codeword. If an incoming pixel is within the luminance range and within some proximity of  $RGB$  of the codeword it is considered to belong to the background. During the model acquisition stage the values are added to the background model at each new frame if there is no match found in the already existing vector. Otherwise the matching codeword is updated to account for the information from the new pixel. Empirically, we have established that there is seldom an overlap between the codewords. However if this is the case, *i.e.* more than one match has been established for the new pixel, we merge the overlapping codewords. We assume that the background noise due to compression is of periodical nature. Therefore, at the end of training we clean up the values (“stale” codewords) that have not appeared for periods of time greater than some predefined percentage frames of in the learning stage as not belonging to the background. For this, as outlined in [43], we keep in each codeword a so-called “maximum negative run-length ( $MNRL$ )” which is the longest interval during the period that the codeword has not occurred. One additional benefit of this modeling approach is that, given a significant learning period, it is not essential that the frames be free of moving foreground object. The background model can be learned on the fly, which is important in the in-store setting, where the scene cannot be easily vacated of moving people. Also, in our adaptive background model, the

changing illumination conditions, such as the addition or removal of light sources (*e.g.* a light bulb burning out) can be promptly accounted for.

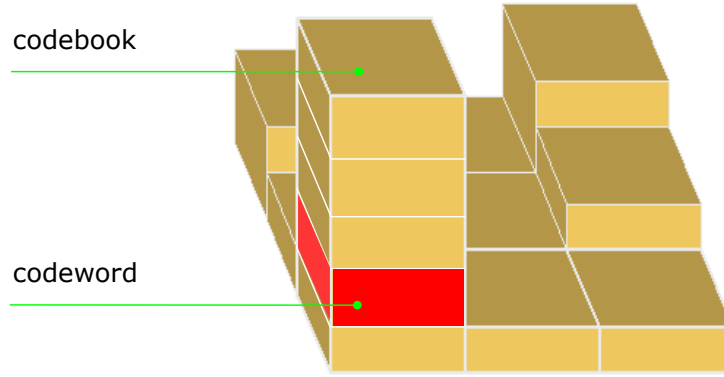


Figure 4.5: Color codebook formation. Each pixel is modelled as a stack of codewords — a codebook.

As a further enhancement, we eliminated the background learning stage as such to enable our system to operate dynamically. This was done by adding an *age* parameter to each codeword as the count of all the frames in which the codeword has appeared. Now, we can start background subtraction as soon as the majority of the codewords contain modalities that are sufficiently old. Typically, around 100 frames in our test sequences presented in Chapter 6 were enough for reliable detection of the foreground objects. This improvement also allows us to perform the removal of “stale” codewords periodically and not as a one-time event. Now, to determine the “staleness” of a codeword we consider the ratio between its *MNRL* and its overall *age*. We have found that when

employing “stale” pixel cleanup for the heavily compressed sequences the length of the codebook required to encapsulate the background complexity within one pixel is usually under 20 codewords.

Additionally, we store the last frame number  $f_{last}$  in which the codeword was activated (*i.e.* it matched a pixel). To make our model dynamic, we discard the codewords that have not appeared for long periods of time, which can be computed as the difference between the current frame and  $f_{last}$  for any given codeword. These instances indicate that the background scene has changed, possibly due to a stationary object placed or removed from the scene, thus causing our model to relearn it dynamically.

As outlined in Figure 3.1 the first critical step in the presented system is accurate and robust segmentation of moving foreground regions from a relatively static background. A blob, in general terms, is a moving region of the image that can consist of the pixels corresponding to one or more bodies mixed with the rigid moving objects (such as carts or strollers) and other noise artifacts (*e.g.* changing illumination). Each blob in our system is represented as a polygon in image coordinates.

The binary mask after background subtraction is filtered with morphological image operators to remove noise pixels and to bridge small gaps that may exist in otherwise connected blobs. This results in an array of blobs created where each blob  $b$  is represented as an ordered list of vertices  $v_i$ ,  $i = 1, \dots, n$  in two-dimensional image space. The vertices describe the contour of  $b$  in which each adjacent pair of vertices  $v_j$  and  $v_i$  is connected by a straight line.

### 4.1.3 Finding Head Candidates

The surveillance cameras are typically mounted on the ceiling, more than three meters above the ground. This can be advantageous in discriminating separate humans within a crowd. The head of a human will have the lowest chance of being occluded, therefore we pursued the goal of finding head candidates — points that represent the tops of the heads in a blob. In this section, we describe head detection in more detail.

To generate human hypotheses within a blob detected in the scene we have used a principle similar to that of the vertical projection histogram of the blob [29]. The vertical projection histogram is a way of finding peaks in the silhouette of the human body, by aggregating the values in a binary foreground mask along each vertical column of pixels. This can yield a histogram, with its maxima roughly corresponding to the head locations. Of course, this method makes the assumption that the observation is made with the front facing camera, with the optical axis parallel to the ground. Such an assumption clearly is not consistent with ceiling mounted cameras, such as the ones typically placed in retail establishments. Our novel method utilizes information about the vanishing point location we obtain from the camera during the calibration stage. The projection of the blob is done along rays going through the vanishing point instead of the parallel lines projecting onto the horizontal axis of the image.

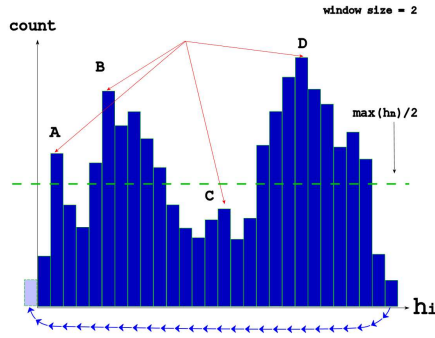


Figure 4.6: Finding extremes in VPP histogram

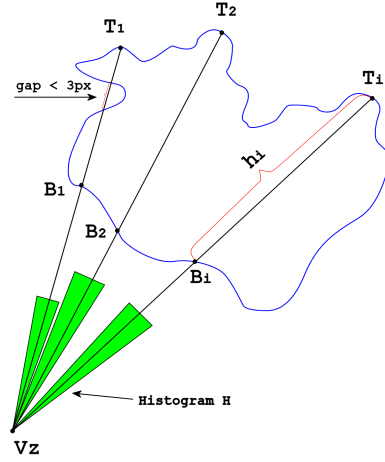


Figure 4.7: Vanishing point projection (VPP) histogram

In our implementation each foreground blob is represented as an array of contour vertices  $\mathbf{T}_i$  (see Figure 4.7), converted to homogeneous coordinates as described in Section 4.1.1. For each  $i$  our method starts at  $\mathbf{T}_i$  and counts the number of pixels  $h_i$  along the line  $r_i = \mathbf{T}_i \times \mathbf{V}_Z$  coming through the vanishing point, obtained earlier as part of the camera calibration process.

Then  $r_i$  is rasterized by Bresenham's algorithm. Notice that  $\mathbf{V}_Z$  is an ideal point which can

sometimes fall out of the image boundary or even be situated at an infinity (in the case that the 3D parallel lines are also parallel to the image plane). Therefore, we needed to modify the rasterization algorithm to stop as soon as it reaches the image boundary or  $\mathbf{V}_Z$ , whichever comes first. Note that there is no risk of the process spreading to adjacent blobs, because the foreground mask is rendered for each blob from its contour independently.

The process continues even after the end of the foreground region is reached, which can be defined as the first non-foreground pixel, to allow for important contour concavities, such as arms as well as gaps that are due to camera noise (*e.g.* see the line originating from  $\mathbf{T}_1$  in Figure 4.7). The last foreground pixel reached in such a manner is considered a bottom candidate  $\mathbf{B}_i$  and the count of foreground pixels between  $\mathbf{T}_i$  and  $\mathbf{B}_i$  is recorded into the histogram bin  $i$ .

Resulting from this is our vanishing point projection histogram  $H = [h_i]$ . We attempt to isolate local maxima in the histogram in two steps. First, the value  $h_i$  is considered a local maximum within a window if it is greater or equal to  $M$  of its neighbors on either side (Figure 4.6 shows as an example the window of size  $M = 5$ ).

$$h_i \geq h_j, i - \frac{M-1}{2} \leq j \leq i + \frac{M-1}{2}$$

Because this may result in a number of neighboring vertices with equal values of  $h$  selected as local maxima, we merge all such peaks within their window  $M$  and use their average as a candidate. In order to account for a cyclic nature of the contour for the leftmost and rightmost bins the neighbors are wrapped around from the end or the beginning of the histogram correspondingly. Typically, the window size can be determined as the total number of bins in the histogram divided by the maximum amount of candidates allowed with one blob. This number is set normally from 3 to 10 depending on the average complexity or *crowdedness* of the scene. We define the crowdedness as the number of currently visible tracked bodies. After this stage all the local peaks  $h_i < \max_n(h_n)/2$  are further removed to ensure that only the vertices that correspond to the upper parts of the body

are considered.

As a result of the head detection process at each frame of the video sequence we obtain an array of head candidates  $HC, \forall hc \in HC = \{hc_x, hc_y\}$ , where each candidate is a pair of coordinates in the image plane. A matching floor candidate  $fc = \{x_{fc}, y_{fc}\}$  is created for each head candidate to represent the point with the  $X, Y$  world coordinates equal to those of the head candidate and  $Z = 0$  (see Figure 4.8)



Figure 4.8: Foreground segmentation results. FG regions are highlighted in green. Head and floor candidates are shown with red and blue dots.

Using the same interactive approach as was used to obtain  $V_Z$  (Figure 4.7) we also compute  $V_X$  and  $V_Y$  (see Section 4.1.1 for more details). For a stationary camera, this calibration procedure has to be performed only once for the entire video sequence, assuming the position of the camera does not change. In the same manner (Figure 4.2), the user can designate a number of vertical linear segments of known height (*e.g.* aisles, shelves or boxes). Using the heights of the reference objects to compute the projection scale and knowing the positions in the image of head candidates with their corresponding floor locations we have employed the approach from [16, 15] to find human heights in meters.

For more illustrative results of foreground segmentation, head candidate detection and height estimation see Figure B.1 in Appendix B.

## 4.2 Tracking

As outlined in Section 3.2, after creating the camera and background models and specifying the appearance and shape modelling technique for a human body, we have sufficient information to build a tracking system. The key goal of the tracker is to find a correspondence between the bodies, already detected in the current frame with the bodies which appear in the next frame. In this section we show how to apply *Markov Chain Monte Carlo (MCMC)* methods to estimate the next state of the tracker system. Markov Chain Monte Carlo methods allow sampling from probability distributions based on the construction of a Markov chain that has the desired distribution as its stationary distribution. The state of the chain after a number of iterations is then used as a sample from the desired distribution. The quality of the sample improves as a function of the number of steps. We do not assume the explicit knowledge of our joint distribution, so we have chosen to use Metropolis-Hastings sampling algorithm [7], which requires only that the probability density can be calculated at a point. Below we describe the design of our tracker.

### 4.2.1 Bayesian Model: Observations and States

To implement Bayesian inference process efficiently we model our system as a Markov chain  $M = \{x, z, x_0\}$  and employ a variant of Metropolis-Hastings probability sampling algorithm [22]. The state of the system at each frame is an aggregate of the state of each body  $x_t = \{b_1, \dots, b_n\}$ . Each body, in order, is parametrically characterized as  $b_i = \{x, y, h, w, c\}$ , where  $x, y$  are coordinates of the body on the floor map,  $h, w$  its width and height measured in centimeters and  $c$  is a 2D color histogram, represented as 32 by 32 bins in hue-saturation space. The body is modeled by the ellipsoid with the axes  $h$  and  $w$ . An additional implicit variable of the model state is the number of tracked bodies  $n$ . I formulate the tracking problem as the maximization of the posterior probability of the state of the Markov chain.

### 4.2.2 Computing the Posterior Probability

The goal of our tracking system is to find the candidate state  $x'$  (a set of bodies along with their parameters) which, given the last known state  $x$ , will best fit the current observation  $z$ . Therefore, at each frame we aim to maximize the posterior probability

$$P(x'|z, x) = L(z|x') \cdot P(x'\{x\}) \quad (4.2)$$

According to Bayes theorem given in Equation (4.2) we formulate our goal as finding the maximum a posteriori:

$$x' = \operatorname{argmax}_{x'} (L(z|x') \cdot P(x'\{x\})) \quad (4.3)$$

The right hand side of Equation (4.3) is comprised of the **observation likelihood**  $L(z|x')$ , given the proposed state and the **prior probability** of the proposed state. They are computed as joint likelihoods for all bodies present in the scene as described below. The prior  $P(x'\{x\})$  is deliberately shown as a function of the previous state  $x$  to reflect the fact that a rule is applied to a first order Markov chain.

Subsequently we use Metropolis-Hastings sampling algorithm [7] to estimate the next state of the chain, by random sampling.

$$\alpha(x_t, x') = \min \left( 1, \frac{\pi(x')}{\pi(x_t)} \cdot \frac{m_t(x_t|x')}{m_t(x'|x_t)} \right). \quad (4.4)$$

Where  $x'$  is the candidate state,  $x_t$  is the current state,  $\pi(x)$  is the stationary distribution of our Markov chain,  $m_t$  is the *proposal distribution*. In Equation (4.4), the first part is the likelihood ratio between the proposed sample  $x'$  and the previous sample  $x_t$ . The second part is the ratio of the proposal density in both directions (1 if the proposal density is symmetric).

This proposal density would generate samples centered around the current state. We draw a new proposal state  $x'$  with probability  $m_t(x'|x)$  and then accept it with the probability  $\alpha(x, x')$ . The proposal distribution is a time function, that is at each frame it will be formed based on the rules outlined below. If  $x'$  is accepted, it becomes the new state of the system  $x_{t+1} = x'$ , otherwise the system reverts to the current state  $x_{t+1} = x_t$ .

To form the proposal distribution we have implemented a number of *reversible operators* or *state mutations*. There are three types of jump transitions *Create new body*, *Remove a body*, *Recover a body* and three types of diffuse transitions *Mean-Shift*, *Move*, *Resize* implemented in our system [46]. We give empirically chosen weights to each mutation to add more emphasis to one or another transition type. In our application normally around 100 jump-diffuse iterations are required for each frame to reach convergence.

The generalized algorithm for choosing and accepting the new state of the tracker is presented in Algorithm 1.

```

for  $i \leftarrow 1$  to  $N_{mutations}$  do
    apply one of the mutations probabilistically to derive a new state  $x'$ ;
    compute observation likelihood  $L(z|x')$ ;
    compute prior  $P(x'|x)$ ;
    if  $\alpha(x, x') = 1$  then
         $x_{t+1} = x'$ ;
    else
         $x_{t+1} = x_t$  with probability  $\alpha$ ;
    end
end

```

**Algorithm 1:** MCMC algorithm for generating a new state of the tracker

### 4.2.3 Body Shape Modelling

Each human body is modelled by a spheroid  $E$ , which is a special case of an ellipsoid, with axis  $a_3 = c$  corresponding to body height and axes  $a_1 = a_2 = a$ , representing body width  $b(w)$ . Experimental trials have shown no significant improvements with the introduction of the tilt angle

of the spheroid, therefore  $E$  is vertically oriented.

The north pole of the spheroid is initialized to coincide with the head candidate  $\bar{V} = hc$  and the bottom peak  $\underline{V} = fc$  is positioned to match the floor candidate with  $Z_{fc} = 0$  being on the floor plane.

For the purposes of efficient computation of body histograms and foreground correspondences, ellipsoids have to be projected onto the 2D image plane. In the perspective projection model ellipsoids are projected to form ellipses in image plane [30]. In the panoramic model the projection results in a distorted ellipse-like shape. Projection implementation details are described in the Appendix A.

#### 4.2.4 Body Appearance Modelling

Let  $b_i$  be a body, bounded in the image space by the rectangle  $x_{min}, x_{max}, y_{min}, y_{max}$ , for which a 2D color histogram  $C$  is to be computed, then a Gaussian weight kernel can be defined as:

$$\begin{aligned}
 K_{x,y} &= A \exp\left(-\left[\left(\frac{(x - (x_{max} - x_{min})/2)}{\sigma_x}\right)^2 + \left(\frac{(y - (y_{max} - y_{min})/2)}{\sigma_y}\right)^2\right]\right) \\
 \sigma_x &= (x_{max} - x_{min})/3 \\
 \sigma_y &= (y_{max} - y_{min})/3
 \end{aligned} \tag{4.5}$$

Where  $A$  is a normalization coefficient, and standard deviations make sure that all significant weights fall within the elliptical region of the body. Gaussian weight kernels  $K$  are also shown in Figure 4.9. The purpose of such a kernel is to simulate the likelihood of any color pixel to be representative of the one of the body's persistent colors. We assume that each body has a fixed set of intrinsic color descriptors, for instance skin color and hair or clothing and accessories. In any given moment the observed colors vary based on the position and orientation of the body, various

articulated transformations (squatting, joint bends, etc) and on the nearby objects and lighting conditions. Let us assume that all of the above factors are treated as random, having a limited variance, and the center of the physical body is less prone to such noise. Then, according to the central limit theorem, given enough observations, the noise factors will be distributed normally.

To make use of the foreground subtraction, we use color information only from the foreground pixels. For this purpose we create a binary mask  $M \in [0, 1]$ , which selects only the visible parts of the blob that correspond to the current body pixels  $O_i$  and is computed as:

$$M = O_i \bigcap (Z = i) \quad (4.6)$$

In Equation (4.6)  $Z$  represents a discrete z-buffer defined in Section 4.2.6.

#### 4.2.5 Priors

In creating a probabilistic model of a body we considered three types of prior probabilities. The first type of priors imposes physical constraints on the body parameters. Namely, body width and height are represented by Gaussian densities  $N(h_\mu, h_{\sigma^2})$  and  $N(w_\mu, w_{\sigma^2})$ , with the corresponding means and variances reflecting the dimensions of an adult human body physique. Body coordinates  $x, y$  are weighted uniformly within the rectangular region  $R$  of the floor map. Since we track bodies which may be partially outside of the image boundaries,  $R$  slightly exceeds the visible part of the image to account for such cases.

The second type of prior probabilities sets the dependency between the candidate state at time  $t$  and the accepted state at time  $t - 1$ . First, the priors reflect the constancy of the physical size of the body. Therefore we set the difference between  $w_t, h_t$  and  $w_{t-1}, h_{t-1}$  to reduce the prior probability. Second, we impose the motion smoothness constraint by using the distance between proposed body position  $(x_t, y_t)$  and  $(\hat{x}_{t-1}, \hat{y}_{t-1})$ . Higher distances result in lower prior probabilities. Position estimates  $(\hat{x}_{t-1}, \hat{y}_{t-1})$  are generated by running the prediction step of the constant velocity Kalman

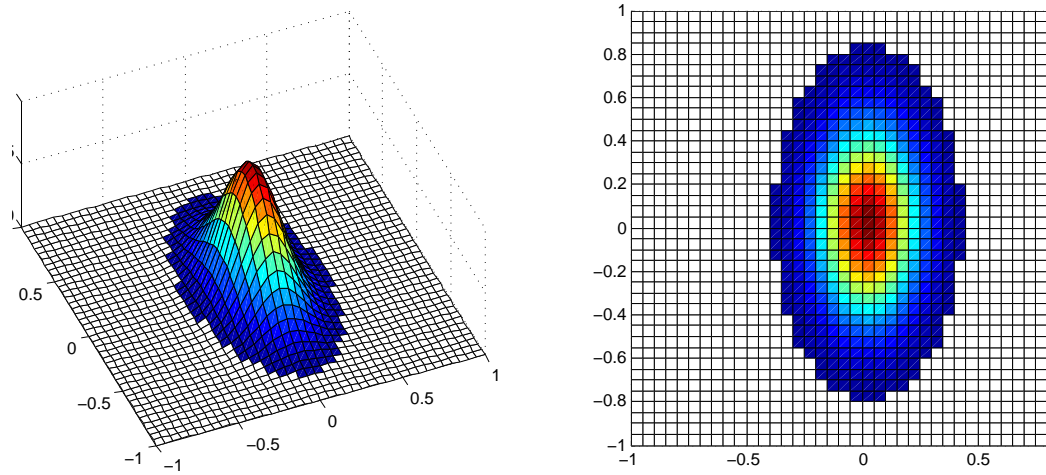


Figure 4.9: Gaussian kernels used in the computation of weighted histogram and in the anisotropic mean-shift tracking (see Section 4.2.6)

filter on body position at the previous frame  $(x_{t-1}, y_{t-1})$ . The state of the Kalman filter consists of the location of the body on the floor and its velocity. Although tracking the head seems like a first reasonable solution, we have established empirically that the perceived human body height varies as a result of walking, thus the position of the feet on the floor was chosen as a more stable reference point. A second order Kalman filter, accommodating for body accelerations, was also tested with no significant improvement in accuracy.

The third type of priors are physical constraints with respect to other moving and static objects in the scene. First, to avoid spatial overlap between adjacent bodies (as physically impossible) we have imposed penalties on pairs of pedestrian models located closer than their corresponding body widths would allow. Second, a similar constraint was imposed on an overlap between pedestrians and stationary obstacles, which were manually marked in the frame and converted to 3D world coordinates (see Figure 4.10). We modelled each obstacle in the scene as a cuboid, with the lower facet lying within the floor plane, the system has a potential of using other types of polyhedra, and quadrics to represent obstacles.

When a new body is created it does not have a temporal match in the previous state of the system. In this case we use a normally distributed prior  $N(d_0, \Sigma)$ , where  $d_0\{x_f, y_f\}$  is the location of the closest door (designated on the floor plan) and  $\Sigma$  is chosen empirically to account for image noise. The same process is taking place and the same measure is used when one of the existing bodies is deleted.

#### 4.2.6 Likelihoods

The second component in forming the proposal probability density relates the observation to the model state. First, for each existing body model the color histogram  $c$  is formed by the process of weighted accumulation, with more recent realizations of  $c$  given more weight. We then compute the Bhattacharyya distance  $B(c'_t, c_{t-1}) = \sqrt{c'_t c_{t-1}}$  between proposed histogram  $c'_t$  and corresponding  $c_{t-1}$  in the last frame as part of the observation likelihood (see Equation (4.7)).

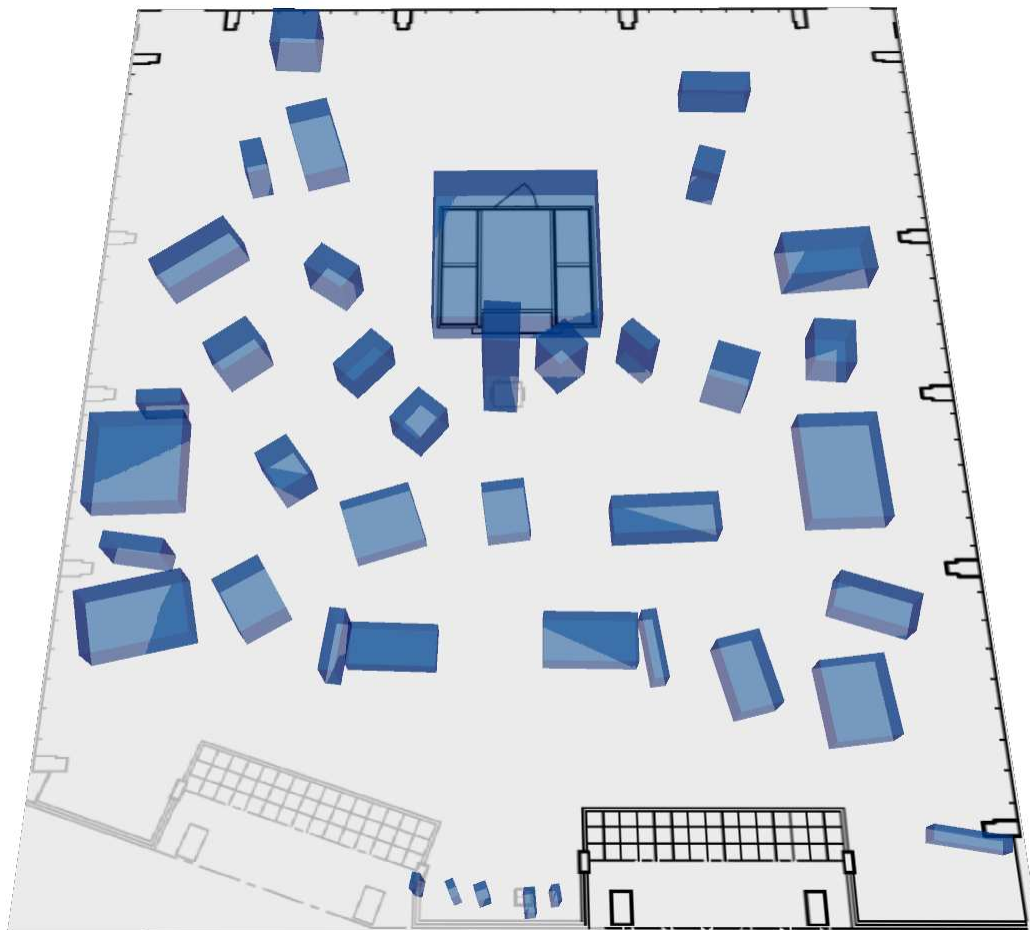


Figure 4.10: Manually generated floor plan that contains information about the static obstacles in the store

$$P_{color} = 1 - w_{color}(1 - B(c'_t, c_{t-1})), \quad (4.7)$$

where  $w_{color}$  is an importance weight of the color matching

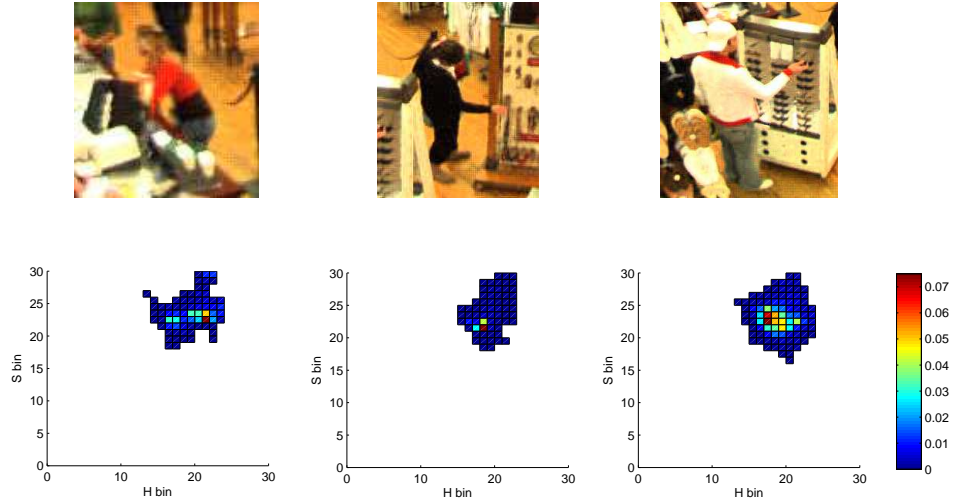


Figure 4.11: Hue-saturation 2D histograms, with 30 by 30 bins. Examples show for three currently tracked bodies.

Color histograms are computed with an anisotropic Gaussian weight mask as described in mean-shift mutation below (see Equation (4.5) and Figure 4.14).

To guide the tracking process by the background map at hand, we use two more components while computing the model likelihood: we define the amount of blob pixels not matching any body pixels as  $P^+$  and the amount of body pixels not matching any blob pixels  $P^-$  (see eq. (4.8),(4.9)).

A *z-buffer*  $Z$  is used to compute the blob to body correspondencies as well as for computing the color histogram of the current observation in order to detect occlusions. In this buffer all the body pixels are marked according to their distance from the camera (*i.e.* 0 = background, 1 = furthestmost body, 2 = next closest body, *etc.*). This way, only visible pixels are considered when computing

the likelihood (see Figure 4.12). The Z-buffer is updated after each transition to reflect the new occlusion map.

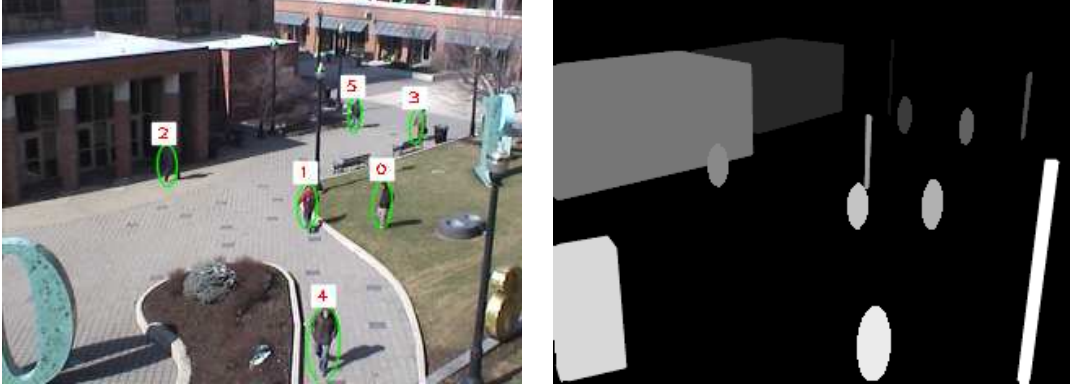


Figure 4.12: Z-buffer *Left*: Original frame with tracked pedestrians *Right*: Z-buffer (lighter shades of gray are closer to the camera)

In computing the likelihood as outlined above, there is one major shortcoming overlooked in the previous works [38, 71]. If the computation is done in terms of the numbers of image pixels, then the bodies closer to the camera to dominate the overall configuration, and the bodies further away are correspondingly neglected. This becomes particularly evident when the camera covers a large area, where pedestrian image presentations can vary from under 20 pixels of overall area in the back of the scene to more than 200 in front. In addition, such neglect makes the system very specific to the current camera configuration and not portable to a different camera model.

However, it is impossible to fully switch to world coordinates while manipulating with blobs in image coordinates, since no 3D body models have been assigned yet. To address these shortcomings we have utilized a so-called “distance weight plane”  $D$  which is the image of the same dimensions as the input frame and  $D_{xy} = |P_{XYZ}, C_{XYZ}|$ , defined as the Euclidean distance between the world coordinates of the camera  $C_{XYZ}$  and the world coordinates of the hypothetical point  $P_{XYZ}$  located at a height  $z = \frac{h_\mu}{2}$  and corresponding to the image coordinates  $(x, y)$ . The map produced in this manner is a rough assessment of the actual size to image size ratio (see Figure 4.13).

To summarize, the implementation of **z-buffer** and **distance weight plane** allows computing



Figure 4.13: Distance weight map. *Left*: Original frame with tracked pedestrians *Right*: Distance weight plane (weights increase from blue to red)

multiple-body configuration with one computationally efficient step. Let  $I$  be the set of all the blob pixels and  $O$  be the set of all the pixels corresponding to bodies currently modelled, then

$$P^+ = \sum \frac{(I \setminus O \cap Z_{(Z_{xy} > 0)}) \cdot D}{|I|} \quad (4.8)$$

$$P^- = \sum \frac{(O \cap Z_{(Z_{xy} > 0)} \setminus I) \cdot D}{|O|} \quad (4.9)$$

where  $' \cap '$  is set intersection,  $' \cdot '$  is element-wise multiplication;  $\setminus$  is set difference and  $'| \cdot |'$  is set cardinality (number of pixels).

#### 4.2.7 Jump-diffusion Transformations

The classical Metropolis-Hastings method operates on the states of fixed dimensionality. A technique called *reversible jump* has been used to allow the change of the dimensionality of proposal distribution [26]. In essence, our approach of probability sampling is a non-deterministic multivariate optimization method. As such it inherits the problems to which other, classical optimization methods can be prone [22]. Here we present a way to overcome one such problem — traversing

valleys in the optimization space by using task specific information. Despite this, random sampling methods are robust because they do not require any assumptions about the probability distributions of the data.

**Create:** Draw a random head candidate/floor candidate pair  $(hc, fc)$  and create a new body model  $b$  using its head and foot coordinates. At this point for the tuple  $b = \{x, y, h, w, C\}$  the actual height and floor coordinates of the body are estimated (see Section 4.1.1). The width of the newly created body is set originally to the mean  $w_\mu$ .

Several heuristic enhancements are applied at this step to optimize the random space traversal. If  $b\{h\} > h_\mu + 3h_\sigma^2$ , *i.e.* the body is untypically tall, it gets split into two bodies of identical height  $h_mu + rand(-1, 1)h_\sigma^2$ , which, in the world coordinates, corresponds to two pedestrians standing on the same projection line and partially occluding each other. Also, if the initial candidate is too short  $b_i\{h\} < h_\mu - 3h_\sigma^2$  the body is expanded around it's center, to a random new height  $\hat{h} = h_mu + rand(-1, 1)h_\sigma^2$ . While creating and deleting bodies, the distance from the door  $N(d_0, \sigma)$  is accounted for in such a way that only the candidates close enough to the existing doors can be added or removed from the scene.

**Remove:** Remove a randomly selected body  $b$ . The body is excluded from further tracking, the path is terminated and saved. Additionally, to improve computational efficiency, the bodies repeatedly showing no underlying blob pixels are periodically removed from the scene.

**Recover:** Recover a recently deleted body from an array of bodies, removed within a recent time window  $\delta(t)$ . The time window is chosen to be long enough to include sudden illumination aberrations and short-term complete occlusions (*e.g.* a person is walking past a tall shelf for 2 seconds). The *recently deleted array* is maintained as a FIFO queue and is updated every time a new body is removed. This mutation dramatically reduced the number of iterations needed in the system to

overcome multiple full occlusions, taking place frequently in crowded scenes. This way each track is presented as a continuous array of observations (one for each frame), with a number of records marked as *invisible*. The invisible records are not used for further activity recognition.

**Mean-shift:** Move one of the existing bodies by applying the mean-shift operator [14] with weighted anisotropic Gaussian kernel. The kernel is formed as a Gaussian, elliptically-shaped mask, where the weights increase with increased Mahalanobis distance. Additionally, if a pixel value of the foreground mask (corresponding to the background) is zero or the same pixel value from the Z-buffer is greater (*i.e.* located further from the camera) than the current body, the weight in the kernel is effectively zeroed out. This, in essence, performs a standard color-based mean shift, but accounts only for the pixels belonging to the hypothesized body model.

The mean-shift gradient descent is performed based on the Bhattacharyya distance between color histograms, as computed in Equation (4.7).

**Move:** Second type of position shift is moving the body to a random “initial head candidate” drawn from a pool of head candidates contained in some proximity from the current body position. It allows for the head candidates, not initially revealed (possibly due to image noise), to be considered in the subsequent frames. The Z-buffer is updated after each such transition to reflect the new occlusion map.

**Resize:** Change the height or width of a random body. Either a new height  $\hat{h} = h_m u + rand(-1, 1)h_\sigma^2$  or a new width  $\hat{w} = w_m u + rand(-1, 1)w_\sigma^2$  is generated for  $b_i$  and the existing  $h$  or  $w$  is replaced by it.

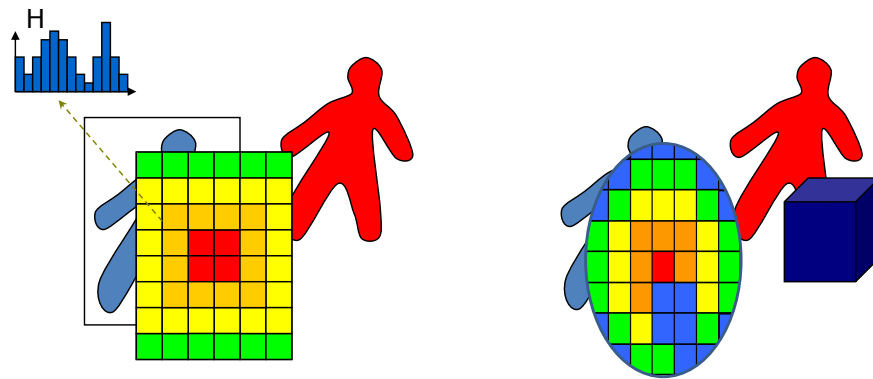


Figure 4.14: Mean-shift illustration: *Left:* Canonical mean-shift with rectangular kernel *Right:* Anisotropic kernel, with depth mask accounting for obstacles

## Activity Analysis

### 5.1 Modeling Group Activities

In this section we present a general methodology for modeling and subsequent automatic classification of the high-level activities of groups of people. To proceed efficiently we had to answer the question of mapping the knowledge efficiently onto the computation to yield a productive methodology. In our case the interpretation must describe a number of high-level human activities while at the same time retaining a certain level of mathematical formalism. With respect to the representation the problem can be divided into three levels of abstraction:

**Level 1:** At the lowest level, a model of each tracked body is viewed as a vector of features:

- $x, y$  location of the human body (in floor coordinates) identified by the tracker,
- $w, h$  human height and width in centimeters
- $\bar{c}$  32x32 2D color histogram in hue-saturation space
- $\alpha$  body motion orientation
- $p$  blob parameters, such as size in pixels and speed of change, can also be used but tend to be noisy

**Level 2:** The motion of each body is viewed in a goal-oriented framework, where the customer has goals and sub-goals and the rest of the frames reflect the process of achieving these goals [20]. The types of goals in a retail store may include such actions as: finding an aisle, browsing a fixture, finding a product, inspecting a product, finding a sales representative, interacting with a sales representative, proceeding to checkout, etc. The list is by no means exhaustive but is indicative of the range of activities by individual customers. We directly observe from this list that all such activities can be regarded as either walking, *i.e.* progressing with a more or less constant speed toward a goal or a sub-goal; or dwelling, (*i.e.* being situated in a fixed area with no or insignificant or undirected motion).

**Level 3:** The higher-level types of customer activity which are indicative of shopping habits are the average size and formation of shopper groups. Such groups are typically characterized by the large periods of motion coordination between the actors with a group. At this level of representation simple motion tasks from one or multiple actors are grouped to form more complex behaviors. The types of retail customer activities to consider can be classified into two categories. The first category is personalized activities like interacting with products, reacting to advertisements, searching for an item or help, avoiding traffic or browsing. The second category is group activities: customer assistance, shopper groups, checkout line. From the wide array of subtasks naturally occurring in retail environments, we have singled out one that is of primary interest to marketing research and yet is feasible to solve as a computer vision problem, given the physical limitations of the framework. We detect “shopper groups”, which can be further used as a means to interpret shopping habits of an average customer.

We have considered a number of approaches to model spatial group behaviors. Most prominent in the literature is the approach to model each individual's state as one of the hidden Markov model (HMM). Further, to emulate a group activity, HMMs can be coupled or layered to represent different levels of behavior complexity. While suitable for certain types of activities (mostly, short term “actions”), first order HMMs are limited in their power to link with the previous states, since most

human activity is non-Markovian, *i.e.*, it does not depend only on the previous state. As a generalization for the group behavior, the variants of Bayesian networks have been used with each node, modeled as a Markov chain, however these are typically limited by the number of agents taking part in group activity. A different view is building behavioral models of one or many actors by training an artificial neural network classifier. The advantage of this method is that it can learn hidden dependencies without the knowledge of priors and thus can be used to classify a wide range of activities. The major drawback here is the need for many training samples to simulate prior knowledge, which in most cases are not readily available or hard to generate [56].

Despite the relative success of the methods described above in detecting short-term or individual actions, their performance is limited in detecting group activities. We developed a novel two-stage approach that fits well with the idea of a three-layer abstraction of activities, ranging from lower-level observations, to task detection, to the detection of behaviors. We provide a detailed account of how clustering techniques can be used to convert tracking information into a set of short-time behavioral events and further to group such events in the blocks, representative of the grouping behavior. In our work we applied **hierarchical** or **agglomerative** clustering in two instances both of which are characterized by the presence of non-euclidian distances. More specifically, the rules that hold for standard "cloud of Euclidian points" clustering, (*e.g.* the fact that the cluster can be simply regarded as a centroid of its elements) are not applicable under the presence of metric distance measures, for which no addition operations can be defined. Since no cluster mean can be determined, some classical clustering validity measures (such as Davies-Bouldin index [18]) cannot be applied. Additionally, the intuition behind the agglomerative approach is that the algorithm starts by introducing the least possible bias into the entire process. In other words, most likely merges are made in the beginning when the price of error is the highest.

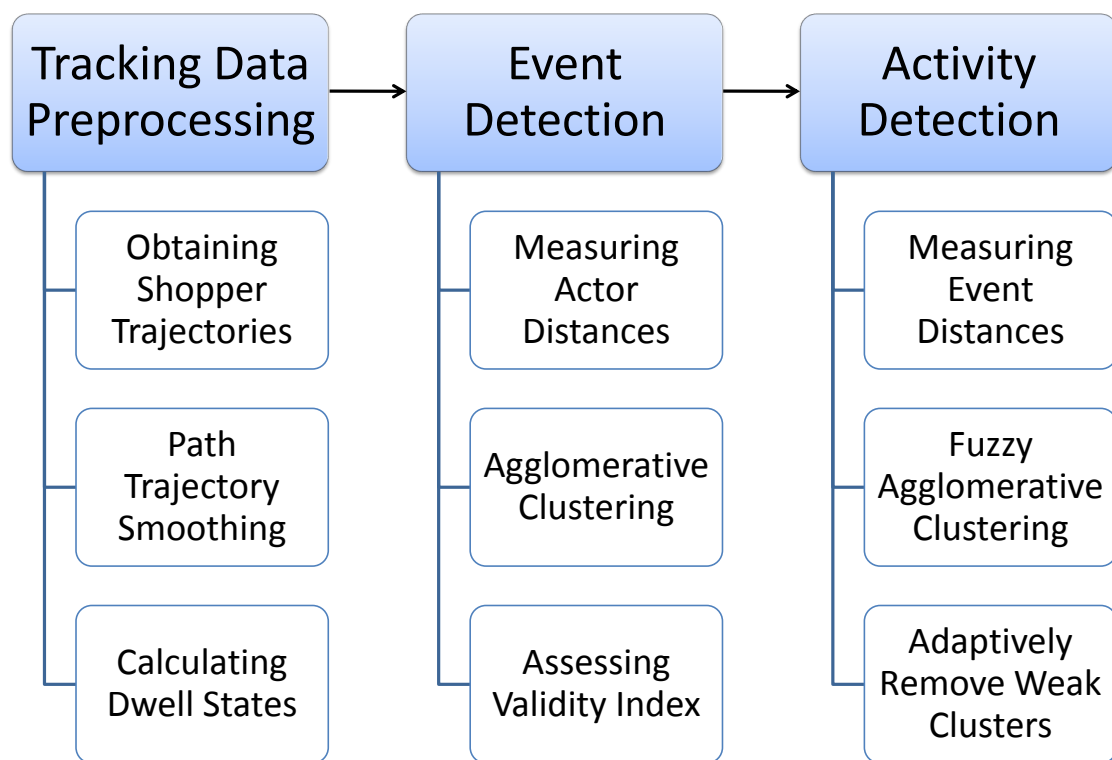


Figure 5.1: Major processing components of activity detection algorithm

The outline of this chapter is presented in Figure 5.1. We first describe the pre-processing steps which we use to form time series suitable for our analysis. Then we define the distance measures used by our event detector and activity recognition algorithm. Finally we provide the outline of the clustering techniques we use to detect grouping of customers.

## 5.2 Obtaining Shopper Trajectories

As each accepted body candidate progresses along the floor map, we record at each step the  $x, y$  coordinates along with the body orientation angle  $\phi$  and a unique body ID and append it to a separate data structure  $A_i = \{x, y, \phi, ID\}$ , where  $i$  is the path number. This way at each time moment the array  $A$  represents a complete trajectory history for all the bodies traveled in the scene since the start of a sequence.

Equipped with this information, we can accomplish several low-level activity analysis tasks for retail marketing applications, such as identifying queue lengths and queue wait times, building store traffic heat maps, aisle penetration maps and, equipped with the purchase data, computing customer conversion rate.

### 5.2.1 Path Trajectory Smoothing

At the level of a single frame, inaccuracies arise in tracking the  $x, y$  locations on the floor map. The inaccuracies of the first type are primarily due to the imprecise camera calibration and camera lens radial distortions. These inaccuracies, however, do not create significant fluctuations from frame to frame. Erratic behavior of the paths comes, we observed, from the second type of inaccuracies — the image noise, such as shadows or false foreground regions, mistaken for the shifts in the positions of the bodies by the jump-diffusion tracker.

A Kalman filter does not have a sufficient effect on smoothing the trajectories because it accounts explicitly only for the last observation. To bring out the major trends in customer walking

trajectories and remove erratic motion patterns we applied Gaussian smoothing. This way, each point along the path  $p_{t_0}$  is taken as

$$p_{t_0} = \sum_{t=t_0-\sigma}^{t=t_0+\sigma} p_t \cdot N(t; t_0, \sigma^2) \quad (5.1)$$

Where  $N$  is a density function for normal distribution with the mean  $t_0$  and variance  $\sigma^2$ .

### 5.3 Event Detection

A radically new approach — the perspective borrowed from artificial life — is to employ a generative model with multiple independent agents acting according to simple rules [65]. This is referred to as a *multi-agent system* or MAS. Consider the case of shopper groups, that is, people who shop together. We can reason about each person in a group as an independent agent acting according to the following rules [57]:

- avoid collisions with the neighbors
- maintain fixed distance with the neighbors
- coordinate velocity vector with the neighbors when in motion

Swarming events  $e$  are defined as short term activity sequences of multiple agents interacting with each other. An agent  $b$  is an instance of a customer's path generated by the tracker, taken at the current frame.

Depending on the types of swarming events to be detected, various proximity measures or other heuristics are used. In the case of grouping events, for each actor we used the relative position on the floor  $p$ , body orientation azimuth  $\phi$  and binary dwelling state  $\delta = [T, F]$  to compute the distance metric as follows:

$$d(b_i, b_j) = w_1 ||p_i, p_j|| + w_2 |(\phi_i - \phi_j)| + w_3 (\delta_i \leftrightarrow \delta_j) \quad (5.2)$$

This way a distance is computed as a linear combination of three components: cohesion, co-alignment and co-dwelling with weights  $w_1$ ,  $w_2$  and  $w_3$  representing relative importance of each rule. Cohesion is the Euclidean distance between two points on the floor plane, co-alignment is the minimal absolute difference between two motion orientations (each in the range from 0 to  $2\pi$ ), and co-dwelling is the logical equality between two dwelling states.

**Co-dwelling Explained.** Our ultimate goal is to model customer orientation using the orientation of the body, direction of movement, blob motion, facial color and co-relation with other actors and objects. For each type of motion a different approach is required for assessing the predominant orientation of the body. While for the walking customer the prevalent factor may be the direction of motion, for the dwelling person a more detailed inspection of the focus of attention, involving body shape and color or texture analysis is required [45]. From the behavior recognition point of view the subsequent goal is to incorporate the knowledge of the individual's state and attention to model complex spatial interactions in the group. For this, two types of customer dynamics can be considered: walking and dwelling. The last is identifiable through detection of customer dwell events: implemented in [48].

### 5.3.1 Deterministic Agglomerative Clustering of Bodies

Once the metric is defined, the algorithm starts out with each body/actor representing a singleton cluster and iteratively applies agglomerative clustering procedures. We sample every  $\Delta$ 'th frame of the tracking sequences and perform the following operations with respect to the visible bodies found in that frame. The distance  $d(b_i, b_j)$  from Equation (5.2) is computed for each pair of  $i$  and  $j$ . The closest pair is found and the bodies are merged. For each new step in clustering process, given the

current clustering  $C_n, n \in [1, N]$  a clustering validity index (see [3]) is computed as follows:

$$\begin{aligned}
 I &= I_i + I_c \\
 I_i &= \frac{\sum_{n=1}^N \sum_{m=1}^M a_m \in C_n}{N} \\
 I_c &= \frac{\sum_{n=1}^N \left[ 1 - \mu \left( \frac{D(C_n)}{D(\{C\})} \right) \right]}{N}
 \end{aligned} \tag{5.3}$$

Where  $D(C_n)$  is the diameter of the cluster  $C_n$  and  $D(\{C\})$  is the diameter of the combined cloud of elements.

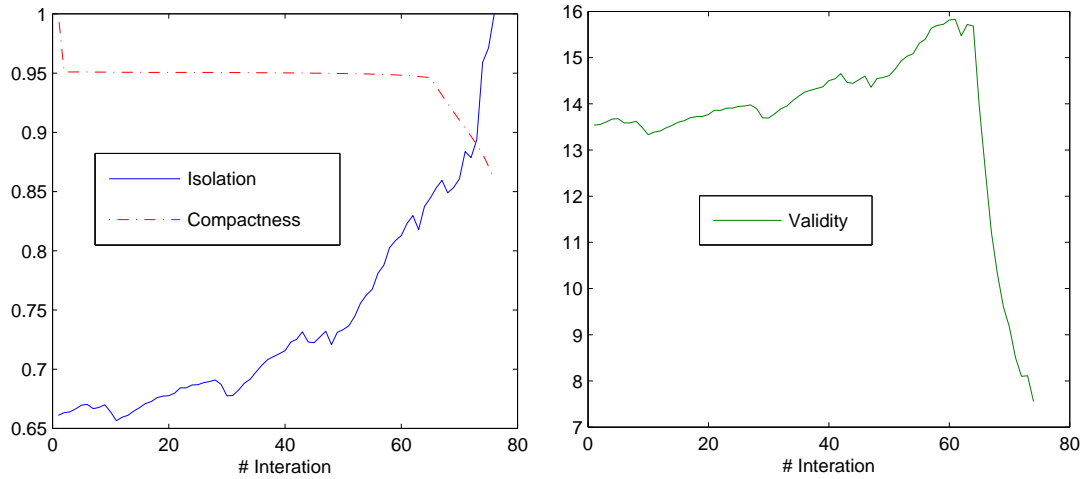


Figure 5.2: Clustering validity. *Left:* isolation  $I_i$  and compactness  $I_c$  *Right:* Combined validity index  $I$

The validity index consists of the isolation index  $I_i$  and compactness index  $I_c$  computed over all existing clusters and normalized by their number. The isolation index for each node shows the percentage of nearest neighbors that belong to the same cluster. The compactness index indicates how compact the clusters are in comparison to the diameter of the entire node cloud. The clustering process continues until the validity index  $I$  stops increasing. This is indicated by the sign of the difference between  $I$  at a current step  $n$  and  $I$  at the previous step  $n - 1$

## 5.4 Activity Detection

Swarming activities are defined as prolonged higher level behavioral activities involving multiple agents and comprised of one or more swarming events, possibly distant in time. We introduce a method of grouping swarming events into such activities based on their time co-ordination and agent composition.

Let  $B_{e_i} = \{b \in e_i\}$  be the set of all bodies/actors  $b_j$  taking part in the event  $e_i$ . Also let  $\tau_{e_i}$  and  $\tau_{e_j}$  be the average times of events  $e_i, e_j$  correspondingly happening, measured in frames (*i.e.* a mean of event start frame and event end frame).

$$D_e^2(e_i, e_j) = \{\lambda_1 D_{actors} + \lambda_2 D_{time}\}^2 \quad (5.4)$$

$$D_{actors}(e_i, e_j) = \sigma_1 \left( \frac{|(B_{e_i} \setminus B_{e_j}) \cup (B_{e_j} \setminus B_{e_i})|}{|B_{e_i} \cup B_{e_j}|} \right)$$

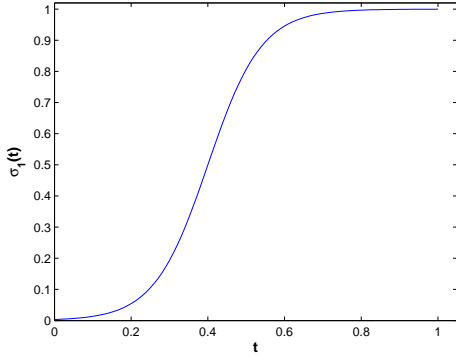
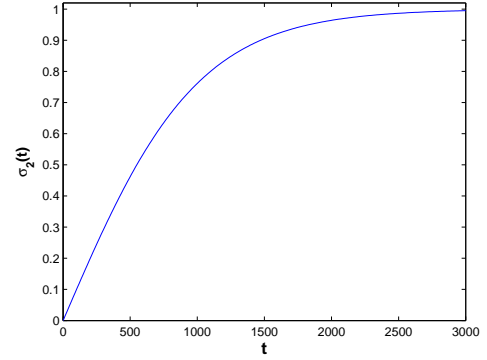
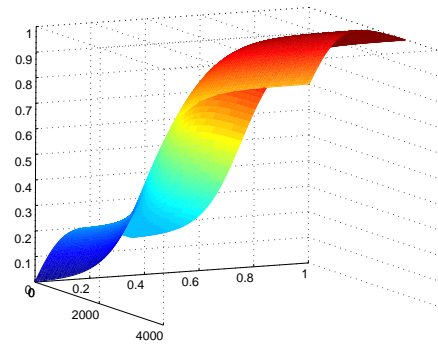
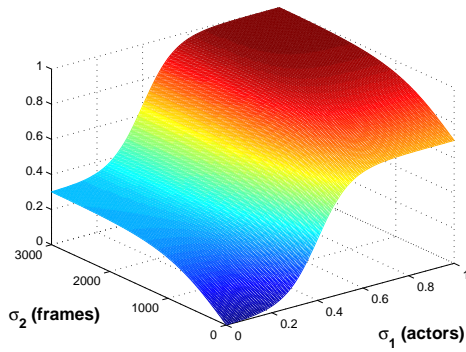
$$D_{time}(e_i, e_j) = \sigma_2(|\tau_{e_i} - \tau_{e_j}|)$$

Where “ $\setminus$ ” — is a set difference operator. The Equation (5.4) consists of two parts. The *actors* part of the equation measures similarity in actor compositions of two events, with all actors matching as the closest case and none of the actors matching as the other extreme. The *time* component is the distance in time. Relative weights  $\lambda_1, \lambda_2$  were set 0.7 and 0.3 to prioritize the fact that events that belong to the same activity tend to have same participants.  $\sigma_1$  and  $\sigma_2$  are sigmoid functions defined as:

$$\sigma_1(t) = \frac{1}{1 + \exp \frac{-(t-\Delta)}{S}} \quad (5.5)$$

$$\sigma_2(t) = \tanh \left( \frac{t}{S} \right) \quad (5.6)$$

For  $\sigma_1$ , its argument range is  $[0, 1]$ , therefore we empirically choose the shift  $\Delta = 0.4$  and scale  $S = 0.07$  to get a function profile which would not punish excessively for the discrepancy of only a few actors. The reasoning is that such a difference may be due either to tracking noise or to shopping group volatility.  $\sigma_2$  measures the distance in frames (with 15 frames in one second), therefore we set the  $S = 1000$  to account for the fact that only distances of several dozen seconds or more are considered significant. Figures 5.3, 5.4 show profiles of both functions.

Figure 5.3: Function  $\sigma_1(t)$ Figure 5.4: Function  $\sigma_2(t)$ Figure 5.5: Two-dimensional profile of distance function  $D_e^2$

Having computed the distance between any pair of events  $D_e^2(e_i, e_j)$ , the distance from an event to the activity (which is a number of events grouped together) can be measured by Equation (5.7). The measure is essentially a distance from a node in hyperspace to a centroid of the cluster of the nodes and is used in clustering of events (see Section 5.4.1)

$$D_a^2(a_i, e_j) = \frac{\sum_{\forall e_k \in a_i} u_{ik}^2 \psi_{ik} D(e_k, e_j)}{\sum_{\forall e_k \in a_i} u_{ik}^2 \psi_{ik}} \quad (5.7)$$

Where  $\rho$  and  $\psi$  are an asymmetric variant of Tukey's biweight estimators from robust statistics theory [32, 27] (see Section 5.4.1) and  $u_{ik}$  is the contribution weight of the event  $e_k$  to the activity  $a_i$ , with  $\sum_{\forall k} u_{ik} = 1$ . As can be observed from Figure 5.6 there are two clearly visible ridges in our distance function  $d$ . The first type arises from the the increase in the number of matching actors and the second from the drop in the average time distance between two events

#### 5.4.1 Fuzzy Agglomerative Clustering of Events

Fuzzy methods are increasingly used to elegantly handle cluster membership ambiguities in data with ill-defined cluster borders [25]. Generally, a classical c-means clustering algorithm is extended to include fuzzy cluster membership weights (see [1]). This way each single node can probabilistically belong to multiple clusters, with the high probabilities indicating the higher degree of confidence.

We use an approach based on agglomerative clustering [25] with fuzzy weight and distances, based on robust statistics we aim to address some of the major shortcomings of conventional partitional clustering: manual choice of number of clusters, problematic initialization, sensitivity to outliers and measurement noise.

Let  $A = \{a_i, i = 1, \dots, C\}$  be a set of  $C$  swarming activities and  $E = \{e_j, j = 1, \dots, N\}$  a set of  $N$  swarming events. Each event's  $e_i$  degree of membership in cluster  $a_j$  is expressed as a fuzzy membership weight  $W = [w_{ij}]$ , such that  $\sum_{i=1}^C w_{ij} = 1, for 1 \leq j \leq N$ .

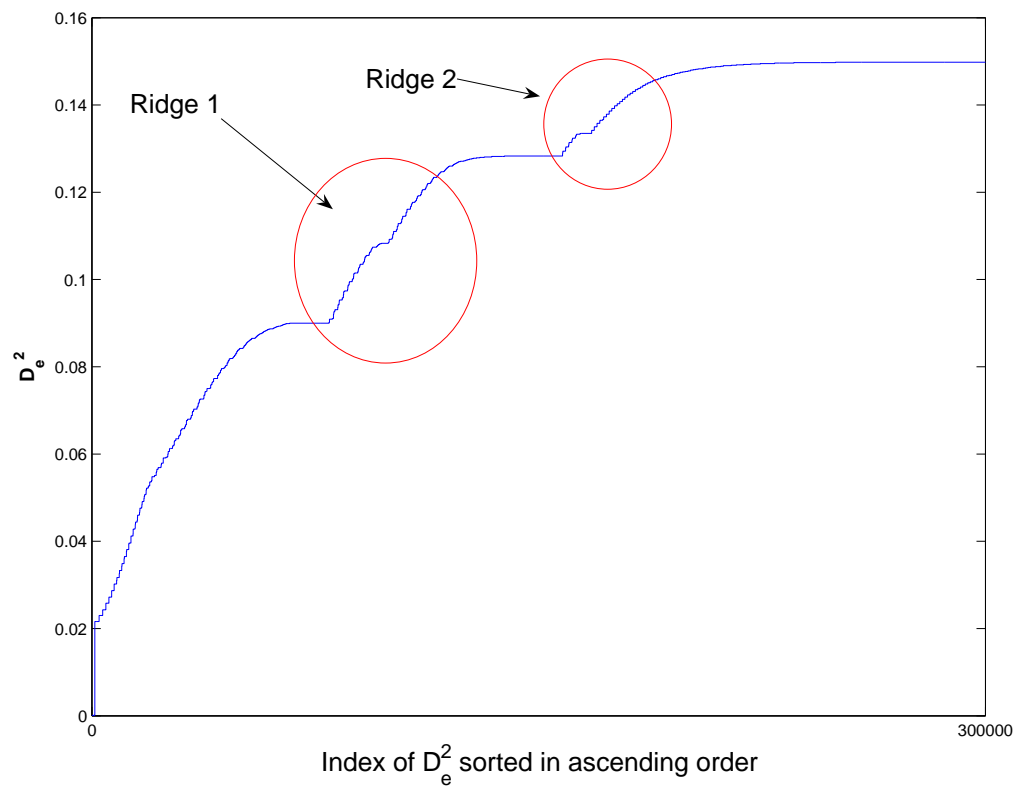


Figure 5.6: Profile of distance function  $D_e^2$  (see eq. (5.4)): sorted pairwise distances between  $\approx 1000$  swarming events

Then the goal of the method is to minimize the objective function:

$$F(A, W; E) = \sum_{i=1}^C \sum_{j=1}^N (w_{ij})^2 \rho(D_{a_{ij}}^2) - \alpha \sum_{i=1}^C \left[ \sum_{j=1}^N w_{ij} \psi(D_{a_{ij}}^2) \right]^2 \quad (5.8)$$

Where  $\rho$  and  $\psi$  are an asymmetric variant of Tukey's biweight estimators from robust statistics theory [32, 27]. The purpose of Bisquare estimators is to reduce the influence of outliers (*i.e.* in our case extremely large distances) and thus stabilize the process of clustering. The loss function  $\rho \in [0, \text{med}_i(D_{a_{ij}}^2) + \alpha \text{MAD}_i(D_{a_{ij}}^2)]$  starts reaching saturation point as the *MAD* (the median of absolute deviations) gets further from the median *med* for all distances in cluster *i*.  $\psi(x)$  is a monotonically nonincreasing weight function. See [25] for more implementation details of robust estimators.

Computing a derivative  $\frac{\partial F}{\partial w_{ij}}$  and setting it to *zero* we obtain the update equation for membership weights:

$$w_{ij} = \frac{1/\rho(D_{a_{ij}}^2)}{\sum_{n=1}^C 1/\rho(D_{a_{ij}}^2)} + \frac{\alpha(|a_i| - \overline{|a_i|})}{\rho(D_{a_{ij}}^2)} \quad (5.9)$$

Where  $|a_i| = \sum_{j=1}^N w_{ij} \psi(D_{a_{ij}}^2)$  is a *robust cardinality* of cluster  $a_i$  and

$$\overline{|a_n|} = \frac{\sum_{i=1}^C |a_n| / \rho(D_{a_{in}}^2)}{\sum_{i=1}^C 1/\rho(D_{a_{in}}^2)} \quad (5.10)$$

is the weighted average of robust cardinalities for all clusters.

Also,  $\alpha$  is a parameter controlling the speed of agglomeration (see [25] for details).

Then, given the objective function in Equation (5.8), the solution to the optimization is computed by Newton's method described in Algorithm 2.

Special care has to be taken during the initialization stage. One undesired side-effect of fuzzy

```

initialize clusters;
repeat
  for  $i \leftarrow 1$  to  $C$ ,  $j \leftarrow 1$  to  $N$  do
    | compute  $D_{a_{ij}}^2$ ;
  end
  for  $i \leftarrow 1$  to  $C$  do
    | estimate  $med_i$  and  $MAD_i$ ;
  end
  for  $i \leftarrow 1$  to  $C$ ,  $j \leftarrow 1$  to  $N$  do
    | update  $\rho(D_{a_{ij}}^2)$  and  $\psi(D_{a_{ij}}^2)$ ;
  end
  for  $i \leftarrow 1$  to  $C$  do
    | compute  $|a_i|$  and  $\overline{|a_i|}$ ;
  end
  for  $i \leftarrow 1$  to  $C$ ,  $j \leftarrow 1$  to  $N$  do
    | update weights  $w_{ij}$  according to eq. (5.9);
  end
  for  $i \leftarrow 1$  to  $C$  do
    | update number of clusters  $N$ ;
  end
until convergence of  $F(A, W; E)$  ;

```

**Algorithm 2:** Fuzzy Clustering of Swarming Events

clustering can be that the cluster with a higher data-density attracts all points from other, less "heavy" clusters (*i.e.* clusters with lower cardinality) [44].

## 5.5 Benchmarking Against Single Step Activity Detectors

To detect groups of shoppers we tried a method of clustering motion trajectories by treating each path as a time series and computing a special measure based on spatio-temporal matching [48].

The outline of this method is provided here. As a one tier approach it does not have the potential to cope with activities based on multiple events during long periods of time (see section 6.2)

Therefore, we assume that people who shop together can be identified by the following criteria: they enter the scene together, leave the scene together, have a small mean intra-group distance, have a small mean difference between paths.

When comparing two trajectories as signals, there are two important aspects: time shift between two signals and the signal shape. To elegantly incorporate both of these considerations and to account for all empirically established criteria we have created the proximity metric based on the combination Euclidean distance and the shape distance of two signals.

$$f_{ij}(T) = \int \left[ (x_i(t) - x_j(t+T))^2 + (y_i(t) - y_j(t+T))^2 \right] dt \quad (5.11)$$

$$d_{ij} = \int f_{ij}(T) \times N(0, \sigma^2) dT \quad (5.12)$$

If the time  $t$  is discrete, as it is in our case with each measurement corresponding to a single video frame, the equations above can be rewritten as:

$$d_{ij} = \sum_{T=-\Delta}^{\Delta} \left( \sum_{t=t_1+T}^{t=t_2-T} [(x_i(t) - x_j(t+T))^2 + (y_i(t) - y_j(t+T))^2] \right) \times \frac{N(0, \sigma^2)}{t_2 - t_1} \quad (5.13)$$

Thus the distance  $d_{ij}$  between two trajectories  $d_{ij}$  is the weighted sum of trajectory proximities at each time moment. The interval  $[-\Delta, \Delta]$  is a time cutoff that can reduce the computation time.

The standard deviations of normally distributed weights can be increased to account for higher time spread between people in the same group. We have chosen  $\Delta = 3\sigma$ .

The integration is started at time  $t_1$  when both objects are visible in the scene for the first time and end it at time  $t_2$ , when at least one object has left the scene. The interval  $[t_1, t_2]$  is sometimes referred to as the *longest common subsequence* in the literature. Since there must be no favoring of shorter or longer paths, the distance measure is normalized by the length of the traveled interval  $t_2 - t_1$ .

Because people in a shopping group are not guaranteed to either appear or leave the scene at the same time, we propose to compute the trajectory similarity measure on piecewise uninterrupted segments — that is the intervals of time where both bodies in question were present and were successfully tracked in the scene.

Naturally, we do not compute the distance between pairs with very small longest common sub-trajectories because these will not result in a statistically significant measure. The cutoff for the common subsequence length was chosen empirically at 3 seconds, or 45 frames (at 15 fps).

Furthermore, for computational efficiency, we sub-sample the trajectories. This is justified because the physical location of a person does not change significantly within a time interval of 10 frames ( $< 1$  sec). Once the pairwise distances for all trajectories are known, as our next step the paths are clustered based on this measure.

We apply an agglomerative hierarchical clustering, where each object is initially placed into its

own cluster  $C$ . Therefore, if we have  $N$  objects to cluster, we start with  $N$  singleton groups.

The clustering requires a distance threshold to be specified. Once this is done, the procedure is as follows:

1. Compare all pairs of groups and mark the pair that is closest.
2. The distance between this closest pair of groups is compared to the threshold value.
  - (a) If the distance between this closest pair is less than the threshold distance, these groups become linked and are merged into a single group. Return to Step 1 to continue the clustering.
  - (b) If the distance between the closest pair is greater than the threshold, the clustering stops.

If the threshold value is too small, there will still be many groups present at the end, and many of them will be singletons. Conversely, if the threshold is too large, objects that are not very similar may end up in the same cluster. The threshold value was determined empirically from video sequences of varying complexity.

When merging two clusters, the center point of the new cluster at each frame  $C'$  is determined as a weighted average of two paths corresponding to the centers of the merged clusters

$$C'_t = C_t^1 \cdot |C^1| + C_t^2 \cdot |C^2| \quad (5.14)$$

Where  $C_t$  is the location along the path of the group  $C$  at time  $t$  and  $|C|$  is the number of the trajectories belonging to the cluster with center in  $C$ .

## Experimental Results

The performance of the algorithm described in Chapter 3 was evaluated by running it on a set of video sequences recorded with a perspective projection camera. Activity recognition was tested on a different dataset that has been collected in a real retail store and then annotated manually by domain experts to provide the ground truth.

### 6.1 Visual Tracking Results

We tested the low-level parts of our tracking system on a number video sequences from two different cameras mounted in a retail store chain and on the publicly available CAVIAR dataset [6]. Some sample frames and results of the head candidates detection as well as height estimation from the test video sequences are presented in figure 6.1.

One of the most frequent cases of detecting false positives was occurring when not enough frames were allotted for the background acquisition and, consequently, some people standing were interpreted as part of the background. Once these people later moved, not only the moving person but the pixels where she used to stand were detected as foreground objects. The background subtraction approach has given good results even under extreme lighting conditions (see (i) and (j) in Figure 6.1).

The falsely detected head locations, were primarily the video compression artifacts influencing the background subtraction process. Nevertheless, the algorithm shows robust performance with significant levels of illumination noise, under the low-quality, real-life capturing conditions.

The false negative head candidates had two primary causes. First, parts of the foreground region become separated from the body or sometimes a part of the shadow is considered as a separate body, and this causes a false candidate to be detected (see (a) in Figure 6.3). We believe that human shape modeling will solve this problem. A second factor, one that badly influences the detection, is when the heads are not pronounced enough to create a local maximum in the histogram (see (b) in figure 6.3). This problem can be solved in the future by color and texture analysis within the blob.

To partially evaluate the quality of the results, we have analyzed a number of detected head candidates in the sequences with two people that were detected as a single blob (Figure 6.2). The evaluation shows that the outputs from our methods can be used at the initialization stage of the tracking algorithm. To further evaluate the quality of our method candidate hit/miss and average error analysis based on their coordinates is required.

We performed preliminary evaluation of our tracking system for the presence of three major types of inconsistencies: misses, false hits, and identity switches. A *miss* is when the body is not detected or it is detected but tracked for an insignificant portion of its path ( $< 30\%$ ). A *false hit* is when a new body is created where there is no actual person present. Most of the false hits are a result of more than one body in the model being assigned to a single body in the scene. An *identity switch* is when two or more bodies exchange their IDs once within the close proximity from each other. By visually counting the number of each of types of errors on a number of sequences of

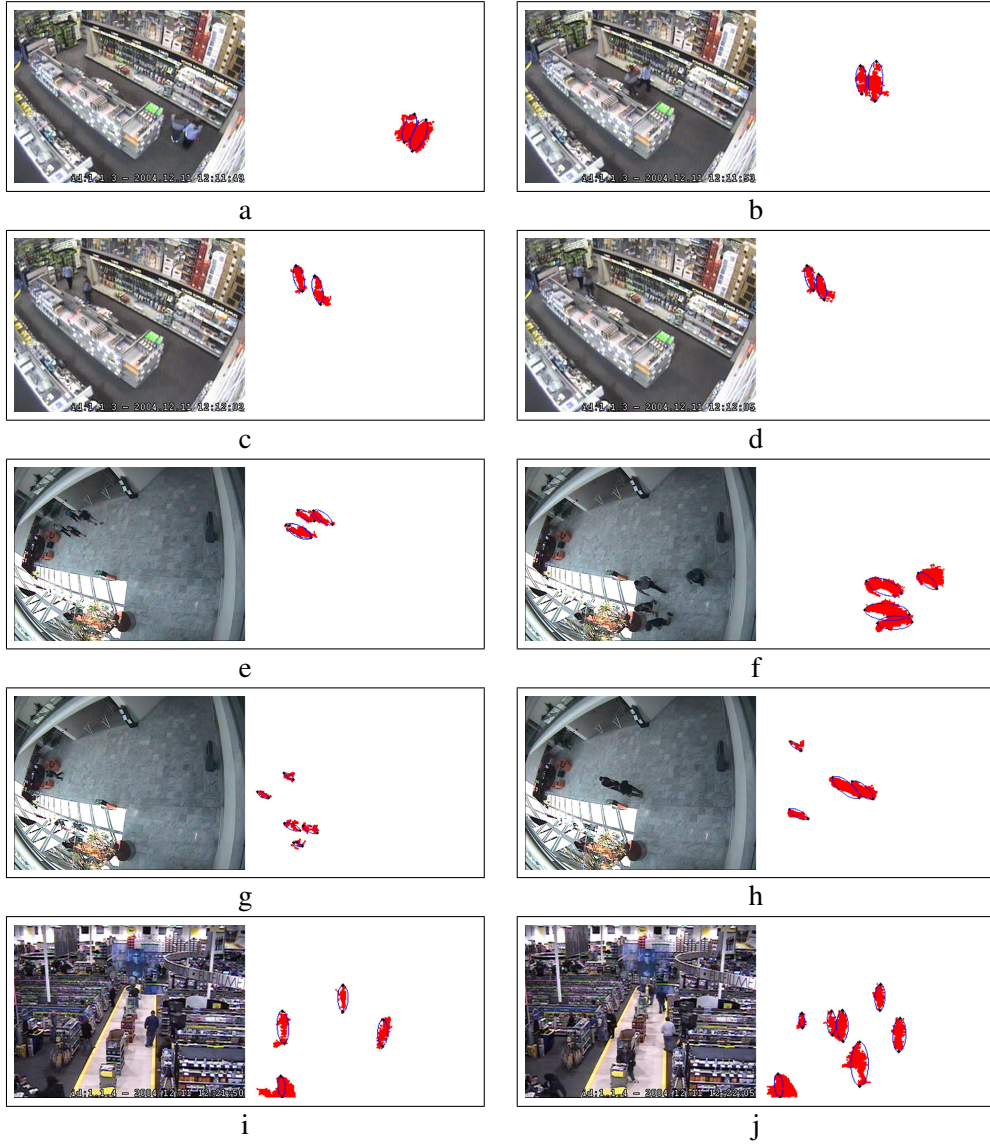


Figure 6.1: (a) - (j) Head candidates from test frames. Left image is the original frame. In the right image "red" represents foreground mask, small black dots indicate the locations of  $T_i$  and  $B_i$ ; blue ellipses are fitted with  $T_i B_i$  as the major axis

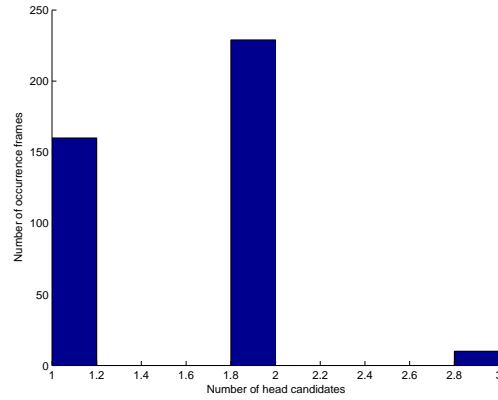


Figure 6.2: Head detection algorithm performance evaluation. This graph shows the number of frames when 1, 2, or 3 heads were detected. The true number of heads is 2.



Figure 6.3: (a,b) Incorrect head detection from test frames. Left image is the original frame. On the right image red represents foreground mask, small black dots indicate the locations of  $T_i$  and  $B_i$ ; blue ellipses are fitted with  $T_i B_i$  as the major axis; (c,d) Illustrate height detection: brown plates contain height mean and variance for each ellipse

overall 6000 frames (about 400 seconds) we have obtained results summarized in Table 6.1. Note that these sequences were taken from the OTCBVS color-thermal dataset [19] and the background subtraction was done by using both RGB and thermal information.

Seq	Ppl	$P^-$	$P^+$	$P^{+/-}$
1	15	3	1	3
2	8	0	0	0
3	16*	0	1	2
4	3**	0	0	0
5	2	0	0	0
6	4	0	0	0
ALL	48	3	2	5
%	100	12.5	4.1	10.4

Table 6.1: Tracking results for projection camera model, based on the manually observed ground truth (\* - two infants, below the tracked height limit, lead tracker to some confusion; \*\* - 2 pedestrian covered by trees not counted).  $P^-$  indicates missed people,  $P^+$  indicates falsely detected people (primarily due to shading artifacts) and  $P^{+/-}$  indicates two pedestrian IDs swapping

The most common mistakes made by the tracker, were false hits. We have observed that the majority of false hits (more than 50%) are short lived, *i.e.* they typically last for only several frames. These cases can be further post-processed by temporal filtering to remove insignificantly short paths. Sometimes, however, false detections are accompanied by ID switches, when a body that is tracked for a long time is substituted for a false hit. This presents a more complicated problem and deserves further study.

Overall performance of the tracker is promising, primarily because it produces satisfactory detection and prolonged tracking in crowded scenes. The output from our tracking module serves as a reliable basis for obtaining customer paths (Figure 6.4) and the detection of shopper groups (Figure 6.5).

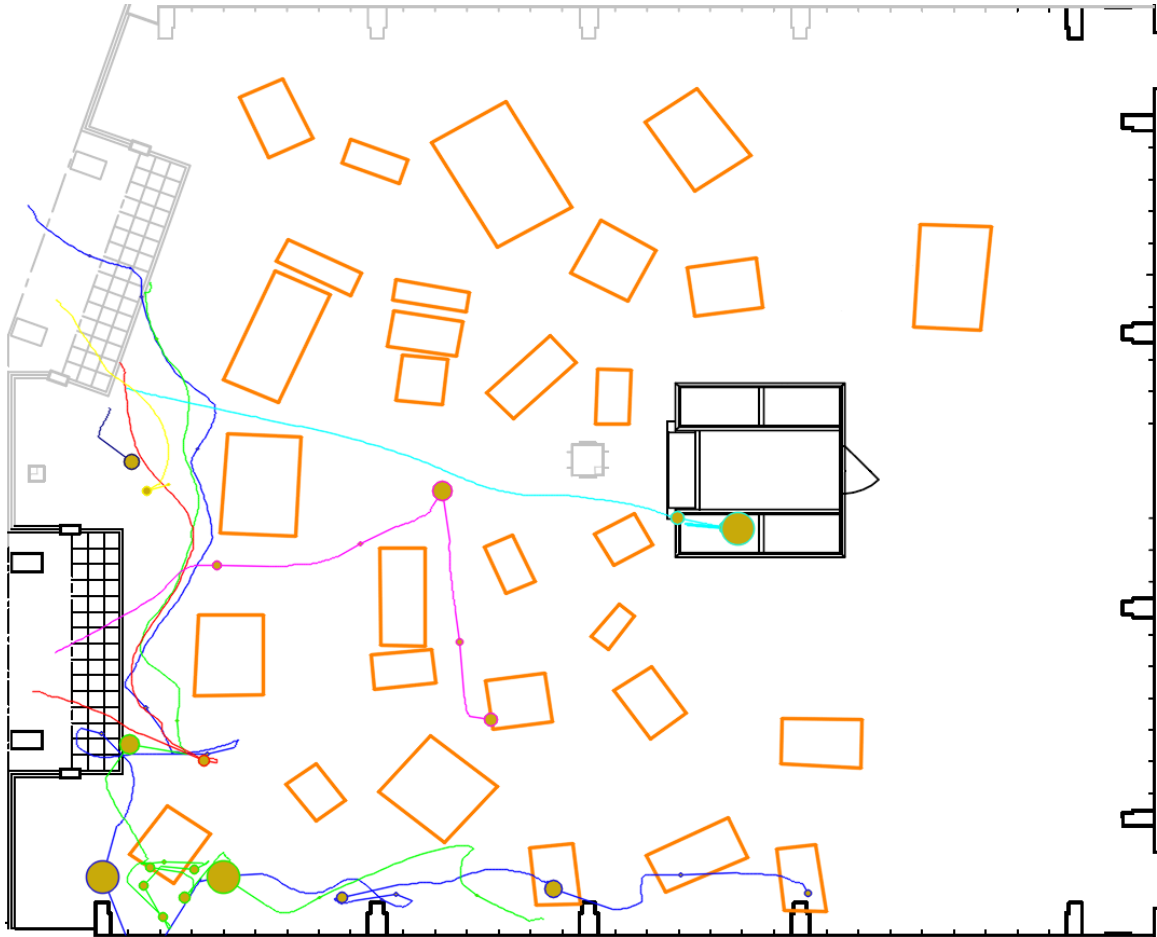


Figure 6.4: Customer paths marked on the floor map (circles represent dwell-locations, with the size of a circle proportional to dwell time)



Figure 6.5: Select frames showing the detection of shopping groups (marked as white boxes)

## 6.2 Activity Recognition Results

To test our method for activity recognition we used three tracking sequences recorded with the panoramic camera in an apparel retail store, each sequence being one hour long. These sequences were annotated manually by domain experts to obtain the assisted shopping groups markup as ground truth. The total number of customers appearing in the scene in these three hours is 245 and the actual number of shopper groups is 50 with 111 of the customers in groups. Of these groups, 7 were composed of three people and 2 composed of four people, the rest were two-customer groups. We decided to exclude from consideration the groups formed earlier than 5 minutes prior to the sequence end as well as customers who were at the store at the beginning and left in the first 5 minutes of tracking, since tracking information was on average less than 20% complete for such tracks. Several store employees were also excluded from the results, to avoid counting events such as a customer seeking assistance, conversing with an employee over a long period of time or on several consequent occasions.

Tables 6.2 and 6.3 show the total number of groups present in each scene (from the ground truth dataset), the percentages of correctly identified shopper groups as well as false positives (groups detected where none were present) and false negatives (missed groups).

Sequence	Groups	$P^+$	$P^-$	Partial
1	20	0	7	0
2	17	1	3	1
3	17	0	7	0
Total	54	1	17	4
Percent	100	1.8	31.5	7.4

Table 6.2: Activity detection validation results against manually observed ground truth (Clusters with 5 or more contributing events are considered valid. Event sampling frequency: every 30 frames.  $P^+$  are false groups not present in the scene,  $P^-$  are missed groups, and partially detected groups have at least 2 actors correctly identified)

These sequences represent one hour of typical store traffic on three different days taken from 4PM to 5PM. It is interesting to observe that the correct detection rates are higher for the first

Sequence	Groups	$P^+$	$P^-$	Partial
1	20	0	7	0
2	17	1	3	1
3	17	0	7	0
Total	54	1	12	2
Percent	100	1.8	22.2	3.7

Table 6.3: Activity detection validation results against manually observed ground truth (Clusters with 5 or more contributing events are considered valid. Event sampling frequency: every 10 frames)

sequence, which also contained some of the heaviest traffic. We conclude from this that the performance accuracy of our group detector is proportional to the length of the tracks involved. Average store visit for a group can range anywhere from 5 to 15 minutes, which provided significant length of tracking data for our two stage clustering unit in most cases.

In general, path length linearly influences the accuracy of the clustering algorithm, with more events generated over a longer period of time, resulting in a higher detection rate. False positives are primarily due to the behaviors exhibiting patterns similar to grouping. One such behavior was, for instance, a customer co-mingling with an unrelated customer over a long period of time or several times. This took place in highly congested areas, such as checkout register or nearby sales racks. Another major cause for mis-detection is the inadequate time during which group participants were visibly present in the scene (*i.e.* were behind the fixture).

One-way analysis of variance with the duration of activity as an independent variable and the detection confidence  $[0, 1]$  as a dependent variable reveals a presence of strong correlation ( $p^{val} \approx 10^{-11}$ ).

We conclude that simple spatio-temporal grouping of actors is not enough for higher level group activity recognition as the groups might form coincidentally. Spatial analysis of simple grouping events provides an increased accuracy for group activity recognition. The essential characteristic of our swarming event data is the presence of outliers as well as the fuzzy character of activity memberships. Some of the swarming events happen coincidentally, due to crowding effects in the

store. Figure 6.9 gives an example of such coincidental grouping.

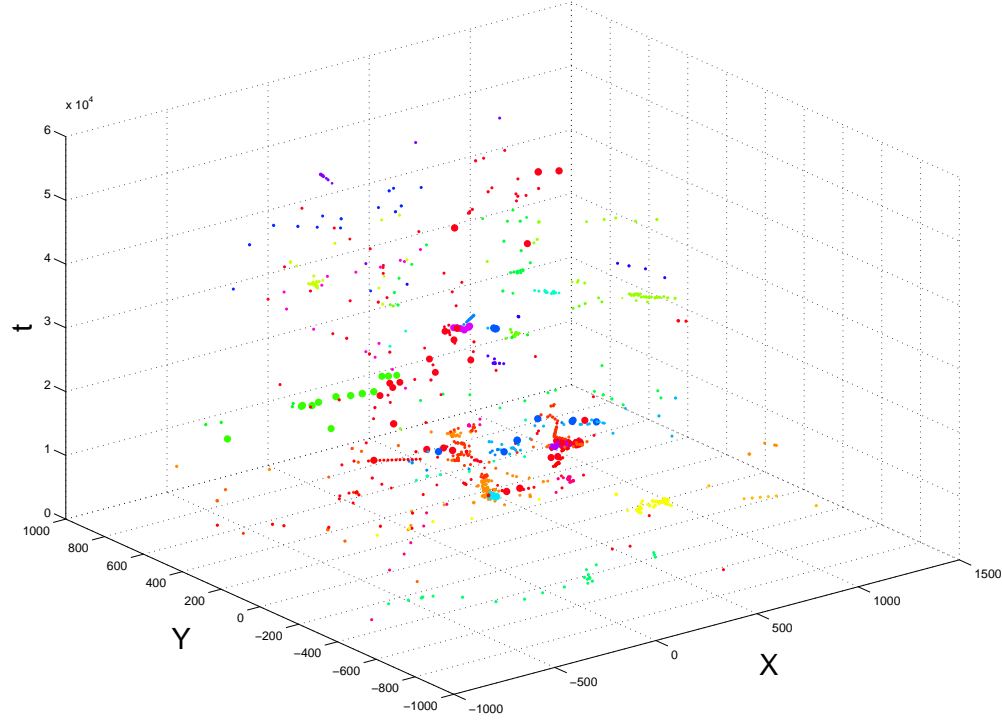


Figure 6.6: Swarming events in space-time (axis  $t$  represents frame number of event;  $X$ ,  $Y$  - average event location on the floor plane) Dot size corresponds to the validity of event (5.3). Dots of matching color belong to the same activity

We compared two-level activity recognition results to a simple one level time-space convolution technique from Section 5.5. Due to its design, the simple detector was not able to handle any of the complex activities, *i.e.* the ones consisting of multiple events separated in time by at least one episode when the actors are apart in space. The approximate duration of the activities on which it worked was in the order of 60 seconds. Therefore we conclude that our two-level design is necessary for the detection of in-store grouping activities.

Another interesting observation we made while comparing groups detected by the system to the manually annotated ground truth is how easy it is to miss such activity for a human observer. In

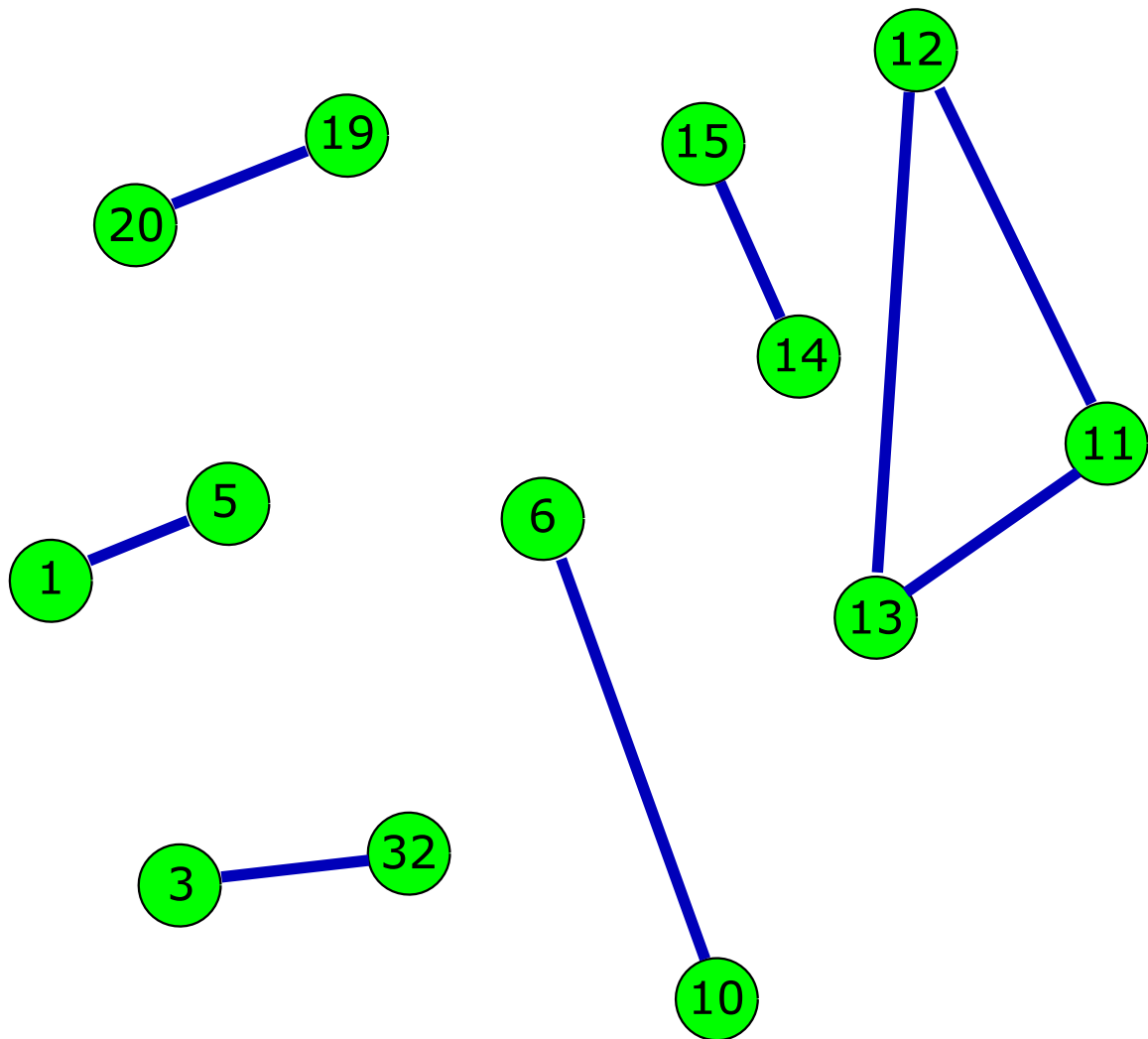


Figure 6.7: Shopping groups from sequence 2 (first 15 minutes) automatically arranged by proximity metric. The lengths of the edges connecting numbered nodes, correspond to the distance in non-Euclidean space calculated per Equation (5.7)

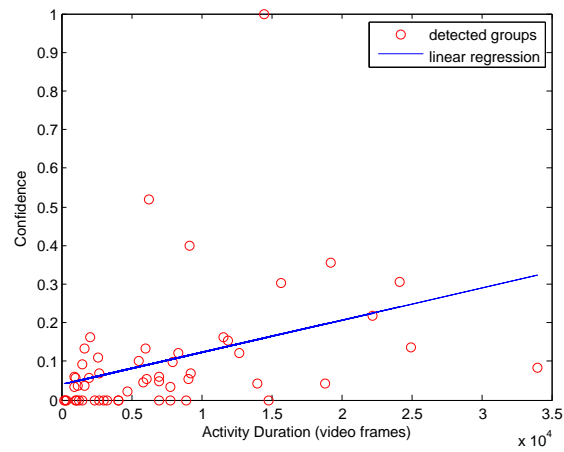


Figure 6.8: Group detection accuracy as a function of path duration (path length is counted as the duration of the visible part of the tracks when at least one of the group members was present in the scene)



Figure 6.9: (1a) Two actors form a real shopper group (marked with solid blue) (1b) One of the actors from 1a in a coincidental event (marked with zebra yellow) (2a) Passer-by temporarily increases group cardinality to three (2b) Passer-by walking away

---

three hours of video the system detected 6 groups that were not originally spotted by the operator, consequently ground truth had to be adjusted to include missing observations (results in tables 6.2 and 6.3 are given in comparison to the corrected ground truth). This leads us to believe that the system provides a valuable tool for aiding and potentially completely eliminating manual group activity labelling in video sequences.

## Conclusion

### 7.1 Contributions

In this work we presented a method for customer tracking in crowded environments. We used automatically obtained tracks to extract customer group information from several hours of retail store video recordings. The method presented in this thesis computes for each customer a path, an approximate size of the body, a list of dwell locations and the orientation of travel.

The tracking system presented starts with the new noise-resistant background subtraction technique followed by the *vanishing point projection* head candidate search algorithm to handle initialization. Temporal tracking is organized as a Markov chain optimizing Monte Carlo sampling, with the state of each body, the number of bodies, knowledge of static occluders and doorways all incorporated into a single model. By utilizing information about the tracking environment (such as human heights and entrance/exit locations) in the form of prior probabilities we are able to increase the tracking accuracy in relation to the appearance-only based trackers.

The activity recognition part of this work constitutes a significant contribution to the field of automated behavior recognition. Group behaviors, as formulated in this thesis, yet have not been addressed in the related literature. We show how to build a two-layer system of detecting grouping activities, by using shorter grouping events as the building blocks. For finding activities we

demonstrated how to use deterministic and fuzzy variants of hierarchical clustering techniques. The reduction of the influence of the outliers in clustering is achieved by introducing robust influence functions into the group distance metric.

To summarize, both in tracking and activity recognition we introduced a number of novel methods:

- **Adaptive codebook background model:** To eliminate the influence of light changes and camera noise as well as to make the tracker perform well in the crowded environment we introduced a codebook based background model which is capable of re-sampling the multi-modal structure of the color of each pixel on-line. (Section 4.1.2). We demonstrated how this model can be used in a combined tracking scenario with color and thermal sensors.
- **Variable system state dimensionality:** Our system uses the reversible Monte Carlo sampling method to model and estimate the state of each pedestrian (Section 4.2.1). The dimensionality of the system was incorporated into the system as one of its variables.
- **Initialization ambiguity:** The blob-tracker initialization suffers when more than one object merge to form one blob. In this work we used the analysis of the blob outline to find head candidates as the peaks in the vanishing point projection histogram. The VPP is a newly introduced alternative to a vertical projection histogram that allows to incorporate camera model to get a less distorted projection (Section 4.1.3).
- **Mean shift:** Extended the mean-shift algorithm [14] by introducing the distance weight plane, z-buffer and including the foreground map into the histogram computation (Section 4.2.7 and 4.2.6).
- **Prolonged tracking:** The system is able to perform robust tracking through illumination changes and occlusions due to the use of obstacle maps and the recovery of recently deleted objects incorporated as one of the mutation in MCMC method (Section 4.2.6).

- **Formalizing high level group activities:** In this thesis we proposed a generalized event-activity based framework for automatic human group behavior categorization. The idea is borrowed from the intelligent swarms field, where it is used in a generative sense, to generate the next state of the simulated process, and re-applied in a discriminative way, to measure distance between various types of grouping behaviors. Each customer is treated as a member of the swarm, acting based on a set of simple, well-defined rules. We demonstrate how to define such rules, based on the type of swarming activity one intends to detect. The idea is illustrated for the customer grouping activity in retail store (Section 5.1).
- **Robustly handling group data with outliers:** Here we applied fuzzy clustering methods with robust metrics to reduce the influence of outliers. This approach proved to be extremely suitable for the problem of hierarchically clustering swarming events, with many of the events, being due to the noise (customers coincidentally showing group characteristics) and often events contributing to more than one activity (Section 5.4).

## 7.2 Future Work

Beyond the scope of this thesis we intend to continue research in this direction. Below is the list of particular problems we would like to address:

- **Improved Tracking:** We have not explored one class of features for human body tracking — so-called *feature points*. A feature point is usually a small area in the image that contains a significant amount of information about the object, *e.g.* an edge or a corner. Using such well-established techniques as SIFT descriptors [50] has a promise of increasing tracking accuracy, while reducing the computational load.
- **Detect Other Types of Swarming Activities:** Using the general framework presented in Section 5.1, but modifying the distance equations we plan to detect several other swarming

behaviors in the retail stores. These include but are not limited to estimation of length of the checkout line and detecting customer-employee interactions. Also, the detection of dwelling state is to be automated by learning two of the method's parameters, dwell time threshold and dwell area radius, through supervised learning using training video sequences.

- **Demographical analysis:** The color histogram information and facial feature descriptors recorded for each customer can potentially be used to determine person's age, gender or ethnicity.
- **Sensor Fusion:** Detailed information about product interaction can be achieved by fusing the results of visual tracking with RFID based positioning of certain merchandise items. This way, when in a close proximity to a specific item, the customer can be categorized as interacting with it if the visual cues support this hypothesis (*e.g.* hands are reaching out towards the fixture).
- **Security:** A combination of tracking and activity recognition can be used for theft detection. If "picking up an object" event is detected and not followed by "checkout" event, the probability of "shoplifting" behavior is increased.
- **Visualization:** An important aspect related to the vision techniques presented in this work is the visualization of customer trajectories and various aggregate marketing statistics (dwell time, traffic density maps, etc). The ultimate plan is to have a visual representation of the entire 3D model of the store with various layers of information embedded. This presents a potentially valuable tool of conveying quantitative tracking and AR results to the business community.

Using the presented general framework for automated recognition of swarming activities as the next steps in activity recognition we expect to recover the following quantitative measurements: identifying queue lengths and queue wait times, building store traffic heat maps, aisle penetration

maps and estimating customer conversion rate, *i.e.* the number of people making purchases in relation to the total number of customers in the store.

We plan to extensively validate the accuracy of the group detection algorithm using the manually marked dataset of more than 30000 frames provided by CAVIAR project [6]. Although the evaluation of the tracking subsystem shows promising results, the author is aware that a more formal evaluation has to be performed for each of the customer activity characteristics.

Another potential improvement is to enhance the quality of our depth maps using a 3D CAD model of the store, which we expect to result in highly stable tracking due to improved handling of scene fixture occlusions. Such models, currently under development, will incorporate the layout of the store fixtures, product placement and camera location.

We are currently investigating the use of the visual tracker in combination with the customer counting camera installed at the entrance that would impose an additional constraint on our system by providing an exact number of the bodies in the store. This method could facilitate uninterrupted tracking of each customer for longer periods of time, which can be further used to compute the percentage customer distribution in the different areas of the store and provide important clues into the “conversion rate” analysis (the ratio of the amount of purchases to the total number of customers) more reliably. With the increased quality of captured video we hope to get enough detail to perform an analysis of certain product interaction aspects: attention (*i.e.* turning the torso towards the product, or squatting/reaching for the product), browsing (when hands are performing “reaching out” gestures).

In the future, the position and orientation of body ellipsoid can be combined with multiple-view color representation for more reliable color tracking [45]. We believe that this kind of tracking will provide information for customer attention analysis, such as rough estimations of customer gaze center or interactions within customer groups [28].

Another potential improvement in tracking can come from the use of multiple cameras with overlapping fields of view. Such a setup, while difficult to implement, can provide additional disambiguation in situations when two people cross paths. Combining the views of two or more cameras can help make better decisions about the locations of the bodies.

The tracking currently is performed under the assumption that all moving objects are human. Future direction that we are researching is using features specific to human pedestrians, such as periodic gait signature [47] to differentiate people from shopping carts and other moving structures. One potential improvement also lies in using multi-modally distributed priors on human height, to successfully track children.

# Bibliography

- [1] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.
- [2] Aaron Bobick and Yuri Ivanov. Action recognition using probabilistic parsing. In *International Conference on Computer Vision and Pattern Recognition*, pages 196–202, Washington, DC, USA, 1998. IEEE Computer Society.
- [3] Francois Boutin and Mountaz Hascoet. Cluster validity indices for graph partitioning. In *International Conference on Information Visualisation*, pages 376–381, Washington, DC, USA, 2004. IEEE Computer Society.
- [4] Raymond Burke. The third wave of marketing intelligence. In Manfred Krafft and Murali Mantrala, editors, *Retailing in the 21st Century*. Springer-Verlag, 2005.
- [5] Dan Buzan, Stan Sclaroff, and George Kollios. Extraction and clustering of motion trajectories in video. In *International Conference on Pattern Recognition*, pages 521–524, Washington, DC, USA, 2004. IEEE Computer Society.
- [6] CAVIAR. Ist 37540. Found at <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>, 2001.
- [7] Siddhartha Chib and Edward Greenberg. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

- [8] Amit Chilgunde, Pankaj Kumar, Surendra Ranganath, and Huang WeiMin. Multi-camera target tracking in blind regions of cameras with non-overlapping fields of view. In *British Machine Vision Conference*, pages –, 2004.
- [9] R Cipolla, Tom Drummond, and D Robertson. Camera calibration from vanishing points in images of architectural scenes. In *British Machine Vision Conference*, volume II, pages 382–392, 1999.
- [10] Maurice Clerc. *Particle Swarm Optimization*. ISTE Publishing Company, 2006.
- [11] Robert Collins, Alan Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt, and Lambert Wixson. A system for video surveillance and monitoring. Technical Report CMU-RI-TR-00-12, Carnegie Mellon University, Pittsburgh, PA, May 2000.
- [12] Robert Collins and R. Weiss. Vanishing point calculation as a statistical inference on the unit sphere. In *International Conference on Computer Vision*, pages 400–403, December 1990.
- [13] R.T. Collins, A.J. Lipton, and T. Kanade. Introduction to the special section on video surveillance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):745–746, 2000.
- [14] Dorin Comaniciu, Peter Meer, and Visvanathan Ramesh. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–575, 2003.
- [15] Antonio Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 1999.
- [16] Antonio Criminisi, Andrew Zisserman, Luc Van Gool, and Simon Bramble. A new approach to obtain height measurements from video. In *Proceedings of The International Society for Optical Engineering*, pages 227–238, 1998.

- [17] Rita Cucchiara, Costantino Grana, Massimo Piccardi, and Andrea Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, 2003.
- [18] D.L. Davies and D.W. Bouldin. A cluster separation measure. *International Journal of Pattern Recognition and Artificial Intelligence*, 1 (2):224–227, 1979.
- [19] James Davis and Vinay Sharma. Fusion-based background-subtraction using contour saliency. In *International Conference on Computer Vision and Pattern Recognition*, pages III: 11–11, 2005.
- [20] Hannah Dee and David Hogg. Detecting inexplicable behaviour. In *British Machine Vision Conference*, pages 477–486, 2004.
- [21] J Deutscher, B North, B Bascle, and Blake. Tracking through singularities and discontinuities by random sampling. In *International Conference on Computer Vision*, page 1144, Washington, DC, USA, 1999. IEEE Computer Society.
- [22] Arnaud Doucet, Nando de Freitas, and Neil Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.
- [23] Ahmed Elgammal and Larry Davis. Probabilistic framework for segmenting people under occlusion. In *International Conference on Computer Vision*, pages II: 145–152, 2001.
- [24] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.
- [25] Hichem Frigui and Raghu Krishnapuram. A robust competitive clustering algorithm with applications in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):450–465, May 1999.
- [26] P. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732, 1995.

- 
- [27] Frank R. Hampel, Elvezio M. Ronchetti, Peter J. Rousseeuw, and Stahel Werner A. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons Ltd, 1986.
- [28] Ismail Haritaoglu and Myron Flickner. Detection and tracking of shopping groups in stores. In *International Conference on Computer Vision and Pattern Recognition*, pages I:431–438, 2001.
- [29] Ismail Haritaoglu, David Harwood, and Larry Davis. W-4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.
- [30] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2003.
- [31] Laszlo Havasi and Tamas Sziranyi. Motion tracking through grouped transient feature points. In *Advanced Concepts for Intelligent Vision Systems*, 2004.
- [32] David C. Hoaglin, Ferderick Mosteller, and John W. Tukey. *Understanding Robust and Exploratory Data Analysis*. John Wiley & Sons Ltd, 1983.
- [33] Somboon Hongeng, Ram Nevatia, and Francois Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *Comput. Vis. Image Underst.*, 96(2):129–162, 2004.
- [34] Thanarat Horprasert, David Harwood, and Larry Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *International Conference on Computer Vision*, pages 1–19, 1999.
- [35] Sam K. Hui, Peter Fader, and Eric Bradlow. Path data in marketing: An integrative framework and prospectus for model-building. *Social Science Research Network*, 2007.
- [36] Point Grey Research Incorporated. Ladybug 2 tm camera, <http://www.ptgrey.com>, 2006.

- [37] Michael Isard and Andrew Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
- [38] Michael Isard and John MacCormick. Bramble: A bayesian multiple-blob tracker. In *International Conference on Computer Vision*, volume 2, pages 34–41, 2001.
- [39] Steven Johnson. *Emergence: The Connected Lives of Ants, Brains, Cities, and Software*. Scribner, 2001.
- [40] Christopher Kemp and Tom Drummond. Multi-modal tracking using texture changes. In *British Machine Vision Conference*, 2004.
- [41] Sohaib Khan and Mubarak Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1355–1360, 2003.
- [42] Zia Khan. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(11):1805–1918, 2005.
- [43] Kyungnam Kim, Thanarat Chalidabhongse, David Harwood, and Larry Davis. Background modeling and subtraction by codebook construction. In *International Conference on Image Processing*, volume 5, pages 3061– 3064, 2004.
- [44] Frank Klawonn and Frank Höppner. What is fuzzy about fuzzy clustering? understanding and improving the concept of the fuzzifier. In *Advances in Intelligent Data Analysis V*, volume 2810 of *LNCS*, pages 254–264, 2003.
- [45] Alex Leykin, Florin Cutzu, and Mihran Tuceryan. Using multiple views to resolve human body tracking ambiguities. In *British Machine Vision Conference*, 2004.
- [46] Alex Leykin and Riad Hammoud. Robust multi-pedestrian tracking in thermal-visible surveillance videos. *Object Tracking and Classification in and Beyond the Visible Spectrum Workshop at the International Conference on Computer Vision and Pattern Recognition*, 0:136, 2006.

- [47] Alex Leykin, Yang Ran, and Riad Hammoud. Thermal-visible video fusion for moving target tracking and pedestrian classification. In *Object Tracking and Classification in and Beyond the Visible Spectrum Workshop at the International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [48] Alex Leykin and Mihran Tuceryan. Tracking and activity analysis in retail environments. Technical Report 620, Indiana University, 2005.
- [49] Alex Leykin and Mihran Tuceryan. A vision system for automated customer tracking for marketing analysis: Low level feature extraction. In *Human Activity Recognition and Modelling Workshop*, 2005.
- [50] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.
- [51] S. Maybank and T. Tan. Introduction to special section on visual surveillance. *International Journal of Computer Vision*, 37(2):173–173, 2000.
- [52] Anurag Mittal and Larry Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In *European Conference on Computer Vision*, volume 51, pages 189–203, 2002.
- [53] Thomas B. Moeslund, Adrian Hilton, and Volker Kröger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, November 2006.
- [54] Vlad I. Morariu and Octavia I. Camps. Modeling correspondences for multi-camera tracking using nonlinear manifold learning and target dynamics. In *International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 545–552, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

- [55] Daniel Morris and James Rehg. Singularity analysis for articulated object tracking. In *International Conference on Computer Vision and Pattern Recognition*, page 289, Washington, DC, USA, 1998. IEEE Computer Society.
- [56] Nuria Oliver, Barbara Rosario, and Alex Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.
- [57] Craig Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *SIGGRAPH* 87, pages 25–34, New York, NY, USA, 1987. ACM Press.
- [58] Craig Reynolds. Steering behaviors for autonomous characters. In *Game Developers Conference*, 1999.
- [59] Barbara Rosario, Nuria Oliver, and Alex Pentland. A synthetic agent system for bayesian modeling of human interactions. In *AGENTS '99: Proceedings of the third annual conference on Autonomous Agents*, pages 342–343, New York, NY, USA, 1999. ACM Press.
- [60] Hedvig Sidenbladh, Michael Black, and Leonid Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *European Conference on Computer Vision*, volume 1, pages 784–800, 2002.
- [61] Cristian Sminchisescu and Bill Triggs. Kinematic jump processes for monocular 3d human tracking. In *International Conference on Computer Vision and Pattern Recognition*, volume 01, page 69, Los Alamitos, CA, USA, 2003. IEEE Computer Society.
- [62] Jonathan Starck and Adrian Hilton. Model-based multiple view reconstruction of people. In *International Conference on Computer Vision*, volume 02, page 915, Los Alamitos, CA, USA, 2003. IEEE Computer Society.
- [63] Chris Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking.

- In *International Conference on Computer Vision and Pattern Recognition*, volume 2, page 252, 1999.
- [64] Hai Tao, Harpreet S. Sawhney, and Rakesh Kumar. A sampling algorithm for tracking multiple objects. In *Workshop on Vision Algorithms*, pages 53–68, 1999.
- [65] Michael Wooldridge. *An Introduction to MultiAgent Systems*, 366 pages, ISBN 0-471-49691-X. John Wiley & Sons Ltd, 2002.
- [66] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.
- [67] Tao Xiang and Shaogang Gong. Beyond tracking: Modelling activity and understanding behaviour. *Int. J. Comput. Vision*, 67(1):21–51, 2006.
- [68] L. Xie, S. Chang, A. Divakaran, and H. Sun. Unsupervised discovery of multilevel statistical video structures using hierarchical hidden markov models. In *IEEE Intl. Conf. Multimedia and Expo (ICME)*, pages 29–32, Washington, DC, USA, July 2003. IEEE Computer Society.
- [69] Yaser Yacoob and Larry Davis. Learned models for estimation of rigid and articulated human motion from stationary or moving camera. *International Journal of Computer Vision*, 36(1):5–30, January 2000.
- [70] Dong Zhang, Daniel Gatica-Perez, Samy Bengio, Iain McCowan, and Guillaume Lathoud. Modeling individual and group actions in meetings: A two-layer hmm framework. In *International Conference on Computer Vision and Pattern Recognition*, volume 07, page 117, Los Alamitos, CA, USA, 2004. IEEE Computer Society.
- [71] Tao Zhao and Ram Nevatia. Tracking multiple humans in crowded environment. In *International Conference on Computer Vision and Pattern Recognition*, volume 02, pages 406–413, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

# **Appendices**

# A

---

## Spheroid Mapping

Let  $E$ , be a special case of an ellipsoid with axis  $a_3 = c$  corresponding to body height  $b_i(h)$  and axes  $a_2 = a_1 = a$ , representing the width of the body  $b_i(w)$ . Let  $\{X, Y, Z\}$  be a system of world coordinates and  $O_{cam} = \{X_c, Y_c, Z_c\}$  camera coordinates where  $X_c = 0, Y_c = 0$  and  $Z_c$  is the elevation of the camera in  $cm$ . Let  $O_{sph} = \{X_s, Y_s, Z_s\}$  denote the center of the spheroid and  $O = \{X_o = 0, Y_o = 0, Z_o = 0\}$  be the origin of the world coordinate system. Then the equation of the spheroid is given by:

$$\frac{(X - X_s)^2}{a^2} + \frac{(Y - Y_s)^2}{a^2} + \frac{(Z - Z_s)^2}{c^2} = 1 \quad (\text{A.1})$$

### A.1 Perspective Projection

A projection of an ellipsoid  $E$  onto a plane is an ellipse if  $E$  is entirely inside the field of view. Otherwise, if  $E$  is fully outside the field of view the projection is an empty set  $\emptyset$ . The process of rasterizing a spheroid thus reduces to rasterizing an ellipse, representing its projection. The perspective projection of the ellipsoid is described in greater detail in [30].

### A.2 Equirectangular Projection

In equirectangular projection coordinates the process becomes essentially a projection of an ellipsoid onto a sphere. Image coordinates are describing a sphere with  $y = \phi$  representing the latitude and  $x = \theta$  representing the longitude. Below we describe the algorithm to rasterize a vertically oriented 3D spheroid onto the image in unprojected map coordinates.

First, let's find the first and last horizontal scanlines in the image between which the projection of the spheroid is contained. In order to do that let's slice  $E$  with a plane  $O_{cam}OO_{sph}$ , to get an ellipse.

$$\frac{(x - d)^2}{a^2} + \frac{(y - b)^2}{c^2} = 1 \quad (\text{A.2})$$

In Equation (A.2),  $y = Z$  and coordinate  $x$  is collinear to  $\overrightarrow{[X_s, Y_s, 0]} - \overrightarrow{O}$ . Also,  $d = |O, \{X_s, Y_s, 0\}|$  is the distance of the floor point of the spheroid from the origin.

Let's define a family of lines originating at  $C$  over a parameter  $\phi$  as:

$$\begin{cases} x = 0 + \sin(\phi)t \\ y = Z_c - \cos(\phi)t \end{cases} \quad (\text{A.3})$$

The intersection of a line from family (A.3) with the ellipse in (A.2) will produce either zero (when the line misses the ellipse), two (when the line goes through the ellipse) or one (when the line is tangent the ellipse) solution. To establish rasterization limits we need to find the latter. By substituting  $x$  and  $y$  from (A.3) into (A.2), we get a quadratic equation:

$$\begin{cases} At^2 + Bt + C = 0 \\ A = c^2 \sin^2(\phi) + a^2 \cos^2(\phi) \\ B = -2[c^2 d \sin(\phi) + a^2(Z_c - Z_s) \cos(\phi)] \\ C = c^2(d^2 - a^2) + a^2(Z_c - Z_s)^2 \end{cases} \quad (\text{A.4})$$

In order to have a single solution the discriminant  $B^2 - 4AC$  has to be equal to zero. Solving this for  $\phi$  will give us two latitudinal angles  $\phi_1, \phi_2$  which correspond to the first and last horizontal scanlines.

As the next step we find  $\theta_1, \theta_2$  the longitudinal limits within each scanline (see Figure A.2). In order to do so, we need to find the intersection of the spheroid (A.1) and a cone (A.5).

$$\begin{cases} X = Z_c \tan(\phi) \cos(\theta)t \\ Y = Z_c \tan(\phi) \sin(\theta)t \\ Z = Z_c(1 - t) \end{cases} \quad (\text{A.5})$$

The intersection of a cone (A.5) with the ellipsoid (A.1) will produce either zero (when the cone misses the ellipse),  $\infty$  (when the cone goes through the ellipse), or one (when the cone is touches the ellipse) solution. To establish rasterization limits on  $\theta$  we need to find the latter. By substituting  $X, Y$  and  $Z$  from Equation (A.5) into Equation (A.1), we get a quadratic equation:

$$\begin{cases} At^2 + Bt + C = 0 \\ A = Z_c^2(c^2 \tan^2(\phi) + a^2) \\ B = -2Z_c[c^2 \tan(\phi)(X_s \cos(\theta) + Y_s \sin(\theta)) + a^2(Z_c - Z_s)] \\ C = c^2(X_s^2 + Y_s^2) + a^2(Z_c - Z_s)^2 - a^2c^2 \end{cases} \quad (\text{A.6})$$

If there are more than zero solutions, a positive discriminant corresponds to the points on the intersection of the cone and spheroid and a zero discriminant matches the boundaries. Setting the discriminant to zero and solving the resulting quadratic equation for  $\theta$  we obtain the limits:

$$\begin{aligned}
\theta_{1,2} &= \frac{U_0 X_s \pm \sqrt{X_s^2 Y_s^2 - U_0^2 Y_s^2 + Y_s^4}}{X_s^2 + Y_s^2} \\
U_0 &= \pm \frac{\sqrt{AC} - F}{E} \\
E &= c^2 \tan(\phi) Z_c \\
F &= a^2 d Z_c
\end{aligned} \tag{A.7}$$

Rasterization process becomes an iterative filling of horizontal lines from  $\phi_1$  to  $\phi_2$ , where within each line the pixels from  $\theta_1$  to  $\theta_2$  are inside the projection of the spheroid.

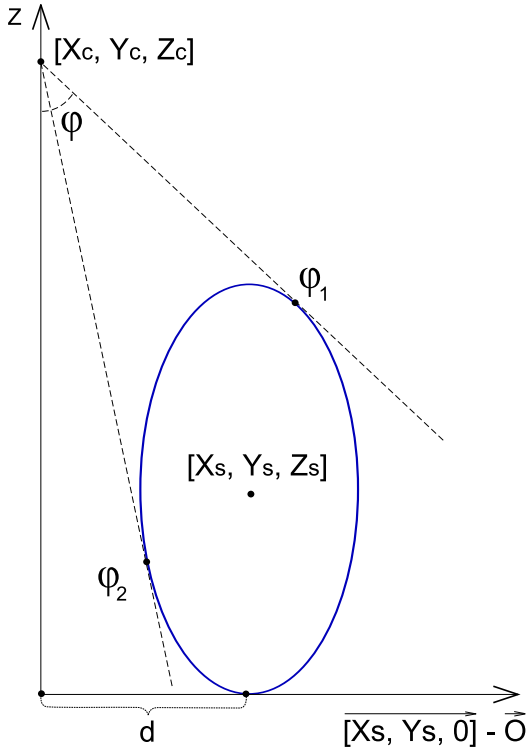


Figure A.1: Finding spheroid range in horizontal scanlines

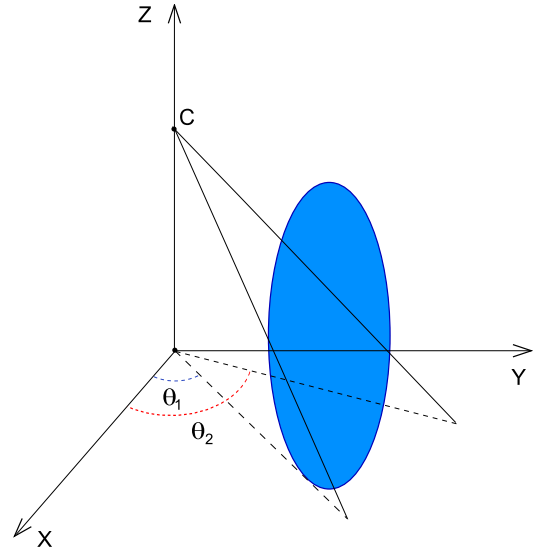


Figure A.2: Finding spheroid limits in horizontal scanlines

**B**

---

## **Sample Detection and Tracking Frames**

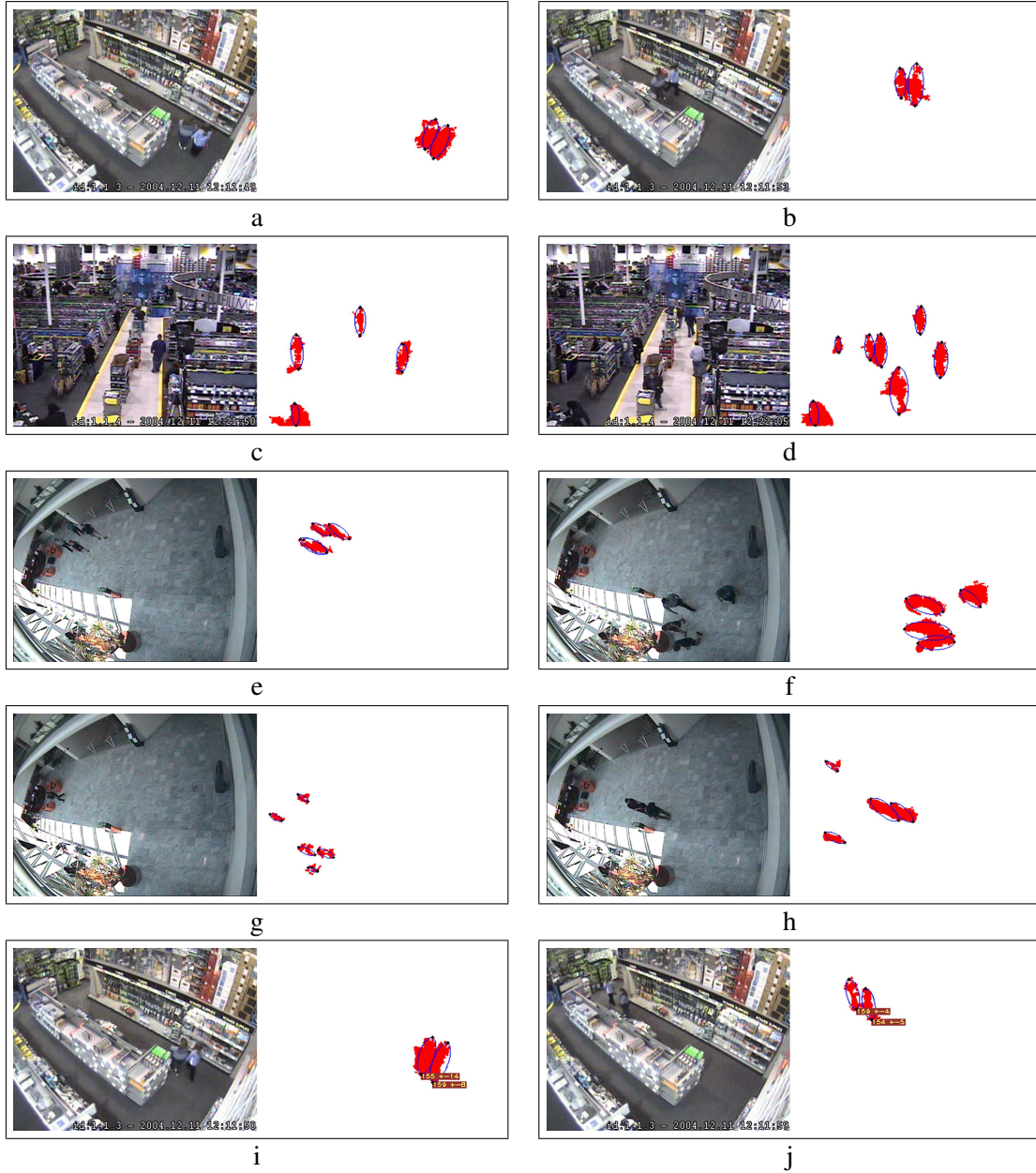


Figure B.1: (a) - (j) Head candidates from test frames. Left image is the original frame. On the right image red represents foreground mask, small black dots indicate the locations of  $T_i$  and  $B_i$ ; blue ellipses are fitted with  $T_i B_i$  as the major axis; (i) and (j) Height detection: brown plates contain height mean and variance for each ellipse



Figure B.2: Sample frames showing tracking in three sequences from CAVIAR dataset [6]: columns are three tracking sequences, rows are frame snapshots taken successively. **Sequence 1:** a group of four people enters the scene. **Sequence 2:** Two pairs of customers enter the store (0,1 and 2,3) and 5,4 re-appear at the back of the store. **Sequence 3:** 1 exits the store, 2 and 3 walk past the store and 4 enters the store

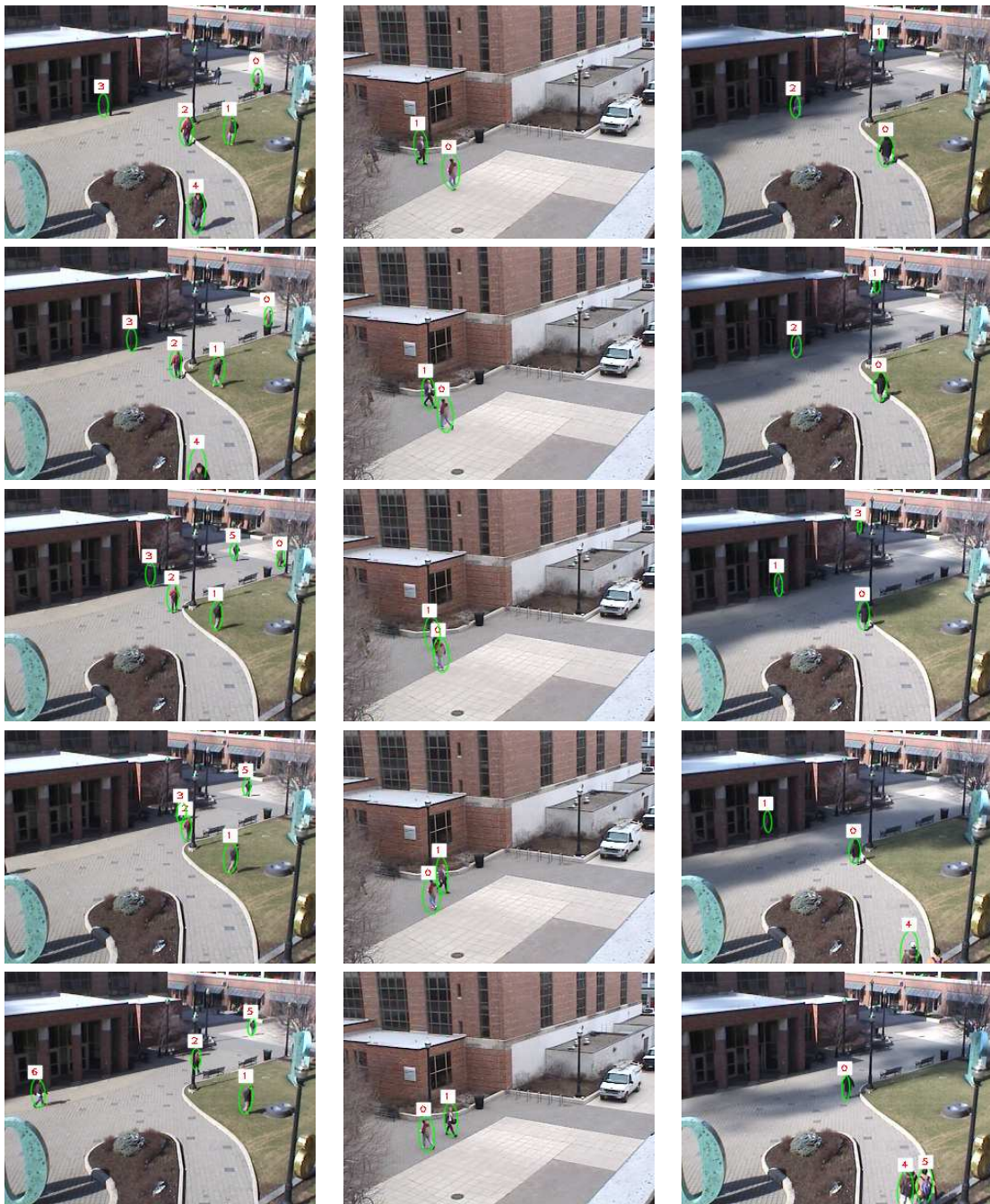


Figure B.3: Sample frames showing tracking in three sequences from OTCBVS dataset [19]: columns are three tracking sequences, rows are frame snapshots taken successively. **Sequence 1:** Multiple pedestrians are tracked on the sidewalk through occlusions. **Sequence 2:** 0 and 2 intersect and then split. **Sequence 3:** A group 4, 5 and several single pedestrians are tracked



Figure B.4: Sample frames showing tracking sequence from apparel retail store (rows are frame snapshots taken successively). Customers 11 and 12 enter the store and proceed along the left side wall. Customers 1, 3 and 5 are at the checkout. Customers 6, 9 and 0 are browsing fixtures with the merchandise



Figure B.5: Sample frames showing tracking sequence from electronics store (columns are three tracking sequences, rows are frame snapshots taken successively) **Sequence 1:** 0, 1 are walking with a cart, 4 is dwelling then leaves, employee 3 is assisting customer 2 **Sequence 2:** employee 1 is assisting customer 2, customers 3,4 are slowly browsing **Sequence 3:** A complex scenario, with multiply bodies partially occluded

## **Curriculum Vitae**

Alex Leykin received his BS and MS in Computer Science and Applied Mathematics, Kharkiv Polytechnical Institute, Ukraine in 2000. Followed by MS in Computer Science, Indiana University, USA in 2002.

In 2001, worked as a Research Assistant, AI Group, Dept. of CS, Indiana University on image classification. Summer 2002, as a Research Assistant, Dept. of CIS, Indiana University-Purdue University Indianapolis studying text readability. From 2005 works as a Research Assistant at the Customer Interface Lab, Kelley School of Business, Marketing Department, Indiana University.

His research interests include vision-based tracking and activity analysis, vision-guided vehicle navigation, semantic level image classification and image processing.

# Index

- activity modeling, [48](#)
- agglomerative clustering, [50](#)
- appearance model, [36](#)
  
- background subtraction, [9](#), [26](#)
- blob, [29](#)
- body, [36](#)
  
- camera model, [21](#)
- co-dwelling, [54](#)
- crowded environments, [8](#)
  
- deterministic clustering, [54](#)
- distance weight plane, [42](#)
- Dynamic Time Warping, [11](#)
  
- equirectangular projection, [24](#), [93](#)
  
- floor map, [26](#)
- fuzzy clustering, [58](#)
  
- head candidates, [30](#)
- Hidden Markov Model, [49](#)
- hierarchical clustering, [50](#)
- homogeneous coordinates, [23](#)
  
- image coordinates, [23](#), [24](#)
- individual activities, [13](#)
  
- jump-diffuse mutations, [44](#)
  
- likelihoods, tracking, [41](#)
  
- Markov chain, [33](#)
- Markov chain Monte Carlo, [33](#)
- MCMC, [33](#)
- mean shift, [45](#)
- Metropolis-Hastings algorithm, [33](#)
  
- obstacle map, [39](#)
  
- perspective projection, [93](#)
- priors, tracking, [37](#)
- proposal distribution, [35](#)
  
- reversible operators, [35](#)
- robust estimators, [60](#)
  
- scene crowdedness, [31](#)
- shopper group, [5](#)
- state mutations, [35](#)
- swarming activity, [5](#), [13](#), [56](#)
- swarming event, [53](#)
  
- Tukey's biweight function, [60](#)
  
- unprojected map, [24](#)
  
- video analytics, [1](#), [3](#)
- video mining, [1](#)
  
- world coordinates, [23](#), [24](#)
  
- z-buffer, [41](#)