2011

Modeling Pedestrian Behavior in Video

Paul Scovanner
University of Central Florida

MODELING PEDESTRIAN BEHAVIOR IN VIDEO

by

PAUL SCOVANNER
B.S. University of Central Florida
M.S. University of Central Florida

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2011

Major Professor: Marshall F. Tappen

ABSTRACT

The purpose of this dissertation is to address the problem of predicting pedestrian movement and behavior in and among crowds. Specifically, we will focus on an agent based approach where pedestrians are treated individually and parameters for an energy model are trained by real world video data. These learned pedestrian models are useful in applications such as tracking, simulation, and artificial intelligence. The applications of this method are explored and experimental results show that our trained pedestrian motion model is beneficial for predicting unseen or lost tracks as well as guiding appearance based tracking algorithms.

The method we have developed for training such a pedestrian model operates by optimizing a set of weights governing an aggregate energy function in order to minimize a loss function computed between a model's prediction and annotated ground-truth pedestrian tracks. The formulation of the underlying energy function is such that using tight convex upper bounds, we are able to efficiently approximate the derivative of the loss function with respect to the parameters of the model. Once this is accomplished, the model parameters are updated using straightforward gradient descent techniques in order to achieve an optimal solution.

This formulation also lends itself towards the development of a multiple behavior model. The multiple pedestrian behavior styles, informally referred to as “stereotypes”, are common in real data. In our model we show that it is possible, due to the unique ability to compute the derivative of the loss function, to build a new model which utilizes a soft-minimization of single behavior models. This allows unsupervised training of multiple different behavior models in parallel. This novel extension makes our method unique among other methods in the attempt to accurately describe human pedestrian behavior for the myriad of applications that exist. The ability to describe multiple behaviors shows significant improvements in the task of pedestrian motion prediction.

To my family for their love, support, guidance, and understanding.

Acknowledgments

I would like to thank my advisor, Dr. Marshall F. Tappen, for his guidance and support. None of this would have been possible without the insight and tireless assistance I received. Thanks to my committee members, Dr. Rahul Sukthankar, Dr. Gita Sukthankar, and Dr. Charles Hughes for their time and contributions to this dissertation.

Thanks to all my colleagues for their collaboration and friendship over the years, including Vladimir Reilly, Shayne Czyzewski, Mikel Rodriguez, Ramin Mehran, Subhabrata Bhattacharya, Enrique Ortiz, Chris Ellis, Zain Masood, Nazar Khan, Jiejie Zhu, Saad Ali, Yaser Sheikh, Saad Khan, Arslan Basharat and the many other students who I have worked alongside at the University of Central Florida. Thanks to Adam Fernandez Esq. and Benjamin Kretzmann M.D. for their inspiration to pursue higher education and a lifetime of friendship. Finally, thanks to Dr. Mubarak Shah for giving me a start in computer vision.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Background and Motivation

Vision is one of the most important tools available to humans, it is apparent in our language where “to see” means “to understand.” In computer vision we attempt to create algorithms and methods that allow computers to understand images and videos. It makes sense that the purpose of many of the tools in computer vision fixate on identifying and understanding human beings. From face detection to fundamental matrices, the focus of the field of computer vision seems to be to better understand the world and allow human beings to better interact with it. Among the many problems within computer vision concerning humans, understanding pedestrian activities and behaviors has been an important research area where many practical applications are starting to be developed in recent years. Modeling these pedestrian movements is a unique area due to the complex social interactions of human beings. Only in the past couple of years have researchers begun to train their models from real world data. In this dissertation we will explore the current work in learning pedestrian models, propose our own method, and compare methods on a number of challenging scenes and datasets.

A pedestrian model in its simplest form is an algorithm that can generate, or predict, the path a pedestrian may take. A pedestrian model, able to predict realistic pedestrian behavior, is useful for applications such as generating emergency simulations in architectural designs, artificial intelligence in games, and improving the ability of human tracking algorithms. Computerized behavior models have been around since the 1980's. These models were initially created by manually tuning parameters until the resulting simulations looked correct, or some desired emergent behavior was seen. Researchers would simulate two intersecting crowds and look for the formation of lanes, and other qualitative formations. More recently models have been developed that can be learned using observed pedestrian tracks. These observed tracks are used to train the parameters of a model, and result in more accurate prediction models than manually tuned models. Our model is robust and highly accurate, but can take time to train; other models rely heavily on developed non-linear machine learning techniques to solve ill-posed problems quickly.

Pedestrian models are not only used for generating simulations; they are also used in tracking applications. The task of tracking objects in a scene is one of the cornerstones of computer vision. Tracking relies on the fundamental problems such as classification and recognition of objects, scene structure and camera geometry, and incorporates machine learning techniques. Tracking of pedestrians is a complex problem in its own right, so we will break it into subproblems. The task is essentially to find the location of pedestrians seen at a previous time. Generally a pedestrian tracking algorithm will leverage two main pieces of information: the appearance of a pedestrian, and some scene information including the known or expected location of a pedestrian. The second part of this problem can be solved using pedestrian modeling. Using a model designed to predict pedestrian movement provides a much better predicted location than statistical methods which were taken from domains where noise is better understood and conforms to Gaussian distributions. Due to this extra domain knowledge, we can show how pedestrian tracking becomes a more solvable problem than general object tracking.

This work is motivated by the many applications of an accurate mathematical formulation of pedestrian motion that can be gained from training on real world data. These models are useful in creating realistic virtual worlds in computer games. They can be applied to serious applications, aiding architects in testing stadium designs so that people can safely exit in case of a disaster. Certainly these models are fundamental to the relatively "pedestrian" application of human surveillance. The explosion in the fields of computer vision and machine learning have shown that given the proper tools, a computer algorithm can help humans to see possibilities that were never before possible.

1.2 Challenges

The challenges presented by pedestrian modeling are as numerous as they are complex. While the need for accurate models is apparent, the best way to solve the problem is not. What motivates a pedestrian? While physical restrictions must be taken into account, how important are social restrictions? Some researchers have taken a macroscopic global approach

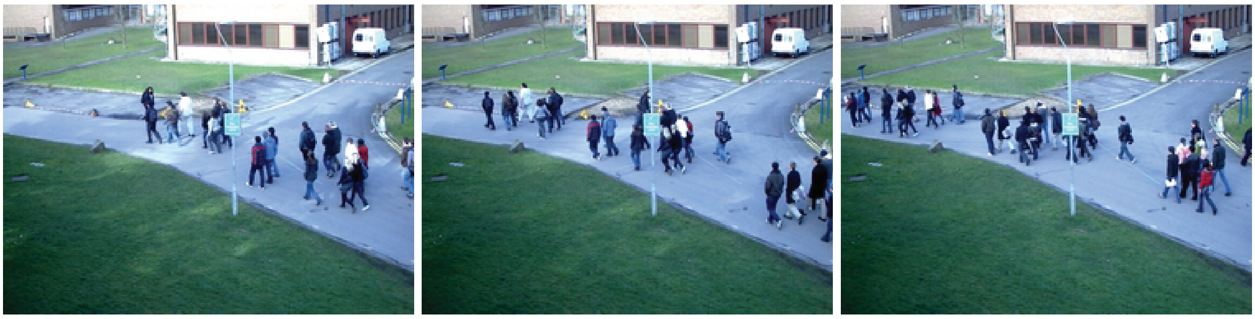


Figure 1.1: Sample pedestrian images, taken from the PETS dataset, showing groups of individuals moving together.

to representing crowds as if they were governed by the laws of fluid dynamics. While others have developed individual agent models and let crowds grow out of large numbers of agents. The best method is not evident, nor is the range of possible methods. Therefore, this work attempts to approach the problem with as few assumptions as possible. We assume that it is possible to learn human behavior through observation, that each pedestrian decides his own path, that the path taken is not necessarily optimal, and that the forces that motivate a pedestrian include things such as a destination, avoiding collisions, and moving at a comfortable pace. Beyond this we make some assumptions about the functions that attract and repel pedestrians; however all our functions are posed as a sum of energy components. In this way, we allow the data to train the model with as much flexibility as possible while being robust enough to generalize human behavior.

1.3 Goals

The purpose of this dissertation is to explore algorithms which are able to learn and predict pedestrian behavior. We investigate methods of parameter training that allow pedestrian behaviors to be inferred from object tracking algorithms. We take those trained models and show that they can be useful in many applications. This dissertation intends to show that trained pedestrian motion models can improve prediction error rates, allowing generation of more accurate simulations. Also, due to the fact that the proposed model in this dissertation is based on psychological factors, our model is able to identify and simulate various "types" of pedestrians. We will also show that the ability to more accurately predict pedestrian movement can result in significant improvements to pedestrian tracking algorithms. The main goal of this dissertation is to further the field by introducing new ideas with practical applications and quantifiable benefits.

1.4 Outline of Research

In this dissertation we present a framework approach to the problems of pedestrian simulation and tracking. We will emphasize scenes that contain large numbers of people, and show how the knowledge of the current scene is able to greatly improve the ability of pedestrian path prediction. Our model is created using parameters that correspond to defined psychological desires and physical restrictions, and thus a set of trained parameters can be understood intuitively. For instance, we can compare the relative weights of "pedestrian avoidance"

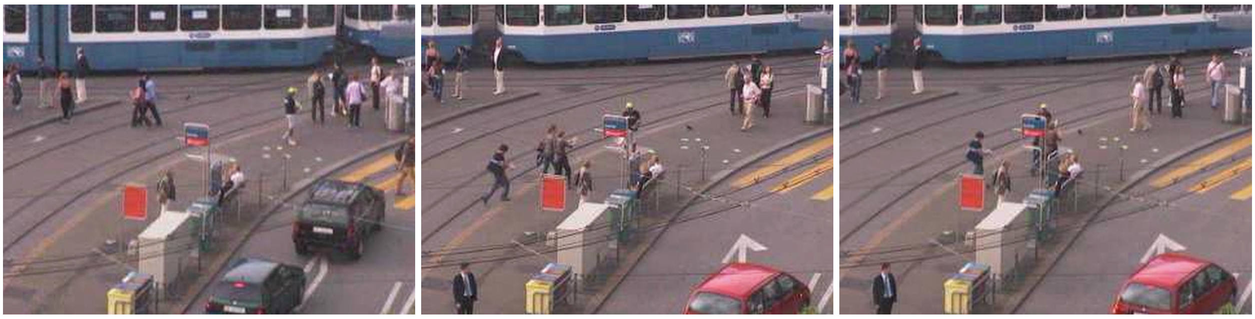


Figure 1.2: Frames taken from the Central dataset show pedestrians interacting with vehicles and other pedestrians.

to “desire to reach destination” and gain an understanding of the overall personality of the pedestrians in a scene. This ability is further useful when multiple behavior types are observed. Our method, which is unique in its ability to train models for multiple behavior styles in parallel, is shown experimentally to be quantitatively superior to models that are limited to describing single behavior types.

1.4.1 Pedestrian Modeling

The first objective covered by this dissertation will be learning a single pedestrian behavior model, able to predict an accurate path based on previous training data. Our research will show how a model can be formulated such that it can be efficiently optimized even with large numbers of parameters that govern pedestrians’ complex motions. We compare this model against similar pedestrian models, as well as classical noise reduction techniques commonly applied to path prediction such as Kalman filters.

1.4.2 Stereotyping

Our original pedestrian model is able to learn an optimal general model for all pedestrian behavior in a scene. We will expand on the basic pedestrian model, and show how an optimal *set* of pedestrian models can be learned from a single scene in an unsupervised fashion. By allowing multiple behaviors, we will show that the combined model is able to greatly improve on previous models that assume every person in a scene obeys the same set of social norms. The approach that this dissertation will discuss does not require any labeling of the personalities of the pedestrians.

In any scene containing large numbers of pedestrians, various behavior patterns will exist. In order to accurately predict behaviors, a pedestrian motion model must take this into account. In this dissertation we will discuss a method of allowing a fixed number of behavior types to be learned in parallel. These behavior types are often referred to as *stereotypes* or *personalities* to simplify the language. These are not stereotypes or personalities in the classic definition, and in no way use the appearance of a person. Rather, these stereotypes separate individual tracks into groups that exhibit similar movement behaviors such as: traveling quickly through a crowd towards a destination, staying nearby a group of friends, or standing and waiting. The ability to separate pedestrians into these types of general groupings is enough to significantly improve results.

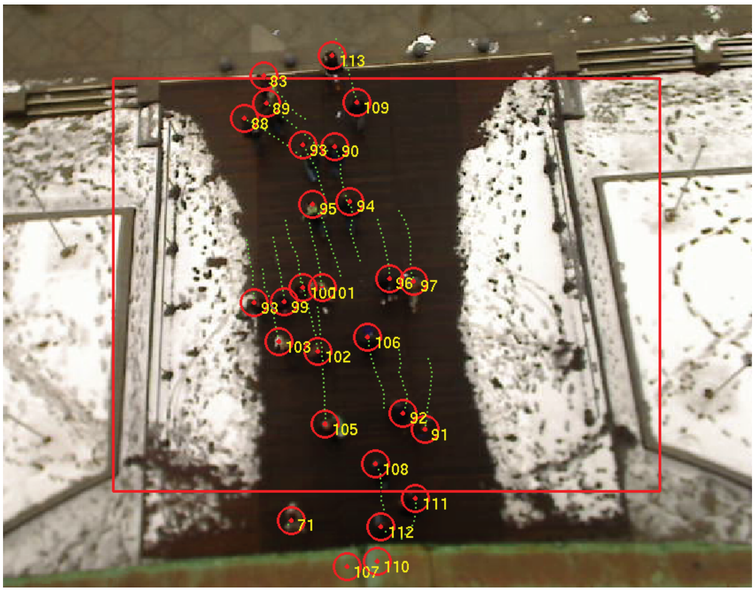


Figure 1.3: Annotated pedestrians from the LTA dataset [PES09].

1.4.3 Tracking

Tracking of objects in video is a mature research area within the broader computer vision discipline. Tracking is useful in many applications such as security and surveillance, video indexing, object counting, and anomaly detection. The focus of this dissertation pertaining to tracking will be confined to the ability of motion prior information to positively influence a standard algorithm for tracking. To this end we have used a standard baseline object tracking approach and incorporated two motion priors: Kalman filtering, and our pedestrian model. The Kalman filter is a standard approach to estimate motion information, robust to random noise. We use a linear filter, which is common in literature.

In addition, the experimental results on tracking in this dissertation utilize dynamics that are not important to prediction or simulation methods. When predicting a path, the information from the other pedestrians in the scene is available, and is used to compute the energy fields that govern the model. When addressing the task of tracking, we modified the prediction algorithm in such a way that the locations of all pedestrians are the positions that are actively being tracked. This is the way a fully automatic tracking algorithm must function in real time applications, and can create additional dynamics due to all pedestrians being actively tracked in concert.

1.5 Organization of Dissertation

The rest of this dissertation will cover learning pedestrian behavior models, creating a multiple behavior model, and improving appearance based tracking using pedestrian models. First, Chapter 2 will review related literature that is important to the understanding of this dissertation. This discussion will cover gradient descent learning methods, various social force models for pedestrian prediction, advantages and disadvantages of flow field models for pedestrian modeling, object tracking and other background topics important in our framework. Chapter 3 will introduce our social pedestrian model. First, the energy model will be explained, which is guided by pedestrian motivations, or desires. After defining the loss function for the basic model, we will describe the extension of the loss function to describe multiple behaviors. The learning process for both the single behavior, and multi-behavior stereotype model will be covered. Experimental results will show how this model compares to state of the art methods as well as more classical approaches. In addition we will compare the results of our own model with varying numbers of behaviors and the significant advantages of the multi-behavior model become apparent. Chapter 4 will cover pedestrian tracking. Motion priors have been useful to tracking algorithms, and this chapter will show how our pedestrian model can be used in conjunction with appearance based tracking to leverage as much information as possible to accurately track pedestrian movement. Finally, Chapter 5 will conclude this dissertation, summarizing the results and discussing avenues for future development.

CHAPTER 2: LITERATURE REVIEW

This chapter will review relevant works that are necessary for an understanding of this dissertation. It will cover many different approaches to pedestrian motion prediction. We will split these works into two general categories which will be organized into Subsections 2.1.1 and 2.1.2. Scene based models describe the entire scene directly using a macroscopic view and the motions of the pedestrians are indirect outputs of governing scene forces. Agent based models instead describe the pedestrians' emergent features are in essence a side effect of these models, as they are in real world crowd situations.

Literature important to the topic of pedestrian tracking will be covered in Section 2.2 of this chapter. This will cover some general object tracking methods, as well as tracking algorithms specifically tailored to the tasks of pedestrian tracking and pedestrian tracking in crowds. This section will also review work related to the Kalman filter, as it is a standard method for motion prediction used by many tracking algorithms.

2.1 Pedestrian Models

Pedestrian models can be generally split into two main categories, macroscopic and microscopic. A macroscopic model describes a crowd, while a microscopic model describes individuals.

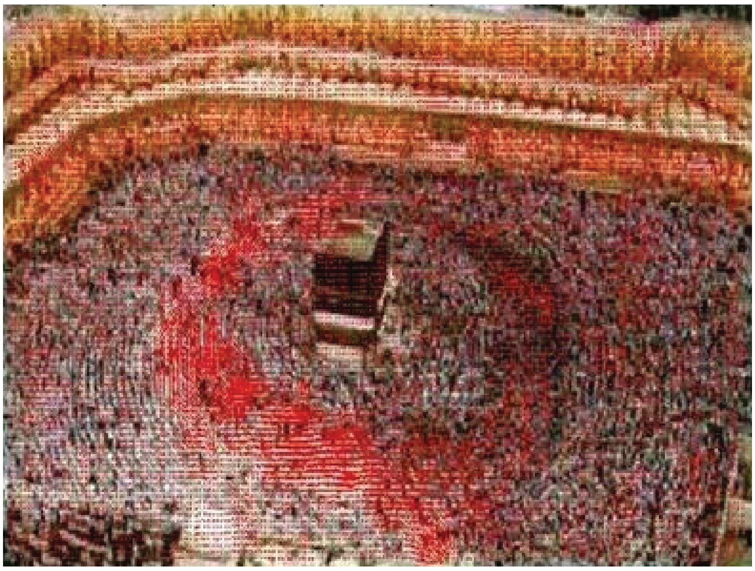


Figure 2.1: Global scene model for determining crowd stability.

2.1.1 Scene Based Models

Scene based models put constraints on the scene, and model/predict pedestrians as if they are controlled by the scene. One could think of pedestrians moving down a sidewalk like leaves floating down a stream. These models are generally used for large crowds and give no explicit regard to the tendencies of the parts which make up the whole [Hug03]. These methods are popular for simulation and stability detection in extremely large crowds. Also, there is a sort of hybrid approach to scene based and agent based models, often called the continuum approach. This approach was introduced by Hughes [Hug02] and was further developed by Treuille et al. [TCP06]. The continuum approach takes a similar form as [HM95], however it calculates the forces with respect to the environment (pedestrian density of a specific region, velocity of the average person at that location, discomfort experienced

Figure 2.2: Agent based simulation model trained by pedestrian videos.

by being at such location) and then assumes that all pedestrians will move according to both these shared forces, as well as the forces guiding them towards their goal. The work of Ali et al. [AS08] uses floor fields, which are similar in nature to the continuum approach, to detect unstable regions in very dense crowds. These scene based approaches tend to do well in large scale crowds where human densities exceed 1 person per square meter. In these scenes it could be argued that a person does not truly chose their own path, but rather their path is decided by the crowd, and thus macroscopic methods of simulation and prediction are well suited for these situations.

2.1.2 Agent Based Models

One of the first pedestrian modeling methods to describe how a person travels with regard to their surroundings was the social force model, proposed by Helbing and Molnár [HM95]. At its core, the social force model operates on the assumption that the scene, the person's preferences, and other pedestrians exert forces on a person, which help to determine his or her path. This model allows a large scale view of large crowds of people to be modeled by describing the characteristics of individual people using a combination of relatively simple forces. This basic model has been extended to more accurately describe various kinds of crowds [LKF05] [HFV00]. Other models for social behavior models include the Human Steering Model [FWT03] and the Space Syntax Method [PT01].

Other motion assumptions not yet covered include probabilistic scene models. These models, such as the one used in [SSS09] learn the probability of a person appearing in a certain location given their previous observed locations. These models are effective at learning commonly used paths in a scene; however they are not aware of a scene's current situation, and differ greatly from social pedestrian based models such as ours. While this method is scene specific (ie: a model trained in one location is not useful to a different scene), ideally one would want to use all available information to better predict pedestrian movements. Integrating such scene based motion models with social behavior based motion models and modern appearance based tracking is an interesting future direction for our research which will be discussed in Chapter 5.

It should be noted that these approaches to pedestrian modeling are significantly different than systems that cluster trajectories, such as [SG99] [WMG09]. Rather than clustering trajectories into similar groups, the pedestrian models proposed here model the decision-making process in how a person moves. Clustering models can predict where a pedestrian is likely to be in a given scene, but do not explain why he or she is there. In contrast, the models considered in this paper directly model the pedestrian's underlying motivations to predict how and why the pedestrian moves.

2.1.2.1 Discrete Choice Model

Discretization of the scene allows for off the shelf machine learning methods to be applied to the pedestrian modeling task. There are two major works which have learned pedestrian models by discretizing the space in which pedestrians exist.

Antonini et al. [AMB06] model pedestrian behavior as a series of discrete choices. In this model, both time and space are discretized. At each time instant, the pedestrian chooses the next location from a set of possible discrete locations. This choice is made using a multi-class linear classifier. This leads to a straightforward formulation of the learning problem; however discretizing the possible destinations introduces issues. The most pressing issue is the difficult balance between making the grid too coarse, which affects the accuracy of the prediction, and making the grid too fine, which enhances accuracy but requires substantially more computation. Their work proposes an adaptive spatial discretization approach to overcome the difficulties associated with a fixed grid, but this increases the complexity of implementation. Their model contains a total of 8 trained parameters.

2.1.2.2 Continuous Pedestrian Models

Another approach, described in [JHS08], adapts the classic social force model [HFV00]. The goal of their work is to find parameters of the model such that the simulated movements match tracks in video. This is accomplished using an unspecified evolutionary algorithm to optimize two parameters in the model. In their work, the learning algorithm is not described, so it is unclear how well the learning will scale up to models with many parameters. In the past few years a number of methods have been developed using continuous agent based models to learn pedestrian movements in crowds [PES09] [PET10] [PEG10] [KAO11] [TS10] [LST10] [TT10] [ST09]. The rest of this subsection will review those works.

Recent work by Pellegrini et al. [PES09] learns a pedestrian model called LTA. This work creates an energy function which during training optimizes 6 parameters using a genetic algorithm. Direct quantitative comparison using the dataset provided by [PES09] and the method described by the paper is discussed later in this dissertation.

The authors have also extended this work in [PET10] where multiple Gaussian functions are fit to the energy function. In this updated work each peak of the Gaussian function is treated as a possible location, creating multiple hypotheses until a prediction is made at a later time. This model is referred to as stochastic linear trajectory avoidance, or sLTA. This work shows a 20% improvement over a baseline constant velocity model when observations are made only once every 4 seconds. At the framerate of the annotation (2.5 fps) the tracking difference reported in [PET10] is negligible.

Most recently this work was also extended in [PEG10] where group behaviors are used to improve tracking in crowded scenes. This need to model group behavior has been acknowledged in previous publications by ourselves [ST09] as well as by the original LTA publication [PES09]. In this most recent work by Pellegrini et al. chose to jointly model group assignments and paths. This results in a third order CRF model which proves to be too complex to train directly. Instead statistics based on position, speed, and orientation over the trajectory are used, and the parameters of the model are trained indirectly based on the distributions of these statistics.

In our research we found the need to integrate the group assignment step into the model itself was unnecessary, a simple SVM using simple features such as mean distance, minimum distance, and difference in velocities was sufficient to accurately estimate group assignments in many of these real world scenes. This was confirmed by Yamaguchi et al. [KAO11], who used a similar SVM to predict the group assignment greatly reducing the complexity of their social force model. All three research groups have independently shown minor improvements when group assignments are allowed; however the differences are not drastic in any of these works including our own.

The Human Steering Model (HSM) has shown promise in both the virtual and real world domains [TS10]. This work builds on the work of Fajen et al. [FWT03], and models movement by individuals heading, speed, goals, and obstacles. The steering model assumes a constant speed; however the orientation of the pedestrian reacts to obstacles in the scene. Tastan et. al [TS10] trained their HSM method using a parameter grid search, these training results were verified by a non-linear least square error minimization. This model was then applied to tracking in both virtual worlds, as well as indoor environments using a particle filter framework [Thr02]. In both the virtual and indoor environments the HSM showed significant advantages over other baseline approaches in navigating amongst static obstacles.

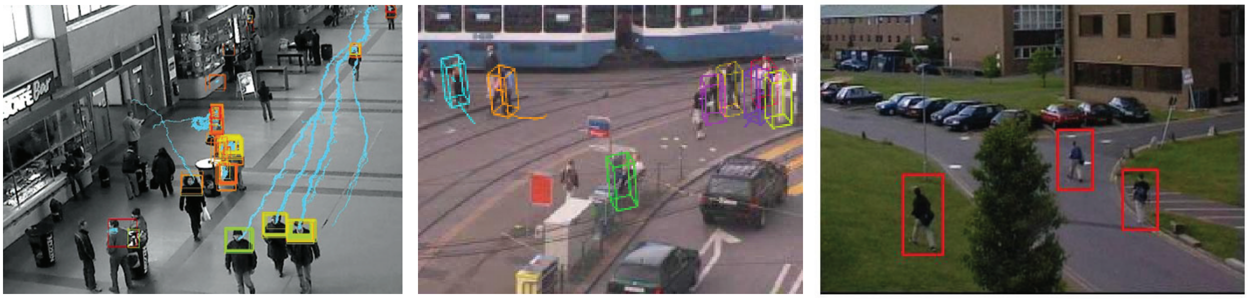


Figure 2.3: Selection of outputs from various pedestrian tracking methods. Left: *Tracking Pedestrians With Machine Vision* [Sla07] Center: *Coupled Detection and Trajectory Estimation for Multi-Object Tracking* [LSV07] Right: *Detecting Pedestrians Using Patterns of Motion and Appearance* [JVV03]

Similar work which is blurring the line between human and robot path planning includes [TK10] [LST10] [DH09] and [TT10].

2.2 Tracking

Tracking involves associating current observations with previous ones. Points, silhouettes, and bounding boxes are commonly used to define objects [YJS06]. The problem of association becomes significantly harder when objects are allowed to appear and disappear, and when multiple objects exist simultaneously in a scene. Multi-target tracking has been explored in recent years [HWN08] [LSV07] and these tracking algorithms have proven to be very successful at tracking pedestrians. Researchers in object tracking have focused on improving the appearance model [GB06] [BRL09], object acquisition and detection [ARS08a] [DT05] [FMR08], and data fusion [ZLN08].

Since tracking is such a well studied area of computer vision, there exist survey publications whose sole focus is to review literature and categorize the research. For more in-depth discussion of the history and recent developments in the broad field of object tracking, please see [YJS06]. The field of pedestrian tracking is also surveyed by Enzweiler et al. [EG09].

The following three subsections will cover the general field of object tracking, the specific subfield of pedestrian tracking, and data fusion methods for combining image and motion information to create reliable tracks.

2.2.1 Object Tracking

The field of object tracking is not only one of the oldest, but also currently one of the largest and most active in computer vision. It would be impossible to cover all approaches to object tracking, instead we will focus on cornerstone publications. Focus will also be directed at the approaches which integrate a priori motion models and combine these statistical estimations of location based on previous locations with appearance based tracking estimations which use features based on color, shape and/or texture.

The general problem of object tracking is a difficult problem and is prone to many issues. Objects may undergo various changes to appearance in both rigid and non-rigid shapes, partial or full occlusion, and complex motion just to name a few. Often as the appearance of an object changes over time, the tracker will slowly "drift" off the intended object and track a complectly different object or become stuck on a part of the background. Researchers attempt to use information besides the original appearance to help handle these commonplace issues. Adaptive appearance models are common [JFE03] [ZCM04] for objects which can change appearance or self-occlude. Reacquisition methods are important to non-stationary camera tracking algorithms [ARS08a]. Motion prior information [HWN08] [LSV07] [BYB09] is a standard which is found in just about every tracking algorithm in some form. Since object tracking is less constrained than pedestrian tracking, the methods of motion prediction are generally much simpler.

Object tracking has relied on some form of motion information since the beginning. One of the first major works on the subject of object tracking was *Tracking Objects in Space* [RA79]. Published in 1979, the method attempted to track objects in the so-called block world used positional expectations based on the velocity of the objects. Object tracking developed more and more complex models. In 1986 Broida et al. [BC86] introduced the use of the Kalman filter to better predict an object's motion. This method of motion prediction became a standard in object tracking and is still quite popular today.

Modern tracking algorithms often leverage advanced machine learning techniques to help solve the issues related to appearance based tracking. In certain situations with moving cameras and complex motions, objects do not always move in a predictable manner. Thus, methods such as [BYB09] which focus on tracking faces merely assume that an object will be within a certain fixed radius to the location it was previously observed. These methods work well in the object tracking domain, however they experience issues when they are applied to the isolated domain of pedestrian tracking.

The Normalized Cross-Correlation method used for appearance based tracking is a standard model used to measure the similarity of an image patch. Pellegrini et al. [PES09] use the following squared exponential equation to determine the probability map for the appearance model.

$$P_{data}(\mathbf{p}) = \frac{1}{Y} \exp\left(-(NCC(\mathbf{p}, \mathbf{p}_i^0) - 1)^2\right) \tag{2.1}$$

The above probability map is multiplied by a Gaussian centered at the motion prediction location to compute the tracker's prediction, a standard approach for information fusion which is further discussed in Section 2.2.4. This method has been used as a straightforward baseline tracker for social force modeling in recent publications [KAO11] [PES09] [PET10], and will be used in our tracking framework as well.

The following sections will review specific literature relevant to the task of tracking pedestrians, motion estimation, and information fusion.

2.2.2 Pedestrian Tracking

Due to the specific challenges which exist in the pedestrian subdomain of tracking, many methods specific to tracking pedestrians exist in the literature. Often these methods are more likely to rely on a motion prior, whether it is a social behavior or a linear behavior model [PES09] [ELS09] [ZN03].

The most specific subfield of tracking which relates to this dissertation is tracking pedestrians in crowds. This subject is a focus of many publications due to the high difficulty of the problem [BC06] [RAK09]. Ali et al. use floor fields [AS08] to track pedestrians in extremely high density crowds.

Followup work on the LTA method [PET10] extended the original work to allow multiple hypotheses. While this approach improves the ability to generate simulations with random and track objects, it does not necessarily improve prediction. This is because the prediction still requires the method to pick the best hypothesis. However it does create a richer and more descriptive probability field for the pedestrian's path. Surprisingly though, the authors found that multiple hypotheses did not have a considerable effect on tracking performance [PET10]. In our work we find that the methods for pedestrian motion models described in the following chapters significantly improve the tracking performance.

2.2.3 Motion Estimation

Methods for motion estimation in tracking algorithms can be generally separated into three non-distinct categories. Social force models [PES09] [AMB06], Kalman filters [BC86] [BK99] [RS99], and particle filters [KBD05] [AS07] [BRL09] [ZCM04]. Each of these methods attempts to estimate the individual object locations using different assumptions. This section will cover Kalman filtering and particle filtering, while social force models were introduced and related work was discussed previously in Section 2.1.2.

Kalman filters are used to estimate the state of a linear system when the distribution is assumed to be Gaussian. The Kalman filter is fast, and has proven its worth in real-time tracking systems. Extensions to the Kalman filter, such as the Extended Kalman Filter [BB88] and the Unscented Kalman Filter [JU97] can be used to predict non-linear data; however the Kalman filter always assumes an underlying Gaussian distribution of possible states.

Particle filtering offers an attractive tracking framework due to its non-Gaussian state assumption. Particle filters work by importance sampling; thus, given enough particles, these filters can describe any distribution imaginable. Particle filters are more computationally intensive than Kalman filters. Khan et al. [KBD05] showed that it is possible to use particle filters to track objects which interact with each other.

2.2.4 Information Fusion

All modern tracking algorithms must make decisions based on a number of inputs such as shape, appearance, location, and camera state. The most straightforward approach to combining input from a number of independent observations is by converting each input into a probability and computing the product of each probability [KBD05] [ZCM04] [PVB04] [CS10] [PES09]. This standard statistical approach is valid whenever inputs are independent, as appearance and location are. Due to this, simple probability multiplication is widely popular for independent probabilities.

2.3 Summary

This chapter has reviewed the most relevant research on the topic of pedestrian motion prediction and tracking in crowds. We have described the advantages and disadvantages of scene and agent based models in the context of pedestrian path modeling. We have reviewed cornerstone publications as well as recent breakthroughs which shape the field of both motion modeling as well as tracking. We have introduced some of the aspects which make our method unique. These elements represent a significant contribution to the field, and will be discussed in great detail in the following chapter.

CHAPTER 3: LEARNING PEDESTRIAN MODELS

3.1 Introduction

Models that can predict how pedestrians choose to move in a scene are becoming increasingly useful for a variety of research problems. In pedestrian tracking applications, an appearance based tracker often relies on a motion based prior [ARS08a] [ELS09] [PES09]. Pedestrian movements are also important for generating realistic crowd movement in virtual environments [TCP06] [GCC10] [TS10]. In simulators, they can be used to evaluate structures for crowd flow or evacuation [LKF05] [HFV00]. More recently, agent models have been used to detect anomalous events in both pedestrians [MS09] [MLB10] and motor traffic [SC10].

In this chapter, we introduce our Stereotyped Pedestrian Model, abbreviated SPM, and show how tracks can be used to learn models of pedestrian movement. Our system is unique in that a pedestrian's movements are formulated as a series of continuous optimizations. This formulation overcomes significant issues with previous attempts at learning behavioral models from video such as [AMB06] [JHS08]. Specifically, our model does not require the discretization of the space of possible locations and is able to learn more complex models with more parameters than other methods.

A significant contribution of our work includes the ability to model multiple pedestrian behaviors. This is a novel extension which has not yet been explored in the computer vision field. This chapter will show that modeling multiple pedestrian behaviors, also referred to as “stereotypes”, represents a marked quantitative improvement over existing methods. As part of the training process, pedestrians are clustered according to the behavior model that best matches their movement in an unsupervised manner. The extension is simple and elegant, and requires no additional parameters or additional computationally expensive expectation maximization style of training.

The training process’ ability to accommodate with a relatively large number of parameters makes it possible to learn a model with multiple types of behaviors and more complex pedestrian movements. It also requires us to make fewer assumptions about the way in which pedestrians travel. In some cases, the training can actually inform us as to a general pedestrian’s mindset.

Our model can both produce qualitatively accurate simulations of pedestrian movement, such as the simulations shown in Figure 3.1, and provide predictions of pedestrian movement that are quantitatively more accurate than standard methods for predicting movement.

Our model will be outlined by Section 3.2. Section 3.3 and 3.4 will go in depth to cover the details of our model. In Section 3.5 we show how our model generates prediction tracks and the details of our learning method are given in Section 3.6. Quantitative and qualitative evaluations are discussed on multiple datasets in Sections 3.7, 3.8, and 3.9.

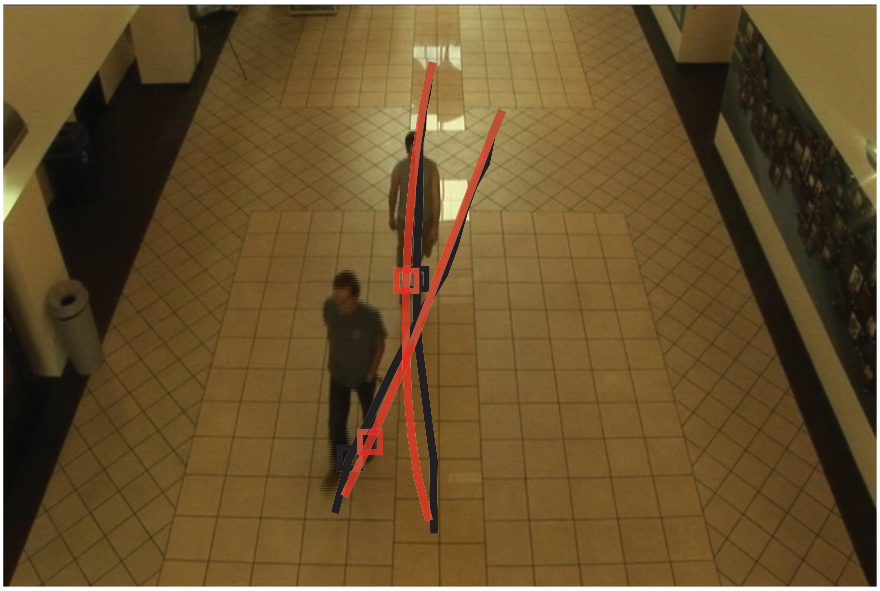


Figure 3.1: This work focuses on learning a model of pedestrian movement from real-world pedestrian tracks taken from video data. This image shows an example of two pedestrians' paths, shown in black, and the system's predicted paths for those pedestrians, shown in red. Each pedestrian attempts to avoid the other in order to reach their desired goal.

3.2 Model Overview

Our method shares some aspects with those discussed in [PES09], such as the use of an energy function as well as the similarity in the motivations which guide pedestrians. Our model is novel in the formulation of the energy function, which is created in such a way that differentiable tight convex upper bounds can be computed. This allows efficient minimization of the loss with many more parameters than existing methods, resulting in a more adaptable and robust model that is capable of describing more complex motions. Models in this dissertation contain as many as 42 parameters, which describe 3 separate behavior models, or stereotypes, of pedestrians. This is many more than previous systems, which use between 2 and 8 parameters. Other methods also assume all pedestrians in a scene obey the same set of parameters, whereas the multi-behavior aspect of SPM learns the distribution and behavior of multiple types of pedestrians in a single scene.

There are additional parameters which are not learned automatically, they remain fixed throughout all training and experimentation. These values are the exponential sharpness parameter, $\gamma = 20$; and the threshold to determine neighbors by distance, $h = 5$. The number of stereotypes, which will be introduced much later in this dissertation will also be fixed. All other parameters are either trained, or determined by the data (eg: number of pedestrians in a scene, number of groups, length of tracks).

3.2.1 Specific Relationships to Previous Work

Similar to the work of [AMB06], our model is built on pedestrians choosing the next location at each discrete time step. However, in our model, that decision is made by optimizing a function that is continuous in space. This eliminates the need for complicated discretization schemes which trade speed for accuracy.

Our approach is also similar to [JHS08], in that we also learn the parameters of a continuous model. A key difference in our approach lies in how the model is specified. Here, we pose the pedestrian's movement as an energy minimization problem. This enables us to build on previous work on learning parameters for energy functions, such as [Tap07a].

While [PES09] also learns an energy function, we are able to differentiate our loss function with respect to the parameters. This gives us the ability to use gradient descent methods for optimizing our models parameters rather than rely on numerically approximated gradients and/or genetic algorithms used in [PES09]. We believe that having analytically computed gradients will be advantageous for learning systems with a large number of parameters.

3.3 Energy Function

The problem of predicting a pedestrian's path is posed as a series of energy minimizations. The pedestrian's path is modeled as a set of discrete steps. While the path is discretized in time, it is not discretized in space. An energy function allows us to calculate the energy at any location. At the next discrete time step, the pedestrian moves to the location that minimizes this energy. We denote this cost as $E(\mathbf{x}_t)$ where $\mathbf{x}_t$ is a 2D vector containing the pedestrian's location at time t.

The path a pedestrian takes is influenced by the following general motivations. An energy function will describe each of these motivations. The motivations are:

1. A desire to not move too far in a short amount of time. We refer to this at the limited movement term and the energy function expressing this motivation will be denoted as $E_{LM}(\mathbf{x}_t)$.

2. A desire to remain at a constant speed and direction. This motivation will be represented by $E_{CV}(\mathbf{x}_t)$, where CV stands for "constant velocity".

3. A desire to reach one's destination, represented by $E_{\text{Dest}}(\mathbf{x}_t)$

4. A desire to move specifically in relation with those in close proximity $E_{NV}(\mathbf{x}_t)$

5. A desire, if a member of a group, to follow that group $E_{GV}(\mathbf{x}_t)$

6. A desire to avoid other pedestrians in the scene, expressed by $E_{AV}(\mathbf{x}_t)$

The complete energy $E(\mathbf{x}_t)$ is a weighted combination of these components. If the weight of each component is expressed within the component functions (see the following subsections), then the complete energy can be written as:

$$\begin{aligned} E(\mathbf{x}_t) &= E_{LM}(\mathbf{x}_t) + E_{CV}(\mathbf{x}_t) + E_{\text{Dest}}(\mathbf{x}_t) + \\ & \quad E_{NV}(\mathbf{x}_t) + E_{GV}(\mathbf{x}_t) + E_{AV}(\mathbf{x}_t) \end{aligned} \tag{3.1}$$

The following subsections describe each of the energy components listed above. During training, our learning algorithm optimizes a vector of parameters, θ. These weights are expressed within each of the component energy functions. Section 3.4 introduces and discusses the addition of the stereotyping aspect of our model. Section 3.5 describes how the energy function is minimized to generate a predicted track, and Section 3.6 will describe how the parameters can be found by training on tracking data.

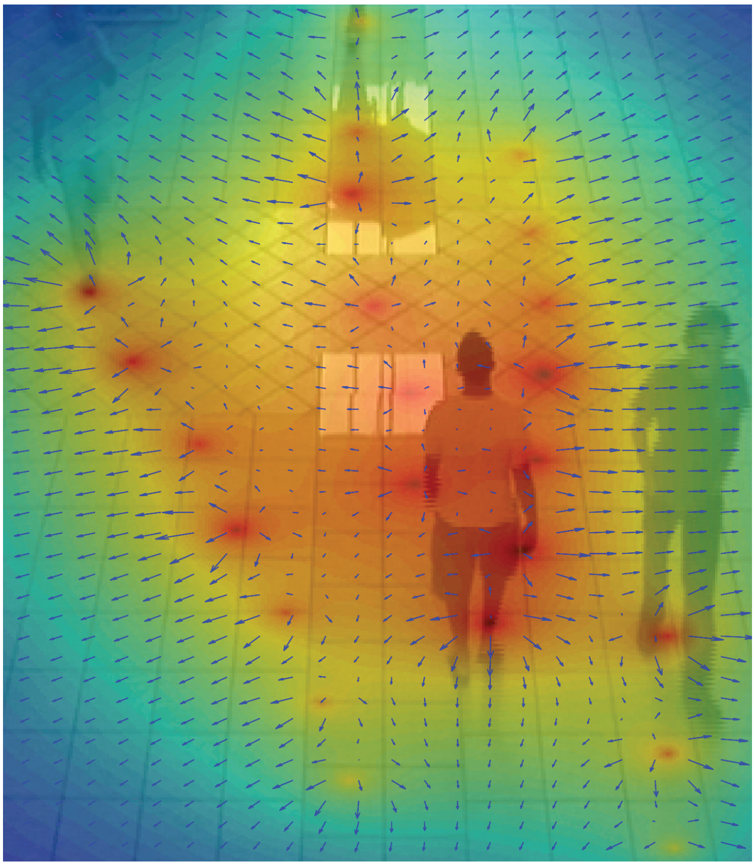


Figure 3.2: The avoidance forces can be seen here as a field which is overlayed on a frame of the video containing four pedestrians. One pedestrian is difficult to see, however his feet can be seen as he is traveling down from the top of the frame. The arrows display the direction and magnitude of the gradient of this avoidance field.

3.3.1 Movement Cost

The movement cost, or cost for moving too far in too short a time, prevents a pedestrian from jumping to a location that is too far away. This cost penalizes all movement, however it balances with other energies to allow reasonable speeds of movement while significantly penalizing physically impossible speeds, due to the polynomial function.

$$E_{LM}(\mathbf{x}_{t+1}) = w(\theta_1)(\mathbf{x}_{t+1} - \mathbf{x}_t)^2 \tag{3.2}$$

The term $w(\theta_1)$ is the weight assigned to this component. In practice, we use an exponential function to compute $w(\theta)$ for all of the components, making $w(\theta_1) = \exp(\theta_1)$. We use the exponential function to ensure that all weights are positive.

3.3.2 Constant Velocity

In our model, a pedestrian also seeks to maintain a constant velocity and direction. This is expressed as an energy function over possible values of $\mathbf{x}_{t+1}$, which is the location of the next step. This function is constructed as a smoothed distance functions between $\mathbf{x}_{t+1}$ and the point that the pedestrian would have reached if maintaining a constant velocity. This point is computed by extrapolating from the previous two steps in the pedestrian's path. The ϵ term, which smooths the function to prevent discontinuity in the derivative at its minimum, is set to 10^{-4}.

$$E_{CV}(\mathbf{x}_{t+1}) = w(\theta_2)\sqrt{||\mathbf{x}_{t+1} - (2\mathbf{x}_t - \mathbf{x}_{t-1})||^2 + \epsilon} \tag{3.3}$$

3.3.3 Neighbor Velocity

A new term, the neighbor velocity is intended to describe how pedestrians in groups or dense crowds might seem to move together. In sparse crowds, or for people who are waiting on something/someone it might describe the opposite. It is the dual nature of this energy function that helps distinguish different behaviors. This energy component takes a similar form to the constant velocity term, it is described by:

$$\begin{aligned} E_{NV}(\mathbf{x}_{t+1}) &= w(\theta_3)\delta(N_n > 0) \times \\ &\quad \sqrt{||\mathbf{x}_{t+1} - (\mathbf{x}_t + NV(\mathbf{x}_t))||^2 + \epsilon} \end{aligned} \tag{3.4}$$

$$NV(\mathbf{x}_t) = \delta(N_n > 0)\frac{\sum_{m=1}^{N_n}\left(\mathbf{p}_t^{m'} - \mathbf{p}_{t-1}^{m'}\right)}{N_n} \tag{3.5}$$

Where there are N_n pedestrians within a certain distance from $\mathbf{x}_t$ and $\mathbf{p}_t^{m'}$ represents the position of each of the neighbors in range. In our experiments we used a threshold, h, that correlated to 5 meters. $\delta(N_n > 0)$ denotes an indicator function that evaluates to 1 if the number of neighbors, N_n, is greater than 0, otherwise it evaluates to 0.

3.3.4 Group Velocity

In addition to individual movement, the model accommodates simple group behavior by incorporating a component encouraging the pedestrian to match the velocity of other people in their group. A central difficulty in incorporating this motivation is that the system must know the group relationships between people in the scene.

Surprisingly, this relationship can be determined with high accuracy using a simple SVM classifier. Using the following set of features we were able to define a feature vector containing only 4 scalar features: minimum distance, maximum distance, mean distance, and duration both pedestrians are observed at the same time. We used these features to train a linear SVM classifier using the LibSVM package [CL01]. Using this approach we were able to predict with 98.38% accuracy on the entire testing set, and 93.22% accuracy on an evenly weighted testing set (equal numbers of group and non-group pairs). This was important since a large majority of the data is made up of individuals who do not move in a group, and by simply classifying all members as being non-group members a system could achieve high accuracy on an unweighted testing set.

For each individual $\mathbf{x}$ at some time t we use the results of the SVM classifier to estimate the set of people belonging to an individual's group and compute an average velocity $GV(\mathbf{x}_t)$ from this set of g pedestrians belonging to the individual's group. We then use the following equations to compute the energy contribution from this part of the model.

$$\begin{aligned} E_{GV}(\mathbf{x}_{t+1}) &= w(\theta_4)\delta(N_g > 0) \times \\ &\quad \sqrt{||\mathbf{x}_{t+1} - (\mathbf{x}_t + GV(\mathbf{x}_t))||^2 + \epsilon} \end{aligned} \tag{3.6}$$

3.3.5 Destination

We hypothesize that pedestrians have some destination in mind and they eventually are observed reaching that destination. It is not accurate to assume that all pedestrians are moving towards a single final destination, so we assume the point where the person exits the scene to be their destination. In applications such as video analysis, destinations may be known and can be used. If not, such as in real-time tracking applications, this force may be left out of the cost calculation. However, if possible to roughly predict a person's destination, that information will greatly improve the ability to accurately predict a person's behavior/motion. Many works have focused on precisely this problem and their results could be incorporated [MPG10] [SBS09b].

Similar to the constant velocity component, described above, we use a smoothed approximation to the radius:

$$E_{\text{Dest}}(\mathbf{x}_{t+1}) = w(\theta_4)\sqrt{||\mathbf{x}_{t+1} - \mathbf{d}||^2 + \epsilon} \tag{3.7}$$

The point $\mathbf{d}$ is the destination found from the track. In Section 3.9.1 we will show how this model can be used without a destination cost to accurately predict a pedestrian's path several time steps ahead.

3.3.6 Avoidance

The previous energies modeled where a pedestrian would like to walk, but it is also important that the model be able to predict areas that the pedestrian would like to avoid. Avoidance is

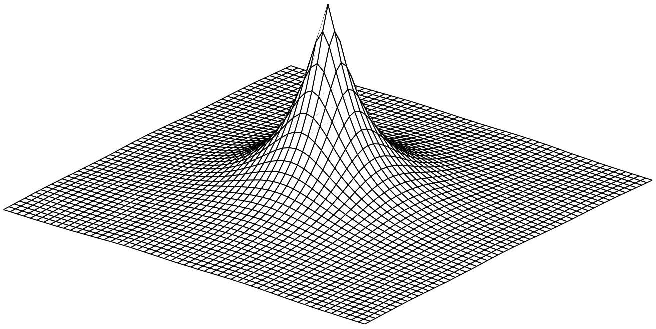


Figure 3.3: The avoidance energy is made up of the sum of avoidance terms at different locations and with different sizes. This function is created from a collection of rotated exponential functions. This makes it convenient to compute convex upper-bounds on this function.

incorporated into the model with a repulsive energy function that goes to zero as one moves away from the center of the function.

The complete avoidance energy, E_{AV}, is the sum of avoidance terms at different locations and of different sizes. Figure 3.3 shows the shape of each individual avoidance term. It is made of a combination of repulsive functions which will be denoted $R(\cdot)$. The repulsive function which is centered at location $\mathbf{p}$, with size parameter σ, has the form

$$R(\mathbf{x}; \mathbf{p}, \sigma) = -\frac{1}{\gamma} \log \left(\sum_{i=1}^{N_\phi} \exp\left(-\gamma e^{-r_i}\right) \right). \tag{3.8}$$

where $r_i = \frac{1}{\sigma}((\mathbf{x}_x - \mathbf{p}_x)\cos(\phi_i) + (\mathbf{x}_y - \mathbf{p}_y)\sin(\phi_i))$. The scalars $\mathbf{x}_x$ and $\mathbf{x}_y$ are the x and y components of $\mathbf{x}$, with a similar notation for $\mathbf{p}$. The angles $\phi_1 \ldots \phi_{N_\phi}$ are uniformly spaced between 0 and 2π. The scalar γ is fixed for all avoidance terms. It affects the sharpness of the fall-off. In practice, we use the value $\gamma = 20$. The combination of these avoidance functions creates a sort of avoidance field which is visualized in Figure 3.2.

The value of this function at a point $\mathbf{x}$ can be thought of as the smooth approximation of the minimum value at $\mathbf{x}$ of a set of rotated exponential functions. This function for the avoidance energy is used in lieu of other functions, like Gaussian, because this function makes it possible to learn the model parameters. As will be discussed in Section 3.5.1.2, it is possible to use Jensen's inequality to compute a convex upper bound on this function. This enables us to use the Variational Mode Learning strategy from [Tap07a] to learn the model parameters.

3.3.6.1 Constructing the Avoidance Component from Terms

If the there are N_o obstacles, $\mathbf{o}_1 \ldots \mathbf{o}_{N_o}$, at time t, an avoidance term is created for each obstacle. This avoidance component itself is the sum of multiple copies of the repulsive function described above. A pedestrian may not avoid just the current location of another individual, but also the location of the individual in the near future. To account for this, repulsive functions are placed at the individual's current location and his predicted location in the future. The number and temporal distance between these predicted locations may vary. The experiments in this chapter use as few as 4 and as many as 6 locations distributed between 0 and 4 seconds in the future. These predicted avoidance locations, denoted as $\mathbf{p}_1, \mathbf{p}_2, \ldots$ are found by assuming that the individual maintains constant velocity.

In addition, it is unknown how far away an individual must be before affecting a pedestrian's path. Therefore, multiple repulsive functions are used with different values of σ at each predicted location of the individual. Thus, if there are N_p predicted locations for each individual and N_σ different size parameters, the avoidance energy due to a single individual will consist of $N_p \times N_\sigma$ total repulsive functions.

To control the number of parameters in the model, assigned weights to each size parameter, θ_{6_j}, and each predicted position time-offset, θ_{7_k}, are trained separately. These two weights are combined to produce the weight of each repulsive function in the avoidance energy. In practice, we multiply the weight due to size by the weight due to predicted position. Combining these weights, the avoidance energy due to a single individual can be expressed as:

$$E_{AV}(\mathbf{x}_{t+1}) = \sum_{j=1}^{N_\sigma} \sum_{k=1}^{N_p} w(\theta_{6_j}) w(\theta_{7_k}) R(\mathbf{x}_{t+1}; \sigma_j, \mathbf{p}_k) \tag{3.9}$$

If multiple individuals are present in the scene, then the avoidance energy is the sum of each individual's avoidance energy. In most of the experiments in this chapter N_σ is 4 and N_p is 5, resulting in 9 avoidance parameters and a total of 14 parameters for the pedestrian model.

3.4 Stereotyping Pedestrians

In any scene, not all pedestrians will act the same. While this variation is common, all previous trained pedestrian motion models have assumed that all pedestrians will obey the same model. This section will show how our model can be improved by identifying different types of pedestrian behavior. This can also be thought of as clustering pedestrians with regard to their behavior. In this work, we refer to the behavior models for different clusters as stereotypes. The remainder of this section focuses on how the model can be formulated with multiple stereotypes.

N_{st} represent the number of stereotype models, each stereotype's model will be denoted as $\theta^1, \theta^2, ..., \theta^{N_{st}}$ since each stereotype is defined by its unique set of parameters θ. With only a single behavioral model (ie: non-stereotyping) the total loss for the model is the sum of the loss for each track computed by

$$L(\mathbf{x}^*, \mathbf{T}) = \sum_{i=1}^{N_s} \sqrt{||\mathbf{x}_t - \mathbf{T}_t||^2 + \epsilon} \tag{3.10}$$

where $x*$ is the predicted track and $\mathbf{T}$ is the ground truth for a track made up of N_s samples. This loss function will be discussed further in Section 3.6.

With the addition of stereotypes, the loss for a particular track becomes the minimum of the loss incurred from the prediction made using each of the stereotypes. The cumulative loss, again, is the sum across all of the tracks,

$$L(\cdot)' = \sum_{i=1}^{N_T} \min(L(\mathbf{x}^i, \mathbf{T}^i; \theta^1), ...L(\mathbf{x}^i, \mathbf{T}^i; \theta^{N_{st}})), \tag{3.11}$$

where $\mathbf{x}^i$ refers to the ith track in the scene and N_T is the number of tracks in the dataset. We also use the prime in this case to denote that this loss function is actually a minimization function on the loss from each stereotype. This distinction will make the explanation clearer in the learning stage. In this way, each pedestrian is described by the model which best matches their own actions. Because the min function is not differentiable, we instead use a continuous approximation to the min:

$$L(\cdot)' = \sum_{i=1}^{N_T} - \log(\sum_{j=1}^{N_{st}} e^{-L(\mathbf{x}^i, \mathbf{T}^i; \theta^j)}) \tag{3.12}$$

3.5 Generating Pedestrian Tracks

A pedestrian's track is generated by minimizing the pedestrian's energy function $E(\cdot)$ to choose the next location. In this section, we describe how this optimization is performed. As Section 3.6 will explain, this procedure is structured to make it possible to compute the derivatives of the predicted path with respect to the model parameters. This makes it possible to minimize the loss function measuring the difference between the predicted pedestrian paths and ground-truth paths.

The optimization procedure uses a modified version of Newton's method, without the backtracking line search, to minimize $E(\cdot)$. Traditionally, Newton's method can be viewed as fitting a second-order Taylor approximation to $E(\cdot)$ at a point, then moving in the direction of the minimum of the approximation. In our implementation, instead of approximating the function directly, a tight, convex upper bound on $E(\cdot)$ is approximated instead. The following subsections describe how these upper-bounds are computed.

Utilizing upper-bounds is necessary because the avoidance penalties described in Section 3.3.6 make $E(\cdot)$ non-convex. Thus, the optimization steps will fail if a point is encountered where the Hessian matrix at that point has negative eigenvalues [1]. Using a convex upper bound on $E(\cdot)$ ensures that this will not happen. While convergence is not guaranteed without the line-search, in our experiments we have not encountered any situations where the optimization does not converge.

Below, the optimization steps are described algorithmically. The variables $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ denote the current and previous locations of the pedestrian. N_I is the number of optimization iterations; convergence was achieved in under 20 steps in all our experimentation. N_s is the number of samples in the track, alternatively it is often called the length of the track. The result of the loop optimization is the pedestrian's location at the next time step, denoted as $\mathbf{x}_{t+1}$. In the course of computing the predicted location, a number of intermediate locations

[1] Intuitively, a second-order approximation at the center of an avoidance term will be a quadratic function pointing downward. Minimizing this approximation will produce values at $+/-\infty$

are computed. These intermediate locations are denoted using the variable $\mathbf{z}$. Using these variables, the generation of a predicted track consists of the following steps:

```
for $l = 1 \dots N_s$ do
    Initialize $\mathbf{z}_0 \leftarrow \mathbf{x}_t$;
    for $j = 1 \dots N_I$ do
        Compute $\hat{E}(\mathbf{z}_j)$ by replacing the component functions with convex upper
        bounds, computed at $\mathbf{z}_{j-1}$;
        Compute $\nabla^2 \hat{E}(\mathbf{z}_j;\theta)$ and $\Delta \hat{E}(\mathbf{z}_j;\theta)$;
        $\mathbf{z}_j \leftarrow \mathbf{z}_{j-1} - \left(\nabla^2 \hat{E}(\mathbf{z}_j)\right)^{-1} \Delta \hat{E}(\mathbf{z}_j)$;
    end
    $\mathbf{x}_{t+1} \leftarrow \mathbf{z}_{N_I}$;
    $t \leftarrow t + 1$;
end
```

The above algorithm will generate a track given a single set of parameters. However, if we wish to generate a track using behavior stereotyping we must follow the algorithm below.

```
1  for $i = 1 \ldots N_{st}$ do
2      for $l = 1 \ldots N_s$ do
3          Initialize $\mathbf{z}_0 \leftarrow \mathbf{x}_t^i$;
4          for $j = 1 \ldots N_I$ do
5              Compute $\hat{E}(\mathbf{z}_j)$ by replacing the component functions with convex upper
               bounds, computed at $\mathbf{z}_{j-1}$;
6              Compute $\nabla^2 \hat{E}(\mathbf{z}_j; \theta^i)$ and $\Delta \hat{E}(\mathbf{z}_j; \theta^i)$;
7              $\mathbf{z}_j \leftarrow \mathbf{z}_{j-1} - \left(\nabla^2 \hat{E}(\mathbf{z}_j)\right)^{-1} \Delta \hat{E}(\mathbf{z}_j)$;
8          end
9          $\mathbf{x}_{t+1}^i \leftarrow \mathbf{z}_{N_I}$;
10     end
11 end
12 $\mathbf{x} \leftarrow \frac{\sum_{i=1}^{N_{st}} \left[ e^{-L(\mathbf{x}_{1:t}^i, \mathbf{T}_{1:t})} \mathbf{x}^i \right]}{\sum_{i=1}^{N_{st}} e^{-L(\mathbf{x}_{1:t}^i, \mathbf{T}_{1:t})}}$;
```

In this algorithm N_{st} is the number of stereotypes, which this dissertation uses at most three. Beyond three stereotypes the improvement to the model is insignificant for the LTA dataset. The addition of the new outer loop generates a track of predictions corresponding to each stereotype where the ith stereotype predicts track $\mathbf{x}^i$. In Step 12 we combine the stereotypes' predictions based on a weighting computed from the loss of the pedestrian's initial track. In practice we hold a small section of the first 4 observations out of all testing. The loss between the initial track, $\mathbf{T}_{1:t}$, and each stereotype's predictions on this initial track, $\mathbf{x}^i_{1:t}$, is used to weight the contribution from the future predictions. Each $\mathbf{x}^i_{1:t}$ can be computed by the first non-stereotyping algorithm in this Section.

3.5.1 Computing Upper Bounds

In Step 4 of the optimization strategy above, upper bounds are computed for all of the quadratic and non-quadratic terms in $E(\cdot)$. This sections describes how the non-quadratic upper bounds are computed.

3.5.1.1 Upper-bounds for Linear Energy Components

E_{Dest}, E_{CV}, E_{NV}, and E_{GV} have the form $\sqrt{r^2+\epsilon}$ where r^2 is some scalar distance. In the case of E_{Dest}, it is the distance to the destination, while for E_{CV}, it is the distance to the point that a pedestrian would move to if traveling with a constant velocity. For these energies, we need to compute a tight quadratic upper bound for our function $f(r) = \sqrt{r^2+\epsilon}$ at a point r_0. As our bound is quadratic, it will have the form $g(r) = ar^2 + br + c$, and obey the following constraints:

$$
\begin{aligned}
g(r_0) &= f(r_0) \\
g'(r_0) &= f'(r_0) \\
g'(0) &= 0
\end{aligned}
\tag{3.13}
$$

We can solve for b, using the third constraint:

$$g'(r_0) = 2a(r_0)^2 + b \tag{3.14}$$

$$b = 0 \tag{3.15}$$

The derivative of f and g can then be used to solve for a.

$$
\begin{aligned}
f'(r_0) &= \tfrac{r_0}{\sqrt{(r_0)^2+\epsilon}} \\
g'(r_0) &= 2ar_0
\end{aligned}
\tag{3.16}
$$

Combining these two equations with the second condition leads to:

$$2ar_0 = \frac{r_0}{\sqrt{(r_0)^2 + \epsilon}} \tag{3.17}$$

$$a = \frac{1}{2}\frac{1}{\sqrt{(r_0)^2 + \epsilon}} \tag{3.18}$$

Finally, we can compute c by plugging into the first condition:

$$\sqrt{(r_0)^2+\epsilon} = \frac{1}{2}\frac{1}{\sqrt{(r_0)^2+\epsilon}}(r_0)^2 + c \tag{3.19}$$

$$c = \sqrt{(r_0)^2+\epsilon} - \frac{1}{2}\frac{(r_0)^2}{\sqrt{(r_0)^2+\epsilon}} \tag{3.20}$$

This leads to the following equation for a tight quadratic upper bound for E_{Dest}, E_{CV}, E_{NV}, and E_{GV}:

$$\frac{1}{2}\frac{r^2}{\sqrt{(r_0)^2+\epsilon}} + \left[\sqrt{(r_0)^2+\epsilon} - \frac{1}{2}\frac{(r_0)^2}{\sqrt{(r_0)^2+\epsilon}}\right] \tag{3.21}$$

3.5.1.2 Upper Bounds for E_{AV}

If we remember the equations for the avoidance energy are defined as:

$$R(\mathbf{x};\mathbf{p},\sigma) = -\frac{1}{\gamma}\log\left(\sum_{i=1}^{N_\phi}\exp\left(-\gamma e^{-r_i}\right)\right). \tag{3.22}$$

$$r_i = \frac{1}{\sigma}((\mathbf{x}_x - \mathbf{p}_x)\cos(\phi_i) + (\mathbf{x}_y - \mathbf{p}_y)\sin(\phi_i)) \tag{3.23}$$

substituting $f_i(x)$ in place of γe^{-r_i} we get:

$$R(\mathbf{x};\mathbf{p},\sigma) = -\frac{1}{\gamma}\log\left(\sum_{i=1}^{N_\phi}\exp\left(-f_i(x)\right)\right). \tag{3.24}$$

$$= -\frac{1}{\gamma}\log\left(\left(\frac{\sum_{j=1}^{N_\phi} e^{-f_j(\lambda)}}{\sum_{j=1}^{N_\phi} e^{-f_j(\lambda)}}\right)\sum_{i=1}^{N_\phi}\exp\left(-f_i(x)+f_i(\lambda)-f_i(\lambda)\right)\right). \tag{3.25}$$

$$= -\frac{1}{\gamma}\log\left(\left(\frac{1}{\sum_{j=1}^{N_\phi} e^{-f_j(\lambda)}}\right)\sum_{i=1}^{N_\phi}\exp\left(-f_i(x)+f_i(\lambda)-f_i(\lambda)\right)\right) - \frac{1}{\gamma}\log\left(\sum_{j=1}^{N_\phi} e^{-f_j(\lambda)}\right). \tag{3.26}$$

We denote the constants which will not effect the derivative with respect to x as K_1. After this substitution we can rewrite Equation 3.26 as:

$$= -\frac{1}{\gamma}\log\left(\sum_{i=1}^{N_\phi}\left(\frac{\exp\left(-f_i(\lambda)\right)}{\sum_{j=1}^{N_\phi} e^{-f_j(\lambda)}}\exp\left(-f_i(x)+f_i(\lambda)\right)\right)\right) + K_1. \tag{3.27}$$

substituting p_i in place of $\frac{\exp -f_i(\lambda)}{\sum_{j=1}^{N_\phi} e^{-f_j(\lambda)}}$ we get:

$$= -\frac{1}{\gamma}\log\left(\sum_{i=1}^{N_\phi}\left(p_i\exp(-f_i(x)+f_i(\lambda))\right)\right) + K_1. \tag{3.28}$$

using Jensen's inequality we can bound the above as:

$$\leq -\frac{1}{\gamma}\sum_{i=1}^{N_\phi}\left(p_i\log\left(\exp((-f_i(x)+f_i(\lambda)))\right)\right) + K_1. \tag{3.29}$$

$$\leq -\frac{1}{\gamma}\sum_{i=1}^{N_\phi}\left(p_i(-f_i(x))\right) - \frac{1}{\gamma}\sum_{i=1}^{N_\phi}\left(p_i(f_i(\lambda))\right) + K_1. \tag{3.30}$$

Adding the constant terms together, we denote them collectively as K. This results in the simplified equation:

$$\leq -\frac{1}{\gamma}\sum_{i=1}^{N_\phi}(p_i(-f_i(x))) + K. \tag{3.31}$$

substituting back in for p_1:

$$\leq -\frac{1}{\gamma}\sum_{i=1}^{N_\phi}\left(\frac{\exp -f_i(\lambda)}{\sum_{j=1}^{N_\phi} e^{-f_j(\lambda)}}(-f_i(x))\right) + K. \tag{3.32}$$

substituting back in for $f_i(x)$:

$$\leq -\frac{1}{\gamma}\sum_{i=1}^{N_\phi}\left(\frac{\exp -\gamma e^{-r'_i}}{\sum_{j=1}^{N_\phi} e^{-\gamma e^{-r'_j}}}(-\gamma e^{-r_i})\right) + K. \tag{3.33}$$

$$\leq \sum_{i=1}^{N_\phi}\left(\frac{\exp -\gamma e^{-r'_i}}{\sum_{j=1}^{N_\phi} e^{-\gamma e^{-r'_j}}}(e^{-r_i})\right) + K. \tag{3.34}$$

3.6 Learning

Our goal is to choose the parameters θ that make the predicted pedestrian tracks match tracks observed in video as closely as possible. To accomplish this, we define a loss function $L(\mathbf{x}^*, \mathbf{T})$ that measures the difference between a predicted track $\mathbf{x}^*$ and the ground truth track $\mathbf{T}$. This loss function was introduced in Section 3.4 as Equation 3.10 to introduce stereotyping. Here we will go into detail about how the loss is used to learn the parameters of the model. Remember that the loss function is a smoothed version of the L_1 difference between the ground-truth track and the predicted track where $\mathbf{x}_t$ and $\mathbf{T}_t$ are locations of pedestrians and ground-truth at time t, and N_s is the number of samples, or length of the track.

Because the loss depends on the predicted path, $\mathbf{x}^*$, the loss can be minimized with gradient-based optimization methods if the derivatives of $\mathbf{x}^*$ can be computed with respect to the parameters θ. The optimization strategy described in Section 3.5 is designed to make these computations possible.

Computing the gradient of the loss is possible because an intermediate value during the optimization, $\mathbf{z}_j$ is related to the previous value, $\mathbf{z}_{j-1}$, by multiplication with an inverse matrix. Thus, the Jacobian matrix $\frac{\partial \mathbf{z}_j}{\partial \mathbf{z}_{j-1}}$ relating $\mathbf{z}_j$ and $\mathbf{z}_{j-1}$ can be found. Using this Jacobian, combined with $\frac{\partial \mathbf{z}_{j-1}}{\partial \theta}$, it is then possible to compute $\frac{\partial \mathbf{z}_j}{\partial \theta}$. These basic steps can be repeated until the derivative of each step $\mathbf{x}_t$ with respect to each of the parameters in θ has been computed. With these derivatives, it is trivial to compute the derivative of the loss function with respect to θ.

As in Variational Mode Learning, because each optimization step is differentiable, the gradient of the result of the optimization can be computed by repeated application of the chain rule, similar to back-propagation in neural networks.

For clarity, the following section shows how these derivatives can be calculated for the model parameters. The algorithm used for computing the derivative of the loss function with respect to the parameters, θ, of a non-stereotyping model is:

```
Initialize $\mathbf{x}_0$ to the initial point on the track;
Initialize $\frac{\partial \mathbf{x}_0}{\partial \theta}$ to a $2 \times N_\theta$ matrix, where $N_\theta$ is the number of parameters in the entire
  non-stereotyping model;
for $t = 1 \ldots N_S$ do
    $\mathbf{z}_0 \leftarrow \frac{\partial \mathbf{x}_{t-1}}{\partial \theta}$;
    for $j = 1 \ldots N_I$ do
        Compute $\frac{\partial \mathbf{z}_j}{\partial \mathbf{x}_{t-1}}$, $\frac{\partial \mathbf{z}_j}{\partial \mathbf{z}_{j-1}}$ , and $\frac{\partial \mathbf{z}_j}{\partial \mathbf{x}_{t-2}}$;
        Compute $\frac{\partial \mathbf{z}_j}{\partial w}\frac{\partial w}{\partial \theta}$ (See below for explanation);
        $\frac{\partial \mathbf{z}_j}{\partial \theta} \leftarrow \frac{\partial \mathbf{z}_j}{\partial \mathbf{z}_{j-1}}\frac{\partial \mathbf{z}_{j-1}}{\partial \theta} + \frac{\partial \mathbf{z}_j}{\partial w}\frac{\partial w}{\partial \theta}$;
        $\frac{\partial \mathbf{z}_j}{\partial \theta} \leftarrow \frac{\partial \mathbf{z}_j}{\partial \theta} + \frac{\partial \mathbf{z}_j}{\partial \mathbf{x}_{t-1}}\frac{\partial \mathbf{x}_{t-1}}{\partial \theta} + \frac{\partial \mathbf{z}_j}{\partial \mathbf{x}_{t-2}}\frac{\partial \mathbf{x}_{t-2}}{\partial \theta}$;
    end
    $\frac{\partial \mathbf{x}_t}{\partial \theta} \leftarrow \frac{\partial \mathbf{z}_{N_I}}{\partial \theta}$;
    $\frac{\partial L}{\partial \theta} \leftarrow \frac{\partial L}{\partial \theta} + \frac{\partial L}{\partial \mathbf{x}_t}\frac{\partial \mathbf{x}_t}{\partial \theta}$;
end
```

The matrix $\frac{\partial \mathbf{z}_j}{\partial w}\frac{\partial w}{\partial \theta}$ appears because the energy function is the sum of a set of component energy functions, each with its own weight, w_c. These weights are generated from the parameters θ. The terms $\frac{\partial \mathbf{z}_j}{\partial \mathbf{x}_{t-1}}$ and $\frac{\partial \mathbf{z}_j}{\partial \mathbf{x}_{t-2}}$ appear because of the inertial and constant velocity components of the energy function. These components involve the location of previous steps in the pedestrian track.

For learning the stereotyping model, we can think of the parameter vector θ as containing multiple separate parameter sets where the ith set is denoted as θ^i. Then the algorithm for computing the derivative of the loss function with respect to the parameters θ for the stereotyping model is:

```
for $i = 1 \dots N_{st}$ do
    Initialize $\mathbf{x}_0$ to the initial point on the track;
    Initialize $\frac{\partial \mathbf{x}_0}{\partial \theta^i}$ to a $2 \times N_{\theta^i}$ matrix, where $N_{\theta^i}$ is the number of parameters in the
    $i$th stereotype;
    for $t = 1 \dots N_S$ do
        $\mathbf{z}_0 \leftarrow \frac{\partial \mathbf{x}_{t-1}}{\partial \theta^i}$;
        for $j = 1 \dots N_I$ do
            Compute $\frac{\partial \mathbf{z}_j}{\partial \mathbf{x}_{t-1}}$, $\frac{\partial \mathbf{z}_j}{\partial \mathbf{z}_{j-1}}$ , and $\frac{\partial \mathbf{z}_j}{\partial \mathbf{x}_{t-2}}$;
            Compute $\frac{\partial \mathbf{z}_j}{\partial w}\frac{\partial w}{\partial \theta^i}$ (See below for explanation);
            $\frac{\partial \mathbf{z}_j}{\partial \theta^i} \leftarrow \frac{\partial \mathbf{z}_j}{\partial \mathbf{z}_{j-1}}\frac{\partial \mathbf{z}_{j-1}}{\partial \theta^i} + \frac{\partial \mathbf{z}_j}{\partial w}\frac{\partial w}{\partial \theta^i}$;
            $\frac{\partial \mathbf{z}_j}{\partial \theta^i} \leftarrow \frac{\partial \mathbf{z}_j}{\partial \theta^i} + \frac{\partial \mathbf{z}_j}{\partial \mathbf{x}_{t-1}}\frac{\partial \mathbf{x}_{t-1}}{\partial \theta^i} + \frac{\partial \mathbf{z}_j}{\partial \mathbf{x}_{t-2}}\frac{\partial \mathbf{x}_{t-2}}{\partial \theta^i}$;
        end
        $\frac{\partial \mathbf{x}_t}{\partial \theta^i} \leftarrow \frac{\partial \mathbf{z}_{N_I}}{\partial \theta^i}$;
        $\frac{\partial L}{\partial \theta^i} \leftarrow \frac{\partial L}{\partial \theta^i} + \frac{\partial L}{\partial \mathbf{x}_t}\frac{\partial \mathbf{x}_t}{\partial \theta^i}$;
    end
    Compute $L(\mathbf{x}, \mathbf{T}; \theta^i)$, the loss between the predicted track given the $i$th
    stereotype's parameters and the ground truth;
end
for $i = 1 \dots N_{st}$ do
    $\frac{\partial L'}{\partial \theta^i} \leftarrow \frac{\exp\{-L(\mathbf{x},\mathbf{T};\theta^i)\}}{\sum \exp\{-L(\mathbf{x},\mathbf{T};\theta)\}}\frac{\partial L}{\partial \theta^i}$;
end
```

Step 18 is weighting the contribution of the derivative of the stereotyping loss, $L'(\cdot)$, for the ith stereotype's parameters by the loss of the ith stereotype's prediction. Stereotypes need to describe different subsets of pedestrians, however there is the question of initialization. Randomly perturbing an initial parameter is satisfactory, and when using this method for initialization the loss does converge; however by creating small random distinct subsets of pedestrians from the training set and running a short bootstrap training, the model can converge in fewer overall steps. In most experiments of this paper the second method was used to initialize the parameters, but both methods have proven to work well in practice.

3.6.1 Deriving Derivatives for E_{Dest}

In this section we will describe how to compute the derivative for E_{Dest}. We refer the reader to [Tap07a] for more information about deriving the derivatives of energy functions similar to those of this method.

The Newton step in the optimization procedure described in Section 3.5 can be thought of as minimizing a second-order approximation of $\hat{E}(\cdot)$, the upper bound on $E(\cdot)$. In this subsection, the solution to this second order approximation will be denoted as $\mathbf{z}_j = A^{-1}\mathbf{h}$, where $\mathbf{z}_j$ is one of the intermediate steps in the optimization.

Because $E(\cdot)$ is the sum of the individual component functions, such as E_{Dest}, the second-order approximation of $E(\cdot)$ is the sum of the second order approximations of the individual energy components. Thus A is actually the sum of matrices, with one matrix for each of the energy components. The vector $\mathbf{h}$ is likewise a sum.

The contribution of $\hat{E}_{\text{Dest}}$ to A can be found by noting that the upper bound on the destination component, $\hat{E}_{\text{Dest}}$, is itself quadratic. It can be expressed in the form

$$\hat{E}_{\text{Dest}}(\mathbf{z}_j; \mathbf{z}_{j-1}) = \frac{1}{2} e^{\theta_5} (\mathbf{z}_j - \mathbf{d})^T \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix} (\mathbf{z}_j - \mathbf{d}) \tag{3.35}$$

where $\mathbf{d}$ is the destination of the pedestrian, $a = (\sqrt{||\mathbf{z}_{j-1} - \mathbf{d}||^2 + \epsilon})^{-1}$, and the term e^{θ_5} is the weight assigned to this component of the energy function. The exponential is used to ensure that all weights are positive.

Differentiation of this quadratic system makes it possible to compute the contribution to A and $\mathbf{h}$, which will be as denoted A_{Dest} and $\mathbf{h}_{\text{Dest}}$, as

$$A_{\text{Dest}} = e^{\theta_5} \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix} \qquad \mathbf{h}_{\text{Dest}} = e^{\theta_5} \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix} \mathbf{d} \tag{3.36}$$

As the second-order approximation $\hat{E}(\cdot)$ is the sum of components from the different motivations, the derivative $\frac{\partial \mathbf{z}_j}{\partial \mathbf{z}_{j-1}}$ is the sum of terms corresponding to each of these components. We denote the contribution of the destination component $E_{\text{Dest}}(\cdot)$ as

$$\frac{\partial \mathbf{z}_j^{\text{Dest}}}{\partial \mathbf{z}_{j-1}} = -e^{\theta_5} A^{-1} \text{diag}(\mathbf{z}_j - \mathbf{d}) \left[\frac{\partial \mathbf{a}}{\partial \mathbf{z}} \right] \tag{3.37}$$

where $\mathbf{a} = \begin{bmatrix} a \\ a \end{bmatrix}$ using a as defined above and diag($\mathbf{z}_j - \mathbf{d}$) is a diagonal matrix with the vector $(\mathbf{z}_j - \mathbf{d})$ placed along the diagonal. The derivation of this contribution can be found in the the following subsection.

The derivative $\frac{\partial \mathbf{z}_j}{\partial w}\frac{\partial w}{\partial \theta}$ can be computed in a similar fashion:

$$\frac{\partial \mathbf{z}_j}{\partial w}\frac{\partial w}{\partial \theta} = -A^{-1}\begin{bmatrix} e^{\theta_5}a & 0 \\ 0 & e^{\theta_5}a \end{bmatrix}(\mathbf{z}_j - \mathbf{d}) \tag{3.38}$$

3.6.2 Derivations from Section 3.6.1

In this subsection, we describe the details in deriving the derivatives from Section 3.6.1, using the notation from that section. When computing the derivatives, we will rely on the identity

$$\frac{\partial A^{-1}}{\partial \theta} = -A^{-1}\frac{\partial A}{\partial \theta}A^{-1} \tag{3.39}$$

We will begin by finding $\frac{\partial \mathbf{z}_j}{\partial w}\frac{\partial w}{\partial \theta}$. Using the product rule, this is equal to $\frac{\partial A^{-1}}{\partial \theta_5}\mathbf{h} + A^{-1}\frac{\partial \mathbf{h}}{\partial \theta_5}$. The first term in this sum is can be rewritten as

$$\begin{aligned} \frac{\partial A^{-1}}{\partial \theta_5}\mathbf{h} &= -A^{-1}\begin{bmatrix} e^{\theta_5}a & 0 \\ 0 & e^{\theta_5}a \end{bmatrix}A^{-1}\mathbf{h} \\ &= -A^{-1}\begin{bmatrix} e^{\theta_5}a & 0 \\ 0 & e^{\theta_5}a \end{bmatrix}\mathbf{z_j} \end{aligned} \tag{3.40}$$

The derivation of the second term, $A^{-1}\frac{\partial \mathbf{h}}{\partial \theta_5}$ is straightforward, leading to Equation 3.38.

The derivative $\frac{\partial \mathbf{z}_j^{\mathrm{Dest}}}{\partial \mathbf{z}_{j-1}}$ is computed in a similar fashion. Defining a matrix $W = \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix}$, the primary difficulty is that W must be differentiated with respect to both components of

$\mathbf{z}_{j-1}$. Following a process similar to that just shown,

$$\begin{aligned}\frac{\partial A^{-1}}{\partial \mathbf{z}^x_{j-1}}\mathbf{h} &= -e^{\theta_5}A^{-1}\frac{\partial W}{\partial \mathbf{z}^x_{j-1}}A^{-1}\mathbf{h} \\ &= -e^{\theta_5}A^{-1}\frac{\partial W}{\partial \mathbf{z}^x_{j-1}}\mathbf{z_j} \\ &= -e^{\theta_5}A^{-1}\mathrm{diag}(\mathbf{z_j})\frac{\partial \mathbf{a}}{\partial \mathbf{z}^x_{j-1}} \qquad (3.41)\end{aligned}$$

where $\mathbf{a}$ is defined as in Section 3.6.1 and $\mathbf{z}^x_{j-1}$ is the component of that vector that refers to the x, or horizontal, position of the pedestrian. Expressing the derivative in this form makes it convenient to compute the derivative with respect to both components of $\mathbf{z}_{j-1}$:

$$\frac{\partial A^{-1}}{\partial \mathbf{z}_{j-1}}\mathbf{h} = -e^{\theta_3}A^{-1}\mathrm{diag}(\mathbf{z_j})\frac{\partial \mathbf{a}}{\partial \mathbf{z}_{j-1}} \qquad (3.42)$$

The entire derivative $\frac{\partial \mathbf{z}_j^{\mathrm{Dest}}}{\partial \mathbf{z}_{j-1}}$ can be found by using a similar set of steps to find $A^{-1}\frac{\partial \mathbf{h}}{\partial \mathbf{z}_{j-1}}$

3.7 Evaluation Baselines

3.7.1 Datasets

We evaluated our model using two datasets. The first is from *You'll Never Walk Alone: Modeling Social Behavior For Multi-Target Tracking* [PES09], commonly known as LTA, and contains a total of 745 tracks from two separate outdoor scenes. This dataset has been used in many recent pedestrian tracking and modeling publications [KAO11] [PEG10] [PEV10]. Because the term LTA can be used to identify either the method or the dataset, we have tried to avoid this ambiguity by specifying them explicitly whenever context is not

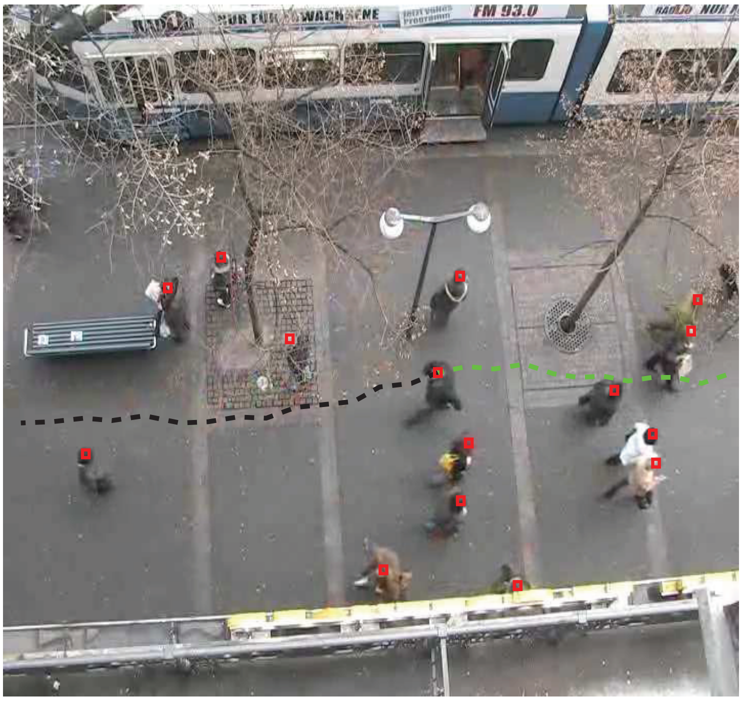


Figure 3.4: A sample from the LTA dataset displaying a single pedestrian's track. The pedestrian's past track is colored green and is used to assign the pedestrian's behavior stereotype when predicting. The future track is colored black and shows how the person avoids others in the scene.

enough to make such a distinction clear. In the publicly available dataset, annotation and homography for rectification are provided. The first 150 tracks in each scene are used for training and the remaining are used for testing. It is important that the training and testing data be as continuous as possible so that any two pedestrians which exist at the same time belong to the same set. All training and testing splits use such a first/last split rather than a random selection for this reason.

The second dataset used for evaluation is from *Learning Pedestrian Dynamics From The Real World* [ST09] which we will refer to as LPD. Similar to LTA, this publication contains both a novel method and a unique dataset. The term LPD can refer to either but will be specified to remove ambiguity. The LPD dataset contains 92 tracks. It is an indoor scene and contains a relatively large number of avoidance maneuvers due to the proximity of people and the resolution of the data. This dataset is annotated twice, once using manual annotations and again using a background modeling and object tracking algorithm. In this dataset the first 32 tracks are used for training and the remaining is used for testing purposes.

Sections 3.8 and 3.9 will discuss experiments on these two datasets.

3.7.2 Baseline Models

Multiple baseline models are used for comparison. This section will outline the models which will be used in the experiments in the following sections.

Constant velocity assumptions are very popular in tracking literature. We use a simple constant velocity assumption which predicts a future location based solely on two previous locations which we refer to as CV_1 in the experiments.

We also use a method which assumes the previous location history is a set of noisy observations to a model which maintains a constant velocity. This method can be thought of as being similar to the Kalman filtering approach, with the exception that we do not update the model over time. Rather a linear model is fit to the known observations and all future predictions are computed using only these known observations; we refer to this model as CV_2 in the following experiments.

Social force models used for comparison include the LTA method [PES09] and the LPD method [ST09]. We used a MATLAB implementation of the LTA model in order to produce results on the LTA data. However, we did not implement a genetic algorithm to train the LTA method. The original paper provides the parameters for a trained model. This means that the LTA method may have an strong advantage of being trained on the LTA data which we use to test the other methods. Despite this advantage, our method still performs well in comparison.

The LPD method is very similar to a single stereotype of the SPM method (eg: ST_1). The difference is that the LPD method only contains four motivations; the SPM method contains an additional two motivations. The SPM method is also broken down for comparison into its component stereotypes: ST_1, ST_2, and ST_3. In these models the pedestrian is forced to obey only the parameters of a single stereotype model.

3.8 Stereotyping Results

We will evaluate the SPM method with multiple stereotypes as well as without stereotyping, which is similar to the method in [ST09]. This section will discuss the results of the stereotyping method know as SPM.

The dataset used in this section is provided by [PES09] and contains a total of 745 tracks from two separate outdoor scenes. Annotation and homography for rectification are provided. The first 150 tracks from each scene are used for training and the remaining are used for testing. It is important that the training and testing data be as continuous as possible so that any two pedestrians which exist at the same time belong to the same set. The existence of pedestrians from training and testing set simultaneously appearing in a scene would result in trained pedestrian tracks being used as obstacles, or as incomplete data for training since the testing tracks would not be able to be used as obstacles in training. Therefore it is important that the two sets be distinct in the times which they occur.

3.8.1 LPD versus Stereotyped Models

Our first comparison compares the LPD model from [ST09] with the enhanced models proposed here. The LPD model used 4 motivations and was published in [ST09]. The next step up is the ST_1 model, a single-stereotype but 6 motivation method. The final model is the full SPM model which uses multiple 3 behavior stereotypes, each containing 6 motivation component energies. Table 3.1 shows the testing loss on the LTA testing data. While the ST_1 model which uses the two extra motivations of Group Velocity and Neighbor Velocity improve testing loss by a modest 5%, the SPM model is able to reduce the testing loss by 52%.

Testing Loss Reduction by Method		
Model	Cumulative Loss	Reduction from Baseline
LPD	5.75e02	0.00%
ST_1	5.45e02	5.22%
SPM	2.74e02	52.34%

Table 3.1: Testing error for different models. Error was calculated by the above loss function and computed in the coordinate space found by the publicly available homography projection for the dataset. ST_1 refers to a single stereotype, and SPM refers to a three stereotype model.

3.8.2 Comparison with LTA and Baseline Models

The LTA dataset makes it possible to compare the models proposed here with the LTA approach from [PES09]. We follow the methodology in [PES09] by measuring the number of correctly predicted trajectories. If a model's predicted trajectory never varies from the ground truth by more than a certain threshold, then the predicted trajectory is considered correct. The accuracy threshold is varied creating multiple performance characteristics where the best performance is at the top-left corner.

Because the LTA method was trained on the entire dataset, we compare the accuracy of the LTA model on this dataset over both the training and testing subsets, described above, that were created to train our models. Figure 3.5 shows that the ST_1 model, which does not include the stereotypes, performs comparably to the LTA model on the training set. On the testing set, Figure 3.6 shows that the baseline models, such as ST_1 which performed similarly

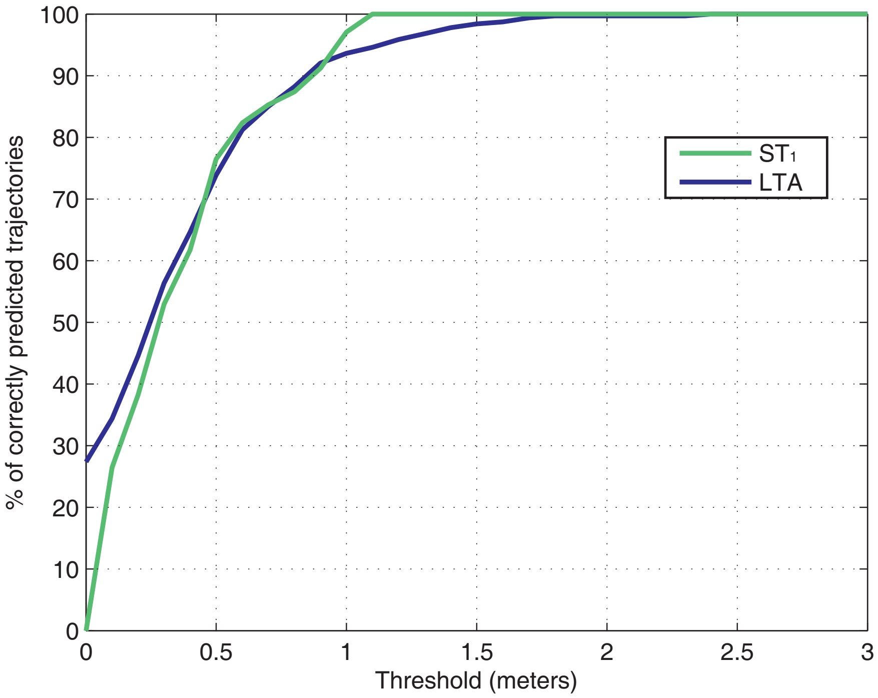


Figure 3.5: Performance of a Non-stereotyping model ST_1 and the LTA model on their training data. The LTA method performs similarly to the ST_1 method.

to LTA, are not able to correctly predict as many tracks as SPM. In fact, the performance of SPM on its testing data is quite similar to that of LTA on its training data.

Comparisons between the stereotyped model, SPM, and the other baseline models are performed solely on the test set. In Figure 3.6 we compare various prediction methods. The worst performance is achieved by the simple linear prediction of constant velocity CV_1. This method assumes that a person will continue at the same speed and direction that was observed between the last two known observations. The performance of many of the other methods is very similar. These methods include CV_2 which takes the last four observations

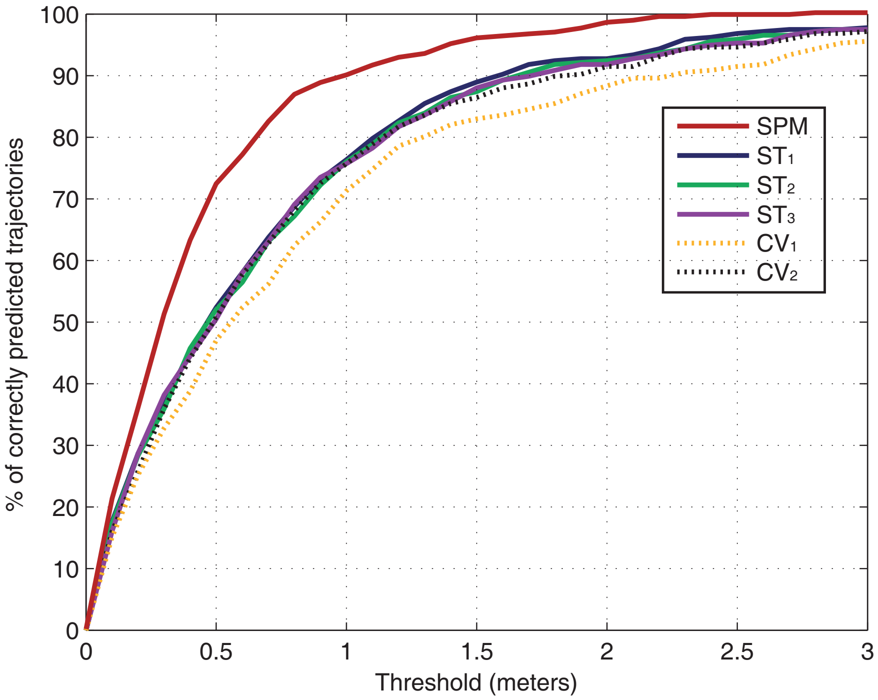


Figure 3.6: Comparison results for SPM as well as its component models against baseline models on the testing set. CV_2 performs almost as well as the component models, and similar to the results in [PES09], while SPM outperforms all other models by a significant margin. SPM performs similarly on the testing set in this figure as ST_1 and LTA perform on the training set.

and assumes an underlying linear motion model similar to a Kalman filter. This method was explained in Section 3.7.2. Each of the three individual models also perform similarly, the variation is due to the fact that each of these component models is trained on a different section of the training data. Above all of these methods is the SPM model which is made from the three component models: ST_1, ST_2, and ST_3.

					Avoidance								
LM	CV	NV	GV	Dest	σ_1	σ_2	σ_3	σ_4	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
0.000	1.170	1.241	0.951	1.547	-0.249	-0.280	-0.280	-0.184	-0.359	-0.292	-0.056	-0.037	0.001
0.000	4.174	3.916	4.491	4.827	-3.576	-4.260	-4.309	-4.246	-3.021	-4.879	-4.116	-3.269	-1.856
0.000	2.509	0.890	1.942	0.716	-0.460	-0.576	-0.458	-0.246	-0.701	-0.431	-0.157	-0.082	-0.119

Table 3.2: Trained model parameters, each line defines a stereotype. The first stereotype tends to describe most individuals who are attempting to make the best time towards their destination. The second describes many of the groups of pedestrians. You can see this by the significantly smaller avoidance terms, as well as the high values for group and neighbor velocity. The third stereotype tends to describe many of the outliers, pedestrians who do not belong in either of the first two groups.

3.8.3 Qualitative Analysis of Stereotype Assignment

One interesting aspect of our training is that stereotype assignments are very fluid. The majority of training tracks changed their initial stereotype at least once during the training process. This is important because if tracks were unable to change their initial stereotype then a simpler model which merely trained three separate models would have sufficed. Figure 3.6 shows this clearly, as SPM is a combination of ST_1, ST_2, and ST_3, yet it is able to improve the prediction rate by as much as 15% over the best component model. This proves that the stereotypes come from the data in an unsupervised fashion, and the multi-behavior model as a whole is significantly better than its parts.

3.8.3.1 Ability to Stereotype

While we did not bias our model toward any specific stereotypes, we noticed that certain clusters emerged from the SPM training which were qualitatively describable. The most common was the individual. Often walking quickly and dodging groups of pedestrians, these pedestrians made up a significant portion of the data in each of the datasets. Next were people walking in groups or pushing a stroller. Lastly were individuals who seemed to aimlessly wander or remain relatively stationary for periods of time. The third cluster contained much fewer pedestrians and their similarities were not always as apparent. This can be seen in Figure 3.7. The first two images in the figure show the most common clusters, the last image contains all pedestrians belonging to the smallest cluster. In this scene all four labeled pedestrians are traveling in very different paths in close proximity, only two of the four are traveling along the path of the sidewalk which the majority of other pedestrians in the scene follow.

Table 3.2 displays the parameters resulting from training on the LTA dataset. Because we are minimizing an energy function these weights are relative, and the least movement term is held constant and the other parameters are allowed to vary. The relative parameter weights are quite unique to each stereotype and even hint at qualitative assignments to individuals and groups. For example, the first stereotype has a relatively low group velocity (GV) weight and the highest avoidance weights (AV), qualitatively describing individuals who pay attention to avoiding their surroundings but care more about following their current neighbors than pedestrians estimated to be in their group. The second stereotype has the

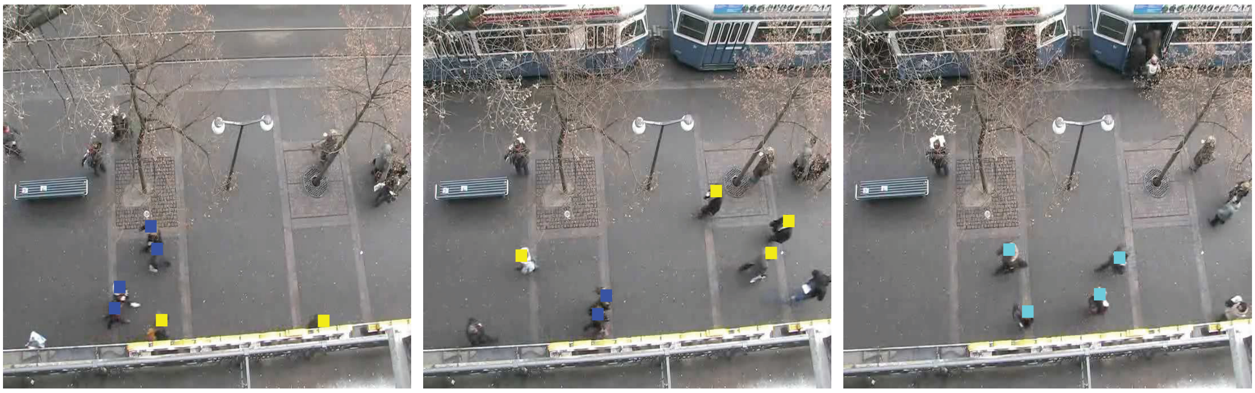


Figure 3.7: Pedestrians are labeled by their assigned stereotype based on their past motion history. Most pedestrians are assigned to the yellow stereotype which seems to describe individuals. The second most popular stereotype, labeled in blue, tends to favor pedestrians in groups. The least common stereotype, labeled in cyan, occurs infrequently, but in the case of the last frame it occurs multiple times in a single scene when behavior is not normal. In the last frame all four pedestrians just dodged each other as they travel in generally the up/down/left/right direction in close proximity.

lowest avoidance weights across the board and more reliance on group velocity than constant velocity (the only stereotype to do so). This second stereotype seems to describe pedestrians who belong to a group. The last stereotype seen in Table 3.2 relies primarily on the constant velocity (CV) component, a common fall-back model which does not seem to describe any specific behavior but is generally true. These behaviors would seem to confirm what we notice in Figure 3.7.

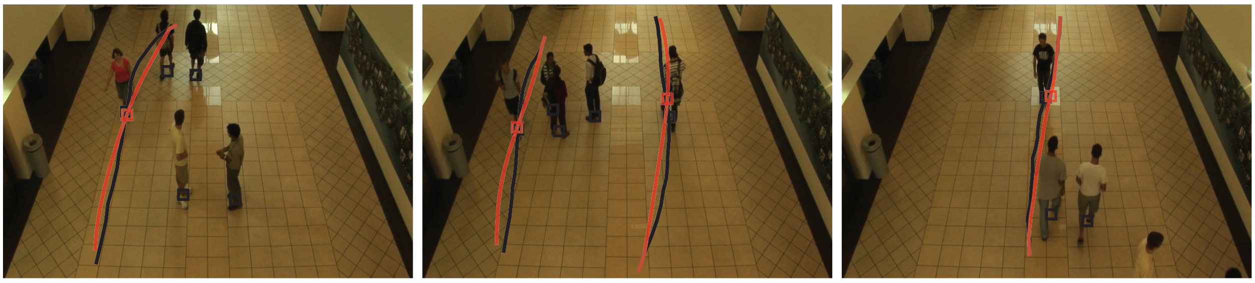


Figure 3.8: Examples of pedestrian paths, shown in black, and predicted paths, shown in red. The model accurately predicts the deflection of pedestrians due to oncoming obstacles.

3.9 Non-Stereotyping Results

The LPD dataset was recorded in a hallway/entrance area of a building. Two sets of ground-truth data were recorded, one which was manual and another which was generated by a background subtraction and object detection/tracking system which required no human input other than setting initial parameters [JS02]. This second set of ground-truth was noisy, however we felt it was important to show that our pedestrian model was able to handle tracks resulting from object tracking algorithms. This dataset contained only 92 tracks, significantly fewer than the outdoor scenes, however they were of greater fidelity due to the higher resolution and smaller field of view.

Image coordinates are mapped to a rectified coordinate space using a simple homography and all learning/prediction is done in the real world coordinate system. A point on the top of the head which was used in the LTA dataset is easier to track; however these points would not be moving on the same plane, as different people are different heights causing errors in the mapping between coordinate systems. Due to the angle and resolution of this video, this noise would be unacceptable.

One third of the tracks were used for training and two-thirds were used for testing. In our first experiment, we used the model to predict each pedestrian's path, given the individual's initial position, velocity, and the locations of the obstacles.

3.9.1 Loss from Automatic Tracking Results

When tracks are lost in many tracking applications, they are assumed to continue in their previously known direction. Here, we show how our model can be used to predict a pedestrian's future position in the case where the position cannot be obtained from image data.

Because the system will not be working from whole tracks, the destination is not known, and for these experiments the corresponding θ_5 parameter is held to $-\infty$, thus $w(\theta_5) = 0$. The model is trained to independently predict, at every time step, the next single step as accurately as possible. This model was trained using tracks automatically generated by the algorithm described in [JS02].

We compared the predictions from our model against predictions formed using only the assumption that the pedestrian maintains their last known trajectory. In these predictions, the pedestrian follows a straight-line defined by the previous two steps. We also experimented with splines fit to the previous path, but found that the straight-line prediction performed better.

Length	CV_1	LPD_{-D}	Improvement
1	0.466 m	0.375 m	19.47%
2	0.784 m	0.575 m	26.62%
3	1.066 m	0.740 m	30.61%
4	1.355 m	0.905 m	33.23%
5	1.638 m	1.075 m	34.34%
6	1.962 m	1.261 m	35.76%

Table 3.3: Length denotes the number of time steps the model must predict; The middle two columns show the drift from the ground truth measured in meters after the given length of time. LPD_{-D} denotes that the model does not contain the Destination cost; Improvement is the percentage decrease in error from the baseline CV_1 model to the LPD_{-D} model.

Table 3.3 shows the total error of the various methods over several time lengths. The error in the estimates of future positions are significantly reduced by using the pedestrian model, which has avoidance terms in addition to the constant velocity assumption. Notice that our model offers greater improvement the further ahead that the prediction must be made.

3.9.2 Avoidance Field

As discussed in Section 3.3.6, multiple values for the σ parameter are each given their own weight, essentially learning the size of the radius of influence that one pedestrian exerts on another in the scene. Similarly, multiple avoidance locations are also used to aid in the

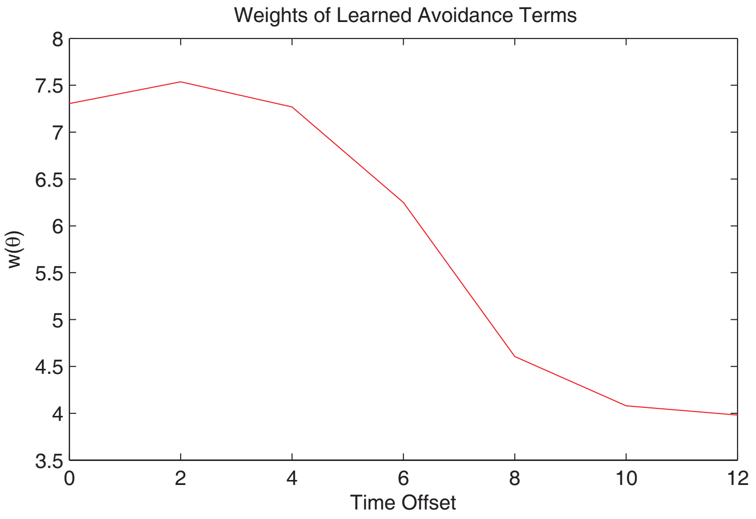


Figure 3.9: Learned parameter values corresponding to the multiple avoidance locations. A time offset of 2 corresponds to .8 seconds.

accuracy of the model. The combination of these avoidance terms creates a sort of field, which was visualized in Figure 3.2.

After training, the θ_7 weights showed that on this dataset a model which describes a single behavior places more weight on avoiding a future predicted location of the obstacles in the scene. Figure 3.9 shows the corresponding weights of the avoidance terms. It is unexpected to see that an error prone estimation is more important to avoid than the observed location of the obstacle; however, it does make sense since the pedestrians themselves are trying to best chose their future location at the next time step. This also supports the idea that the avoidance function should be a multi-parameter function, such as SPM and LPD.

3.10 Summary

This chapter has presented a method for automatically learning parameters for pedestrian models from real world observations, and this method allows for pedestrians to be elegantly clustered by their behaviors. We have shown that the unsupervised clustering of pedestrian behavior stereotypes does result in more accurate motion predictions. Our method has been tested on multiple datasets and has been shown to be more accurate than standard methods. Our learning method is able to optimize a magnitude more parameters than has been shown by other works.

CHAPTER 4: PEDESTRIAN TRACKING USING MOTION PRIORS

4.1 Introduction

This chapter will explore tracking using motion priors, including our own multi-behavior pedestrian model. Appearance based tracking is a significant subfield of computer vision, and many different approaches exist. We will narrow the focus from the more broad *object* tracking to the more specific domain of *pedestrian* tracking. Specifically, we are interested in scenes which contain large numbers of pedestrians that can be seen to navigate their surroundings including other pedestrians. The crowds and groups of people navigating their way through the scene will test the abilities of the tracker.

We will show in this chapter that standard tracking methods are significantly improved when intelligent motion models are utilized using both quantitative measurements as well as by qualitatively analyzing failure cases. When good quality video is available, we show a 26% reduction in tracking error when comparing a multi-behavior tracking prior with the industry standard Kalman tracker. When image quality is less than ideal, we show that the multi-behavior tracker is robust to error in the avoidance locations. In the presence of partial occlusions the socially influenced motion prior is able to track certain pedestrians while the standard approach fails completely. By integrating our published pedestrian model from the

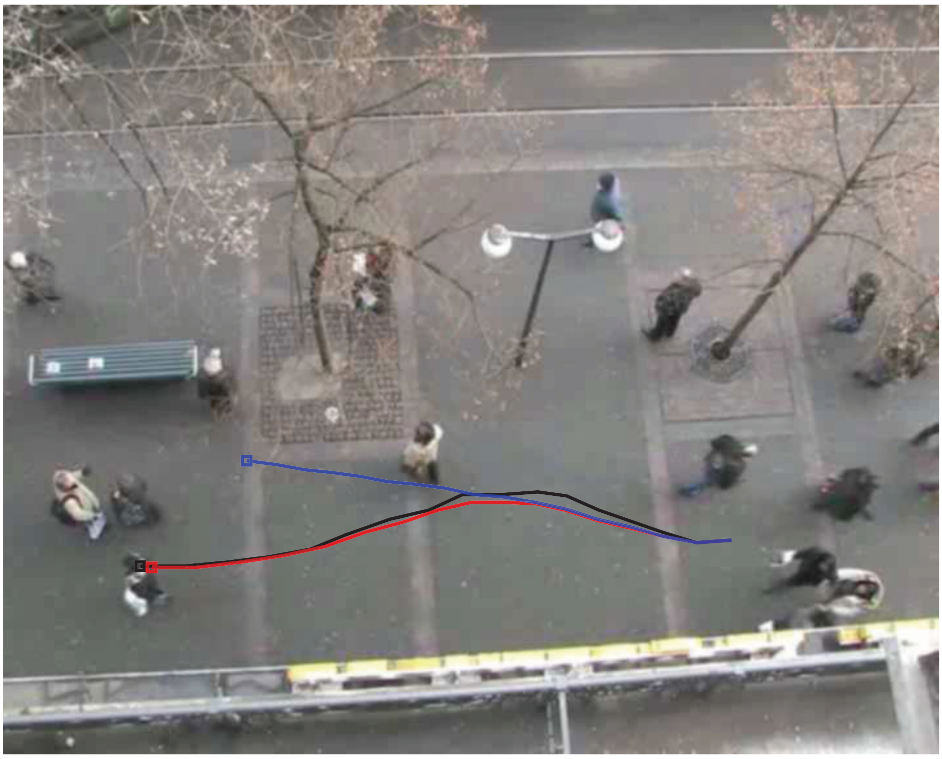


Figure 4.1: Tracking results on the LTA dataset. Black represents the ground-truth pedestrian track. Blue represents the Kalman tracker. Red represents our SPM tracker. The tracker pedestrian deviates from his intended path to avoid the pedestrian in white in the center of the scene.

previous chapters with appearance based tracking, this chapter will highlight the application of our pedestrian model to the active subfield of pedestrian tracking and complete the body of research that is this dissertation.

4.2 Method

This section will discuss the method used to track individuals in a scene. Simply put, tracking is the task of associating the objects observed at one time with another. We assume that there exists some method of detecting pedestrians as they enter a scene. This is done in practice commonly through background subtraction or object detection. Once a pedestrian is detected, we build a straightforward appearance model using Normalized Cross-Correlation. The appearance model and a motion prior, that uses the predicted value of a motion estimation method, are used to predict the most likely current position given the previous observations. This process is repeated until a pedestrian leaves the field of view of the camera.

Object tracking is a well studied area of computer vision, and more advanced object trackers that the one used in this chapter exist in literature. A fair criticism would be that these tracking methods would possibly produce better tracks than the straight-forward approach taken in this chapter. The important point to note is that the vast majority of these tracking algorithms still make very naive assumptions about the motions of the objects. Particle filter based tracking in the vast majority of works either assumes a Brownian or linear motion model. Particle filtering is merely a method for estimating the shape of a distribution; when applied to the task of object tracking, particle filtering is agnostic to the motion model. The lack of intelligent motion models in cutting edge tracking algorithms, as well as the fact that most tracking algorithms treat the motion model as a black box, gives a strong motivation for this work.

4.2.1 Initialization

Agent initialization is the first step in any tracking algorithm. Some methods for tracking will use background models, or appearance based classifiers to initialize pedestrians [SG99] [JS02]. Other methods rely on continual re-detection of tracked objects, such as methods with moving cameras or other difficult situations [ARS08a] [LSV07]. Our method is limited to a single static camera, although there is no reason image registration couldn't handle small camera movements, and multiple calibrated cameras could be used to generate a more accurate appearance model. The experiments in this dissertation assume that some sort of detection and agent initialization method does exist which can provide accurate initial positions and initial velocities.

The LTA dataset used in the experiments contains a single point to annotate the positions of pedestrians. The ground-truth location for each pedestrian is located on the top of each pedestrian's head. We used a bounding box immediately below the given head position of a fixed size. The size and shape of pedestrians can vary, and background subtraction techniques can improve this naive approach, however background subtraction techniques can also introduce errors by incorrectly initializing pedestrians to the wrong dimensions. Due to the template nature of Normalized Cross-Correlation which will be discussed in

Section 4.2.2, once a template is created for a pedestrian, changes in template window size are not possible. We determined a size of 40 by 40 pixels was appropriate after manually estimating the heights and widths of the pedestrians in the training set. The variation in size was small due to the size of pedestrians and the angle of the camera. The 40 by 40 window encompassed the pedestrians as well as a small amount of background; that proved, in initial testing, to be better than a "too small" window which may cut off parts of pedestrians.

4.2.2 Appearance Model

In order to accurately track pedestrians, we will create a probabilistic model to combine inputs from two independent sources of information. The first is a standard 2D Normalized Cross-Correlation (NCC) method for measuring image similarity. The NCC method is useful for tracking because it is robust to illumination changes, however it is less robust to rotation or pose changes than histogram based appearance models or probability density models. However, pedestrians do not rotate in the test dataset, nor do they change pose significantly. The advantage of the NCC method is the encoding of both shape information and appearance information, which are the most important aspects in the given dataset. Additionally, the NCC method does not require any training, other than the initialization image patch. The NCC method is easily converted to a probability map for the entire scene, making it a fast and efficient appearance based tracking method. The appearance based probability map will be multiplied by the motion based probability map, this results in a weighted probability map of the scene. Pedestrians are assumed to be located at the location with the

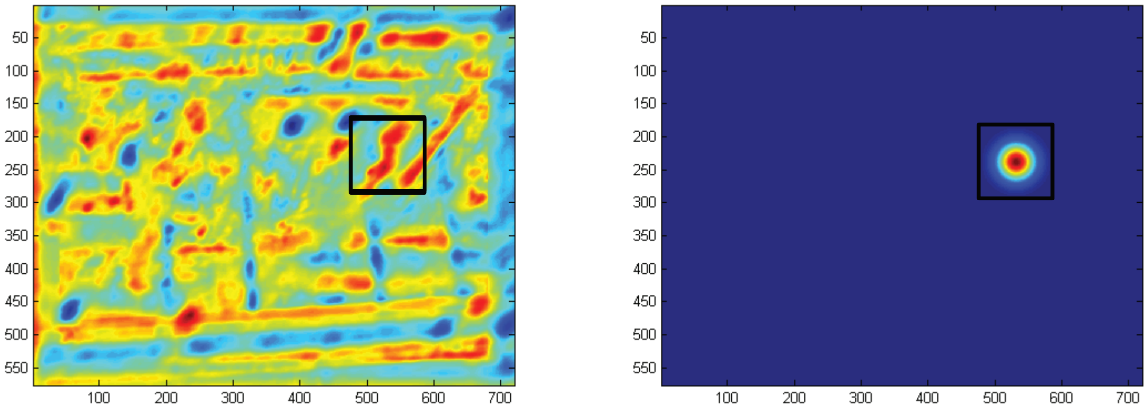


Figure 4.2: The results of (a) Equation 4.1 and (b) Equation 4.2 on a sample frame of the dataset. The black box is shown to specify the location of the pedestrian, detail inside the black box is shown in Figure 4.3.

highest probability. This approach for combining appearance and motion information into a straightforward tracking algorithm has been used in previous social force tracking methods [PES09] [KAO11].

The appearance based NCC map is computed as follows:

$$NCC(x,y) = \frac{\sum_{x,y}\left[(f(x,y)-\bar{f}_{u,v})(t(x-u,y-v)-\bar{t})\right]}{\sqrt{\sum_{x,y}\left[f(x,y)-\bar{f}_{u,v}\right]^2\sum_{x,y}\left[t(x-u,y-v)-\bar{t}\right]^2}} \tag{4.1}$$

where t represents the template image, $\bar{t}$ represents the mean of the template, and f represents the image patch which is being tracked. $\bar{f}_{u,v}$ is the mean of $f(x,y)$ in the region under the template. The result of NCC is a value measuring the template's similarity for each location in the scene.

In addition to appearance information, the motion information will be provided by the multi-behavior social force model that will be used to estimate the location based on the current track. This estimated position will be converted to a probability using a Gaussian distribution centered at the predicted location. The σ value, specified as σ_k will determine the size of the region where the tracked pedestrian is estimated to be. This σ_k parameter will be the only parameter which controls the tracking algorithm. A small value for σ_k corresponds to a very tight Gaussian and most of the tracking will be done in this case by the motion model. A large value corresponds to a very large Gaussian probability distribution, where many positions in the scene share similar motion estimation likelihoods. In Section 4.3.

We define the motion prior by the following Gaussian probability function:

$$P_{motion}(x,y) = \frac{e^{\left(-[(u-x)^2+(v-y)^2]/[2*\sigma_k^2]\right)}}{\sum_{u,v}\left[e^{\left([(u-x)^2+(v-y)^2]/[2*\sigma_k^2]\right)}\right]} \tag{4.2}$$

where the window (u,v) is large enough to properly contain all sufficiently non-zero values.

Equations 4.1 and 4.2 are combined using the following function.

$$P_{track}(x,y) = (NCC(x,y)+1) * P_{motion}(x,y) \tag{4.3}$$

In the original prediction evaluation methods it was assumed that the position of the other pedestrians in the scene was known, and could be accurately tracked during prediction. For the application of tracking, this assumption is incorrect. When tracking, the positions of each object being tracked is only as good as the tracker. This means that non-stationary obstacles (other pedestrians) must be tracked in parallel. To that end, we modified the algorithm such that we no longer tracked each pedestrian one-by-one. Instead, at each discrete time step pedestrians would be tracked into the next time step. This is how a real-time tracking algorithm must be written. The tracking method is described algorithmically as follows:

```
for $t = 1 \ldots T$ do
    for $\mathbf{p} \in \mathbf{P}$ do
        Generate $\mathbf{x}_{t+1}$ using the algorithm from Section 3.5;
        Compute $P_{motion}(\mathbf{x}_{t+1})$ and $NCC(x, y)$;
        Compute the tracked location $\check{\mathbf{x}}_{t+1} = \arg\min_{x,y} P_{track}(x, y)$;
        Update $\mathbf{x}_{t+1}$, replacing it by the value $\check{\mathbf{x}}_{t+1}$ in all stored locations.
    end
end
```

One caveat of this tracking algorithm is that over time, as tracks drift, the scene information used to compute the pedestrian avoidance energies may become less reliable. Errors in tracking could cause incorrect motion predictions for the pedestrians being avoided, resulting in even worse tracking. This feedback loop could potentially cause the SPM method to perform worse than the Kalman filter as a motion prior, despite the fact that we have already shown the SPM method significantly outperforms the Kalman filter at the task of pedestrian motion prediction. Since the Kalman filter prior does not depend on the locations of the other pedestrians in a scene, this feedback loop would not be possible. However, this

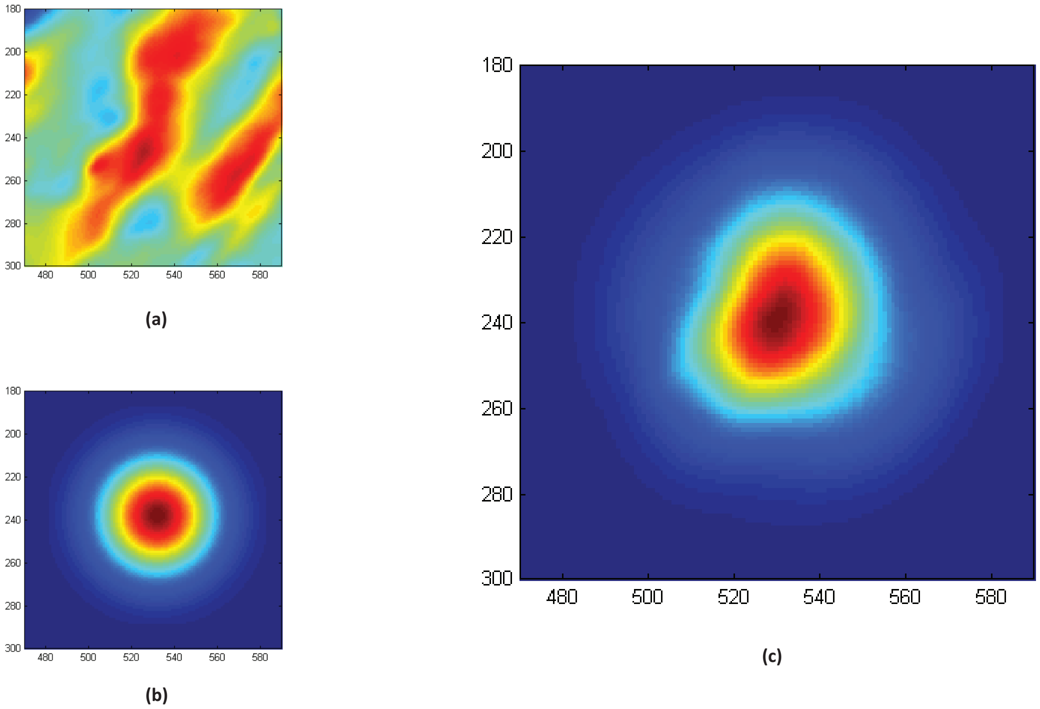


Figure 4.3: Detail from Figure 4.2. (a) The NCC appearance based prediction. (b) The motion estimated prior. (c) The combination of (a) and (b) which is computed by Equation 4.3.

phenomenon was not apparent in our tests. While the Kalman did outperform on some pedestrians, over the entirety of the test set the SPM prior significantly outperformed the Kalman prior as will be shown in Section 4.3.

4.2.3 Kalman Filtering

In the classical Kalman filtering application, the input to the algorithm are a set of observations as well as a covariance matrix. In our application, the observations are the locations of the pedestrians (x, y), and the covariance matrix is the identity matrix. The states modeled by the Kalman filter are the position, x and y, and the velocity, v_x and v_y. As each location is predicted, the system would be updated with the actual observation and the process would repeat itself. In a tracking application, the observations beyond the initialization stage are not always correct. The observations beyond the initialization frames are actually the locations predicted by the tracking algorithm. This is slightly different from the classical Kalman filtering definition which assumes that at the next discrete time step true observations are available at the previous time steps. However, this limitation is required because the information necessary for the classical definition is not available to an object tracking algorithm, true observations are merely the results of the tracking algorithm.

The actual implementation of the Kalman filter used in this dissertation can be found online [1].

4.2.4 Motion Prior Probability Distributions

This section will discuss alternative motion prior distributions, and discuss why the Gaussian weighting probability was used.

One could imagine a potential motion prior where the energy values are converted directly into probabilities. This would remove the need for minimizing the energy function in the SPM method and seemingly save processing time. This is false, since the minimization of the energy function is quite efficient due to the direct upper-bound minimization process.

[1] http://www.cs.ubc.ca/ murphyk/Software/Kalman/kalman.html

Sampling the energy field at the resolution necessary for prediction is far more computationally expensive. Additional runtime and complexity analysis is provided in Section 4.4.1. Another reason for using a Gaussian located at the point estimated by the motion model is that a single estimated location is what the model was trained to produce, not any specifically shaped energy function. While it is possible that using the energy function to compute location probabilities would result in accurate tracking, the model would not be trained to fulfill this task. In addition to the above two reasons, by using a single estimated location and a Gaussian probability, we can directly compare to the industry standard: constant velocity Kalman filter estimation.

4.3 Experimental Results

For the experiments in this chapter, we will compare a standard motion prior, the Kalman filter, against our pedestrian modeling method introduced in the previous chapter using the framework described in the previous section. Once again, the model is trained using the first third of the LTA dataset [PES09], and evaluated on the second two-thirds of the data. The tracking algorithm will be evaluated using the publicly available LTA dataset which was used in the previous chapter as well. This section will show how the error in tracking accuracy is reduced by 26% when multi-behavior social forces are used to model pedestrian motion.

The Kalman filter is a standard approach, and while it is incredibly useful, we will show in this chapter that it is insufficient to describe and eventually track the complex interactions of pedestrians. In near collisions amongst pedestrians, the linear prediction leads to losing the target, or in some cases, the switching from one pedestrian to another. These cases will be discussed later in this section as well as in Section 4.4. The details of our Kalman filter formulation and implementation can be found in Section 4.2.3.

The scene labeled "seq_hotel" from the LTA dataset [PES09] presents a particularly good test case for tracking and the use of socially aware motion priors. The scene, which covers a busy sidewalk, is partially occluded by trees. Since the data was recorded in the winter, these trees have lost their leaves, and pedestrians can be seen through them. However the partial occlusion which is present in this scene adds extra difficulty towards the task of tracking. Sample tracks from this portion of the data can be seen in Figure 4.1, 4.4, and 4.6.

4.3.1 Quantitative Comparison

In order to quantitatively compare the two motion priors we allowed both trackers to independently track the entire test dataset. We then measured the overall loss between the tracked paths and the ground-truth using the following loss function:

$$L(\check{\mathbf{x}}, \mathbf{T}) = \sum_{i=1}^{N_s} ||\mathbf{x}_i - \mathbf{T}_i|| \qquad (4.4)$$

where $\check{\mathbf{x}}$ and $\mathbf{T}$ are the tracked and ground-truth pedestrian paths and N_s is the number of samples in each path.

Motion Prior Weight					
6	12	18	24	36	48
217	156	137	136	153	181

Table 4.1: SPM tracker cumulative error under various operating conditions. The tracking algorithm was tested using increasingly larger values for σ_k, seen in the middle row, until we were satisfied that further testing would not result in significantly better results.

Motion Prior Weight					
6	12	18	24	36	48
368	224	199	199	243	305

Table 4.2: Kalman tracker cumulative error under various operating conditions. The tracking algorithm was tested using increasingly larger values for σ_k, seen in the middle row, until we were satisfied that further testing would not result in significantly better results.

The above function sums the L2 distance between the tracked position and the ground-truth at every point in time. Therefore, a track which lags behind the ground-truth but eventually catches up will have a positive loss, even if it follows the same path as the ground-truth. This is different than if the minimum distance were taken between each track ignoring the time component, more common in methods suited for handwriting analysis. Practically, this distinction is not very important for our application since trackers that lose their target are not likely to "catch up."

The results of this evaluation can be seen in Tables 4.1 and 4.2 where the total error is accumulated over the test set according to Equation 4.4. By testing the algorithm using a line search of possible σ_k values, we have attempted to find the best value for this dataset.

If σ_k is too large, then the motion information will be ignored; conversely if σ_k is too small then the appearance information will be ignored. The value which corresponds to be optimal tracking results will depend on the dataset. The data in Table 4.1 and Table 4.2 indicate that for the LTA dataset, the optimal σ_k is between 18 and 24.

4.4 Image Degradation

Image quality has a significant effect on appearance based models for obvious reasons. There are many common natural causes of poor image quality; for example: weather conditions, incorrect/malfunctioning sensor information, image compression, and transmission noise. The LTA dataset does not contain many such problems which are commonly encountered in real world applications. Therefore, we chose to test our trackers in the presence of various levels of Gaussian blur. By blurring the input image we are essentially decreasing the sensor resolution without negatively affecting the annotation used to initialize our tracks. We will refer to the σ parameter used to govern the image blur as σ_i to prevent confusion with the motion prior weight σ_k.

The results of this evaluation can be seen in Tables 4.3 and 4.4, which are expanded versions of Tables 4.1 and 4.2. Additionally, these tables are visualized for easier understanding in Figure 4.5. While the 3D surface can be used to see relationships that are not immediately apparent when viewing the table, it can be difficult to read very precisely. Therefore, we have also included Figure 4.7 which compares the Kalman prior to our SPM prior directly under an array of σ_k values at a fixed image blur. Each subgraph in Figure 4.7 corresponds to a fixed amount of image degradation, which can be viewed in Figure 4.8.

		Motion Prior Weight					
		6	12	18	24	36	48
Image Blur	0	217	156	137	136	153	181
	1	219	159	138	**135**	144	160
	3	240	175	153	144	149	168
	6	262	207	179	166	176	198
	10	269	231	207	197	202	227
	21	272	265	241	243	261	301

Table 4.3: SPM tracker cumulative error under various operating conditions, the error from the best tracker configuration for this motion prior is in **bold**. Horizontally, the tracking algorithm was tested using increasingly larger values for σ_k (motion prior weighting value) until we were satisfied that further testing would not result in significantly better results. Vertically, the tracker was tested under decreasing image quality due to increased image blur, samples of these blurred images can be seen in Figure 4.8.

In all test cases the SPM motion prior outperformed the Kalman prior. This does not mean that the SPM motion prior outperformed on every track; for some pedestrians, the Kalman filter did track much better. However, over the entire test set, the SPM motion prior proved the better option. Often the difference was quite significant. Under the large blur of $\sigma_i = 21$, the SPM motion prior measured on average 50% less error than the equivalent Kalman prior. An example of this difference can be seen in Figure 4.4 where the same pedestrian was tracked under good and very poor image quality. In full resolution both

		Motion Prior Weight					
		6	12	18	24	36	48
Image Blur	0	368	224	199	199	243	305
	1	370	215	**193**	210	223	264
	3	422	271	213	223	230	253
	6	494	339	283	269	301	341
	10	507	421	357	341	355	416
	21	517	497	479	502	546	641

Table 4.4: Kalman tracker cumulative error using the same settings as Table 4.3. See the caption of Table 4.3 above for more details.

trackers produce very similar, and mostly correct results. However, when the same pedestrian is tracked under poor image conditions, both tracks begin to drift, but the SPM prior does not allow the track to collide with another pedestrian and reacquires the original target.

The closest operating conditions in our tests occurred at $\sigma_i = 1$ and $\sigma_k = 12$, where the SPM method produced only 26% less error than the Kalman tracker. This is still a very significant reduction of error. This is due to the fact that the SPM prior is far better suited to predicting pedestrian movements, and the LTA dataset was originally created to observe human pedestrian behavior.

Small amounts of image blur can be beneficial to NCC, or other template based trackers. By slightly blurring the input image, the appearance model gains some robustness to minor pose and shape changes without significantly diminishing the color information. As can be seen in Tables 4.3 and 4.4, the best test results occurred under a $\sigma_i = 1$ Gaussian blur.

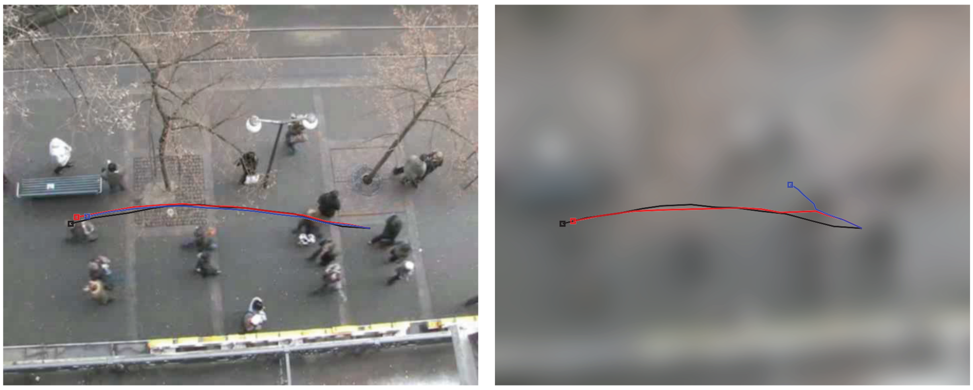


Figure 4.4: The same pedestrian tracked under very different image conditions. The left image shows that both motion estimation models are able to accurately predict this individual on a crowded sidewalk. The right image shows that even under significant image degradation the SPM prior continues to track the pedestrian where the Kalman prior fails.

We have shown that poor image quality is a contributing factor towards poor tracking performance, especially when using linear motion assumptions. Figure 4.6 shows an example where partial occlusion leads to poor tracking results. In this scene a pedestrian is seen moving underneath a tree which has lost its leaves. The fact that so much of a pedestrian is occluded when underneath this tree causes poor appearance information. The top row in Figure 4.6 shows the scene without artificial image degradation, and the bottom row shows the scene under a significant amount of Gaussian blur. Additionally, the left, middle and right columns show small, moderate and large values values for the σ_k weight. By looking at the left-most column where the motion prior is most restrictive, we can see that the artificial image degradation has little effect on the overall tracking in this case. Using an appropriate σ_k value, the middle column shows decent tracking under both quality conditions, while the

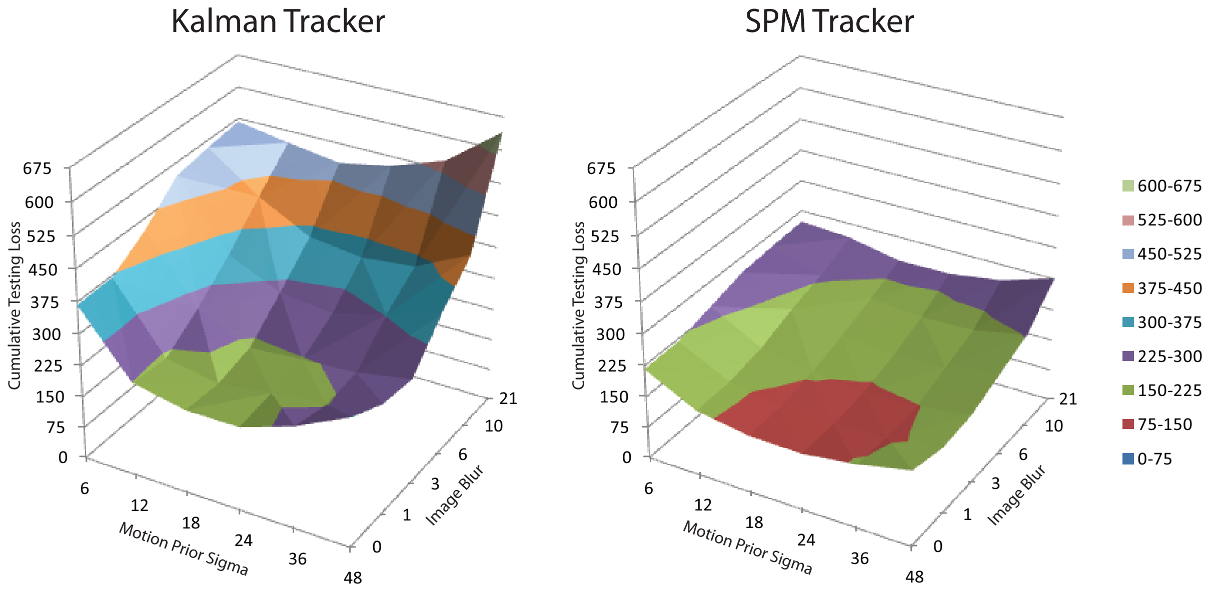


Figure 4.5: Tracking error for Kalman (left) and SPM (right) motion priors. The z-axis represents the overall testing error, the x-axis represents the weight of the motion prior, and the y-axis represents the degradation of the image quality for the appearance based tracker. The SPM motion prior outperforms the Kalman filter in all test settings. The optimal settings are a moderate sized σ_k and a small amount of image blur σ_i. Too large or too small of a motion prior Gaussian results in poor tracking, as well as significant amounts of image degradation.

Kalman tracker continues to struggle. Finally, in the right-most column the Kalman tracker can be seen to completely fail. In fact, both the left and right columns show complete failures in the Kalman tracker; in the left column the small σ_k value causes the failure case to at least travel along in the last known direction, in the right column the σ_k value is large and thus does not influence the tracker enough to move from the initial location.

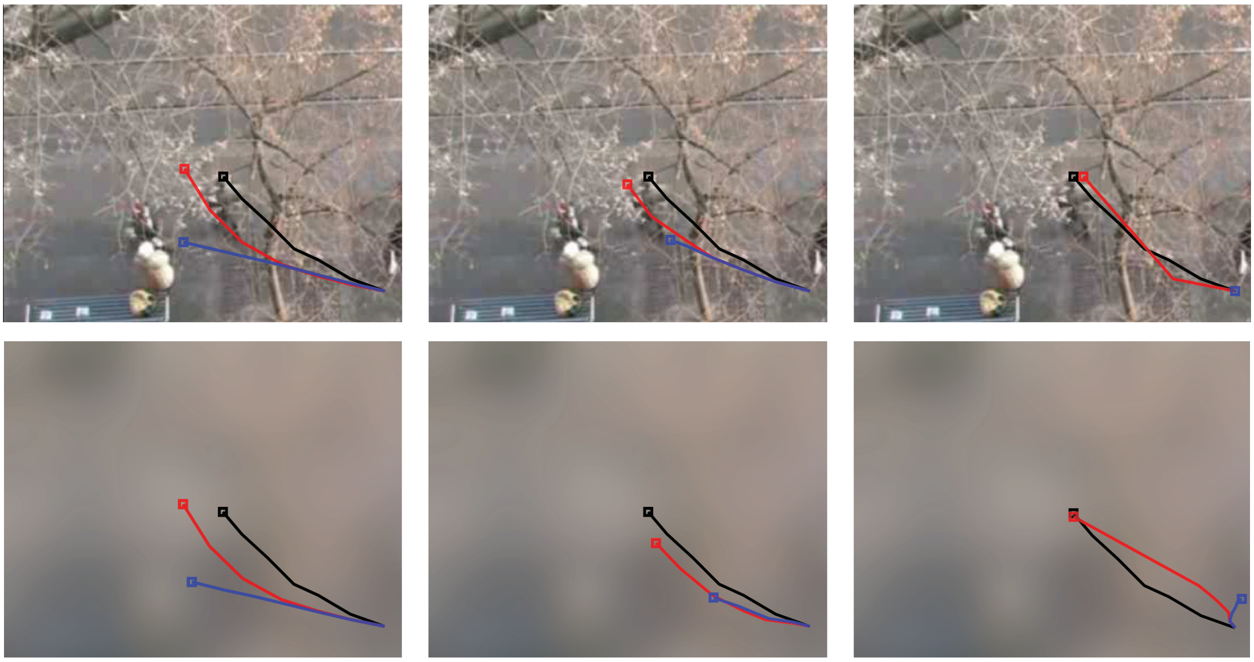


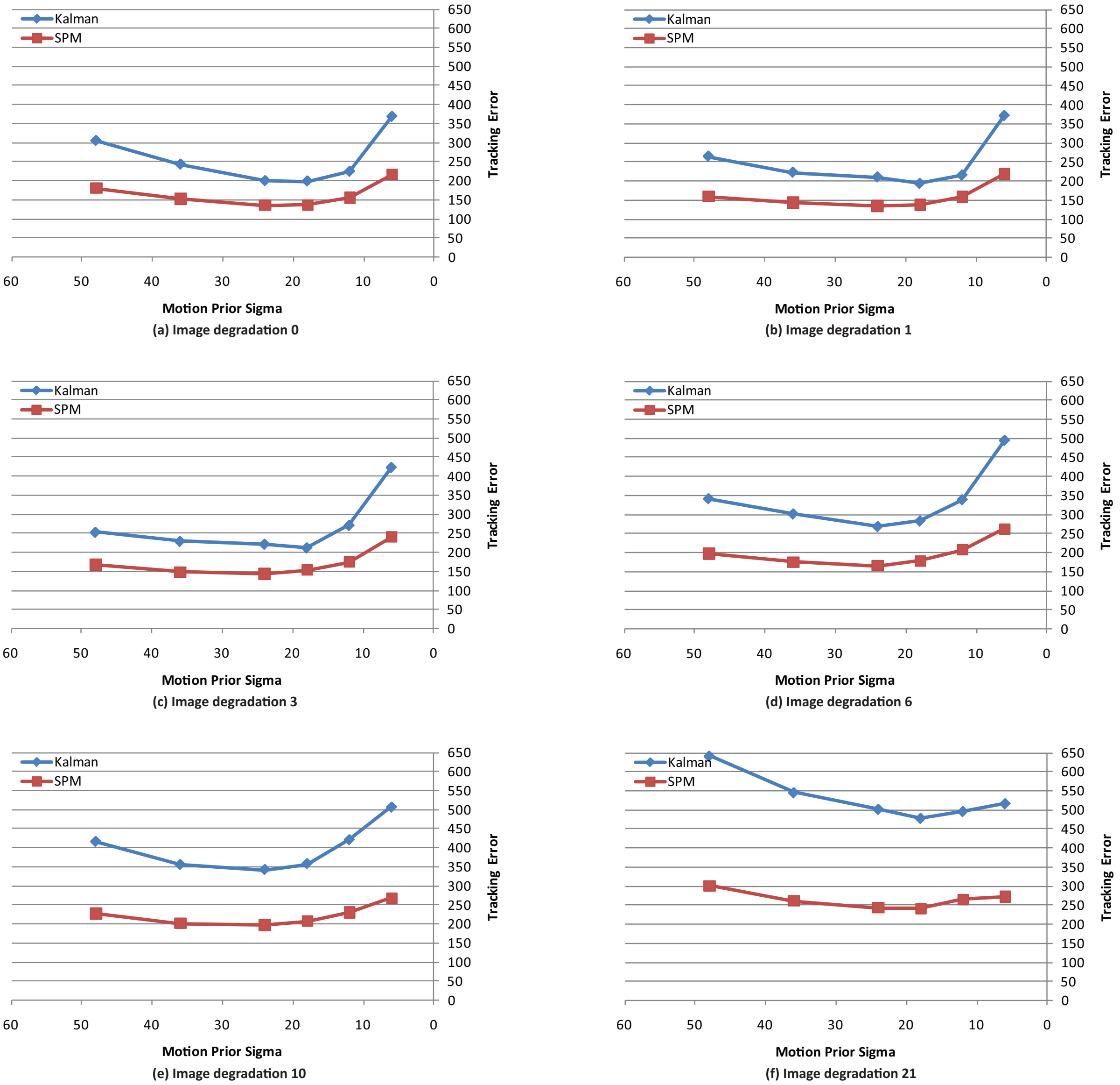
Figure 4.6: Tracking results under partial occlusion from the tree. Ground-truth labeled in black. Top-Left: Small values for the prior sigma result in paths which deviate little from the motion prior's path. Top-Middle: Using the most optimal prior weight, SPM is able to keep track of the pedestrian, however the Kalman prior continues in the wrong direction. Top-Right: Large prior sigmas result in complete failure from the Kalman tracker, however the SPM tracker is able to maintain the pedestrian. As seen from the bottom row, image blur does little to effect the smallest sigma tracker; other values for sigma do result in different tracks, however qualitatively they are quite similar.

4.4.1 Runtime and Complexity Analysis

Due to the nature of tracking algorithms, real time performance is often required. This section will discuss this goal and the challenges faced by SPM which is currently written in MATLAB in a parallelized program.

Because each pedestrian in a scene interacts with the the other pedestrians, we can see that the complexity of the algorithm grows with the square of the number of pedestrians. MATLAB, which stands for Matrix Laboratory, is different than C/C++ since it is far more efficient to compute fixed size matrix operations than computing the same value using for loops. This limitation caused an implementation restriction to be put on the maximum number of pedestrians in a scene. We analyzed the dataset and determined that no more than 12 annotated pedestrians were ever seen at the same time in the scene, therefore we fixed the size of the matrix to this value. This essentially forces the algorithm to always run under worst-case conditions, however it is actually more efficient than the alternative. If larger scenes were used, then this number would need to be increased or some pedestrian interactions would have to be ignored. Practically speaking, pedestrians will ignore the large majority of other pedestrians in a large scene, so this limitation may not cause a significant drop in performance. If the scene were too large pedestrians who were behind an individual, or too far from them could be ignored rather than the pedestrians nearby and in front of an individual.

The current parallelized implementation running on a Quad-core i7 desktop computer can predict an entire average length track in 1.02 seconds. Given that the average track is 20.77 frames in length, and the first 5 points are ignored/used for initialization, this means on average pedestrian can be predicted at a rate of 15.46 frames per second. This is roughly half the speed of what is considered realtime, however it is not an unreasonable amount of latency depending on the application.



(a) Image degradation 0

(b) Image degradation 1

(c) Image degradation 3

(d) Image degradation 6

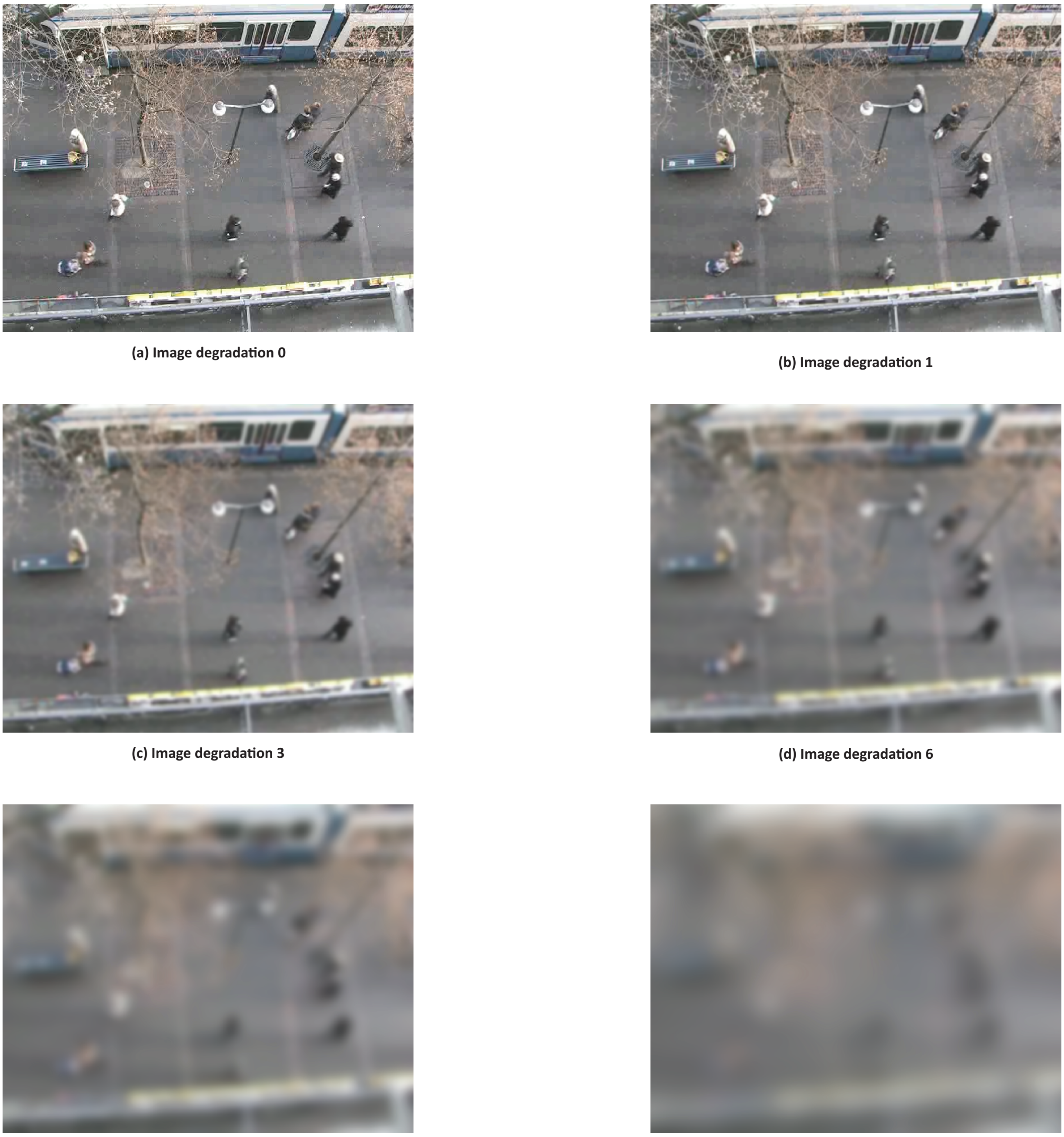
(e) Image degradation 10

(f) Image degradation 21

Figure 4.7: Tracking error for Kalman (blue) and SPM (red) motion priors. The y-axis represents the total tracking error accumulated over the testing set. The x-axis represents the motion prior sigma value; a small prior means that the tracker will obey the motion information more than a large prior value which will allow the motion information to be ignored. The exact function can be found in Equation 4.3. Each graph represents a different amount of Gaussian image blur which was applied to challenge the tracking method (See Figure 4.8 for details). At high values of degradation ((e) and (f)), the differences between the motion priors are even more pronounced since the appearance information is less reliable.

4.5 Summary

This chapter has explored the ability of our SPM model to be used as a motion prior for pedestrian tracking. Tracking algorithms for pedestrians are integral to many practical applications such as: vehicle early warning systems, crowd stability analysis, surveillance and security. The results have shown that our novel SPM method can significantly improve pedestrian tracking when compared to the industry standard approach. We have attempted to provide as much data for analysis as possible.



(a) Image degradation 0

(b) Image degradation 1

(c) Image degradation 3

(d) Image degradation 6

(e) Image degradation 10

(f) Image degradation 21

Figure 4.8: Image degradation examples. Subfigures correspond to Figure 4.7.

CHAPTER 5: CONCLUSION

This work has detailed the original research on the subject of pedestrian tracking and the efficient training of models for predicting human movements. The pedestrian model described in this dissertation has been shown to be novel and effective for the purposes of simulation, prediction, and tracking. This work began as a simplistic energy model which was shown to be trained efficiently using Variational Mode Learning. A novel extension was made which allowed multiple motion models to be trained in parallel. This new method was shown to significantly improve the predictive ability of the original pedestrian model. This work has shown that the advantages of predictive ability can be carried on to the task of pedestrian tracking in real world scenes, resulting in significant quantitative advantages over linear motion assumptions.

Some of the specific contributions of the work described in this dissertation are outlined below.

5.1 Summary of Contributions

Chapter 3 introduced the energy model for our LPD method that predicts the social interactions of pedestrians. In Section 3.6 we show how a set of parameters can be learned using real pedestrian tracks. This trained model is able to outperform the industry standard for motion prediction, Kalman filtering. Section 3.9.1 shows that the LPD method can be trained on automatic pedestrian tracks generated by a object tracking algorithm. Furthermore, it shows that the LPD method offers greater improvement the longer a prediction is made, up to 35% reduction in error, when compared to straight line prediction. Using the evaluation metrics used by another social force motion prediction method, LTA, we show in Section 3.8.2 that we are are able to learn similarly descriptive models.

The LPD model is extended to handle multiple pedestrian behaviors in Section 3.4. Other works have begun to handle the formation of social groups by incorporating additional parameters, however we show that this is insufficient. Table 3.1 shows that the addition of group behaviors, at a correct prediction group assignment rate of 98%, as well as the addition of local neighborhood influence does improve prediction by 5.22%. Stereotyping pedestrians based on their movement patterns involves training multiple sets of parameters. By stereotyping pedestrian behaviors, we are able to estimate pedestrian motion with less than half the error of a more naive single behavior social force model.

Chapter 4 discusses the work related to applying the pedestrian motion model to the common computer vision task of object tracking. This chapter looks into tracking in various circumstances. It shows that using ideal image quality, a multi-behavior social force motion information can offer a 26.25% reduction in tracking error when compared to Kalman filtering motion priors. Under significant image quality reduction this difference is amplified, despite the fact that the locations of the pedestrians in the crowd are less reliable. This robustness is important to proving how pedestrian motion models can be useful in real world applications. A common requirement of real world tracking applications is runtime efficiency and scalability. While scalability is an issue due to the fact that the number of interactions between pedestrians grows with the square of the number of pedestrians, the current parallelized MATLAB implementation is able to predict pedestrian tracks in just over 1 second on average on a Quad-core i7 machine.

5.2 Future Directions

The framework described in this dissertation has been developed to explore the area of pedestrian motion models and the obvious applications. While these methods attempt to be as complete as possible, there are additional avenues of research yet to be explored. One possibility is the addition of scene characteristics. Obstacles are annotated in a scene's ground truth; however obstacles should truly be defined by the individuals in the scene. It would be possible that certain behavior stereotypes may walk through regions of a scene that others would avoid, such as the grassy lawns that separate sidewalks on campuses or in parks. Another possibility would be the use of attractors, these points would be a sort of intermediary destination which is shared among multiple pedestrians, such as water-coolers or scenic locations along a walk; the opposite of the obstacle, these points are locations that people like to be.

Additionally, human pedestrian behavior is not the only motion which is poorly described by linear assumptions. From the very small (eg: movement and behavior of microscopic organisms) to the very large (eg: large container ships on the ocean use early warning collision prediction systems), systems of behavior which seem complex can be modeled whenever the underlying motivations can be defined. As such, the models of this dissertation could be directly applied to an array of new applications.

LIST OF REFERENCES

[AMB06] G. Antonini, S. Martinez, M. Bierlaire, and J. Thiran. “Behavioral Priors for Detection and Tracking of Pedestrians in Video Sequences.” *IJCV*, 2006.

[ARS08a] M. Andriluka, S. Roth, and B. Schiele. “People-Tracking-by-Detection and People-Detection-by-Tracking.” *CVPR*, 2008.

[ARS08b] M. Andriluka, S. Roth, and B. Schiele. “People-tracking-by-detection and people-detection-by-tracking.” In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1 –8, june 2008.

[AS07] S. Ali and M. Shah. “A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis.” *CVPR*, 2007.

[AS08] S. Ali and M. Shah. “Floor Fields for Tracking in High Density Crowd Scenes.” *ECCV*, 2008.

[ATC05] D Anguelov, B Taskar, V Chatalbashev, D Koller, D Gupta, G Heitz, and A Ng. “Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR05)*, San Diego, CA, June 2005.

[ATH03] Yasemin Altun, Ioannis Tsochantaridis, and Thomas Hofmann. “Hidden Markov Support Vector Machines.” In *ICML*, 2003.

[BB88] H.A.P. Blom and Y. Bar-Shalom. “The interacting multiple model algorithm for systems with Markovian switching coefficients.” *Automatic Control, IEEE Transactions on*, **33**(8):780 –783, aug 1988.

[BC86] T. J. Broida and R. Chellappa. “Estimation of Object Motion Parameters from Noisy Images.” *PAMI*, 1986.

[BC06] Gabriel J. Brostow and Roberto Cipolla. “Unsupervised Bayesian Detection of Independent Motion in Crowds.” In *IEEE Computer Vision and Pattern Recognition*, pp. I: 594–601, 2006.

[Ben00] Yoshua Bengio. “Gradient-Based Optimization of Hyperparameters.” *Neural Comput.*, **12**(8):1889–1900, 2000.

[Bes75] Julian Besag. “Statistical Analysis of Non-Lattice Data.” *The Statistician*, **24**(3):179–195, Sept 1975.

[BG05] M. Blank and L. Gorelick. “Actions as Space-Time Shapes.” 2005. ICCV.

[Bis95] Christopher M Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[Bis07] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer, 1 edition, 2007.

[BK99] David Beymer and Kurt Konolige. “Real-Time Tracking of Multiple People Using Continuous Detection.” 1999.

[BNJ03] D. Blei, A. Ng, and M. Jordan. “Latent Dirichlet Allocation.” 2003. Journal of Machine Learning Research.

[BR96] Michael J Black and A Rangarajan. “On the unification of line processes, outlier rejection, and robust statistics with applications in early vision.” *International Journal of Computer Vision*, **19**(1):57–92, July 1996.

[BRB04] Andrew Blake, Carsten Rother, Matthew Brown, Patrick Perez, and Philip Torr. “Interactive Image Segmentation using an adaptive GMMRF model.” In *European Conference on Computer Vision (ECCV)*, 2004.

[BRL09] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. “Robust tracking-by-detection using a detector confidence particle filter.” In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1515 –1522, 29 2009-oct. 2 2009.

[BSU04] Eran Borenstein, Eitan Sharon, and Shimon Ullman. “Combining Top-Down and Bottom-Up Segmentation.” In *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 4*, p. 46, Washington, DC, USA, 2004. IEEE Computer Society.

[BU04] E Borenstein and S Ullman. “Learning to Segment.” In *European Conference on Computer Vision (ECCV)*, May 2004.

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. “Fast Approximate Energy Minimization via Graph Cuts.” *IEEE Transactions of Pattern Analysis and Machine Intelligence*, **23**(11):1222–1239, 2001.

[BYB09] B. Babenko, Ming-Hsuan Yang, and S. Belongie. “Visual tracking with online Multiple Instance Learning.” In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 983 –990, june 2009.

[BZ87] Andrew Blake and Andrew Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, Massachusetts, 1987.

[CC07] Nicolas Courty and Thomas Corpetti. "Data-driven animation of crowds." In *Proceedings of the 3rd international conference on Computer vision/computer graphics collaboration techniques*, MIRAGE'07, pp. 377–388, Berlin, Heidelberg, 2007. Springer-Verlag.

[CH05] M E Carreira-Perpignan and G E Hinton. "On Contrastive Divergence Learning." In *Artificial Intelligence and Statistics (AISTATS)*, Barbados, 2005.

[CL01] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Col02] Michael Collins. "Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms." In *EMNLP 2002*, 2002.

[CRM00] D. Comaniciu, V. Ramesh, and P. Meer. "Real-Time Tracking of Non-Rigid Objects using Mean Shift." *CVPR*, 2000.

[CS10] Wongun Choi and Silvio Savarese. "Multiple target tracking in world coordinate with single, minimally calibrated camera." In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pp. 553–567, Berlin, Heidelberg, 2010. Springer-Verlag.

[CVB02] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. "Choosing Multiple Parameters for Support Vector Machines." *Mach. Learn.*, **46**(1-3):131–159, 2002.

[DFN08] Chuong Do, Chuan-Sheng Foo, and Andrew Ng. "Efficient multiple hyperparameter learning for log-linear models." In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pp. 377–384. MIT Press, Cambridge, MA, 2008.

[DH09] Hannah M. Dee and David C. Hogg. "Navigational strategies in behaviour modelling." *Artificial Intelligence*, **173**(2):329 – 342, 2009.

[DRC05] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. "Behavior Recognition via Sparse Spatio-Temporal Features." 2005. ICCCN.

[DT05] Navneet Dalal and Bill Triggs. "Histograms of Oriented Gradients for Human Detection." *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, **1**:886–893, 2005.

[EBM03] A. Efros, A. Berg, G. Mori, and J. Malik. "Recognizing Action at a Distance." 2003. ICCV.

[EG08] M. Enzweiler and D. M. Gavrila. "Monocular Pedestrian Detection: Survey and Experiments." *PAMI*, 2008.

[EG09] Markus Enzweiler and Dariu M. Gavrila. "Monocular Pedestrian Detection: Survey and Experiments." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**:2179–2195, 2009.

[ELS09] A. Ess, B. Leibe, K. Schindler, and L. van Gool. "Robust Multiperson Tracking from a Mobile Platform." *PAMI*, 2009.

[FFP03] L. Fei-Fei, R. Fergus, and P. Perona. "A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories." 2003. ICCV.

[FMR08] Pedro Felzenszwalb, David Mcallester, and Deva Ramanan. "A discriminatively trained, multiscale, deformable part model." In *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2008*, 2008.

[FPC00] William T Freeman, Egon C Pasztor, and Owen T Carmichael. "Learning Low-Level Vision." *International Journal of Computer Vision*, **40**(1):25–47, 2000.

[FPZ03] R. Fergus, P. Perona, and A. Zisserman. "Weakly Supervised Scale-Invariant Learning of Models for Visual Recognition." 2003. IJCV.

[FSH06] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W.T. Freeman. "Removing Camera Shake From A Single Photograph." *ACM Transactions on Graphics, SIGGRAPH 2006 Conference Proceedings, Boston, MA*, **25**:787–794, 2006.

[FTA04] R. W. Fleming, A. Torralba, and E. H. Adelson. "Specular reflections and the perception of shape." *Journal of Vision*, **4**(9):798–820, 2004.

[Fura] B Furino. "UCF Engineering Futures Forum." http://istf.ucf.edu/EngForum/.

[Furb] B Furino. "Why Engineering?" http://partner.cecs.ucf.edu/whyeng/.

[FWT03] Brett R. Fajen, William H. Warren, Selim Temizer, and Leslie Pack Kaelbling. "A Dynamical Model of Visually-Guided Steering, Obstacle Avoidance, and Route Selection." *International Journal of Computer Vision*, **54**:13–34, 2003. 10.1023/A:1023701300169.

[GB06] H. Grabner and H. Bischof. "On-line Boosting and Vision." In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pp. 260 – 267, june 2006.

[GCC10] S. J. Guy, J. Chhugani, S. Curtis, P. Dubey, M. Lin, and D. Manocha. "PLEdestrains: A Least-Effort Approach to Crowd Simulation." *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*, 2010.

[GG84] S Geman and D Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images." *IEEE Transactions of Pattern Analysis and Machine Intelligence*, (6):721–741, 1984.

[GR92] D Geman and G Reynolds. "Constrained restoration and the recovery of discontinuities." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**(3):367–383, March 1992.

[GW06] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition).* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[HEH06] Derek Hoiem, Alexei A. Efros, and Martial Hebert. "Putting Objects in Perspective." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, 2006.

[HFV00] D. Helbing, I. Farkas, and T. Vicsek. "Simulating Dynamical Features of Escape Panic." *Nature*, 2000.

[Hin02a] Geoffrey Hinton. "Training products of experts by minimizing contrastive divergence." *Neural Computation*, **14**(7):1771–1800, 2002.

[Hin02b] Geoffrey E. Hinton. "Training products of experts by minimizing contrastive divergence." *Neural Computation*, **14**(8):1771–1800, 2002.

[HM95] D. Helbing and P. Molnár. "Social force model for pedestrian dynamics." *Physical Review E*, 1995.

[HSE07] D Hoiem, A N Stein, A A Efros, and M Hebert. "Recovering Occlusion Boundaries from a Single Image." In *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

[Hug02] R. Hughes. "A continuum theory for the flow of pedestrians." *Transportation Research Part B*, 2002.

[Hug03] R. Hughes. "The Flow of Human Crowds." *Annual Review of Fluid Mechanics*, 2003.

[HWN08] Chang Huang, Bo Wu, and Ramakant Nevatia. "Robust Object Tracking by Hierarchical Association of Detection Responses." In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pp. 788–801, Berlin, Heidelberg, 2008. Springer-Verlag.

[HXF06] Weiming Hu, Xuejuan Xiao, Zhouyu Fu, Dan Xie, Tieniu Tan, and Steve Maybank. "A System for Learning Statistical Motion Patterns." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**:1450–1464, 2006.

[HZC04] X. He, R. Zemel, and M. Carreira-Perpinan. "Multiscale conditional random fields for image labelling." In *In 2004 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[ID07] N. Ikizler and P. Duygulu. "Human Action Recognition Using Distribution of Oriented Rectangular Patches." 2007. ICCV.

[JFE03] Allan D. Jepson, David J. Fleet, and Thomas F. El-Maraghi. "Robust Online Appearance Models for Visual Tracking." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**:1296–1311, 2003.

[JGJ99] M I Jordan, Z Ghahramani, T S Jaakkola, and L K Saul. "An introduction to variational methods for graphical models." In M I Jordan, editor, *Learning in Graphical Models*. MIT Press, Cambridge, 1999.

[JH96] Neil Johnson and David Hogg. "Learning the distribution of object trajectories for event recognition." *Image and Vision Computing*, **14**(8):609 – 615, 1996. 6th British Machine Vision Conference.

[JHS08] A. Johansson, D. Helbing, and P. Shukla. "Specification of a Microspopic Pedestrian Model by Evolutionary Adjustment to Video Tracking Data." *Advances in Complex Systems*, 2008.

[JJS04] Imran N. Junejo, Omar Javed, and Mubarak Shah. "Multi Feature Path Modeling for Video Surveillance." *Pattern Recognition, International Conference on*, **2**:716–719, 2004.

[JS02] O. Javed and M. Shah. "Tracking And Object Classification For Automated Surveillance." In *ECCV*, 2002.

[JU97] Simon J. Julier and Jeffrey K. Uhlmann. "A New Extension of the Kalman Filter to Nonlinear Systems." pp. 182–193, 1997.

[JVV03] Michael Jones, Paul Viola, Paul Viola, Michael J. Jones, Daniel Snow, and Daniel Snow. "Detecting Pedestrians Using Patterns of Motion and Appearance." In *In ICCV*, pp. 734–741, 2003.

[KAO11] Yamaguchi K., Berg A., Ortiz O., and Berg T. "Who are you with and Where are you going?" *CVPR*, 2011.

[KBD05] Z. Khan, T. Balch, and F. Dellaert. "MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets." *PAMI*, 2005.

[KH03] Sanjiv Kumar and Martial Hebert. “Discriminative Random Fields: A Discriminative Framework for Contextual Interaction in Classification.” In *Proceedings of the 2003 IEEE International Conference on Computer Vision (ICCV ’03)*, volume 2, pp. 1150–1157, 2003.

[KMT07] Pushmeet Kohli, Pawan Mudigonda, and Philip Torr. “P^3 and Beyond: Solving Energies with Higher Order Cliques.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR07)*, 2007.

[Kol06] Vladimir Kolmogorov. “Convergent Tree-reweighted Message Passing for Energy Minimization.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **28**(10):1568–1583, October 2006.

[KR07] Franziska Klügl and Guido Rindsfüser. “Large-Scale Agent-Based Pedestrian Simulation.” In *Proceedings of the 5th German conference on Multiagent System Technologies*, MATES ’07, pp. 145–156, Berlin, Heidelberg, 2007. Springer-Verlag.

[KSC07] S. Sathiya Keerthi, Vikas Sindhwani, and Olivier Chapelle. “An Efficient Method for Gradient-Based Adaptation of Hyperparameters in SVM Models.” In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pp. 673–680. MIT Press, Cambridge, MA, 2007.

[KSH05] Y. Ke, R. Sukthankar, and M. Hebert. “Efficient Visual Event Detection using Volumetric Features.” 2005. ICCV.

[KTZ05] M. P. Kumar, P. H. S. Torr, and A. Zisserman. “OBJ CUT.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego*, 2005.

[LFD07] Anat Levin, Rob Fergus, Frédo Durand, and William T. Freeman. “Image and depth from a conventional camera with a coded aperture.” In *ACM SIGGRAPH 2007*, p. 70, New York, NY, USA, 2007. ACM.

[LH05a] Yann LeCun and Fu Jie Huang. “Loss Functions for Discriminative Training of Energy-Based Models.” In *Proc. of the 10-th International Workshop on Artificial Intelligence and Statistics (AIStats’05)*, 2005.

[LH05b] Yann LeCun and Fu Jie Huang. “Loss Functions for Discriminative Training of Energy-Based Models.” In *Proc. of the 10-th International Workshop on Artificial Intelligence and Statistics (AIStats’05)*, 2005.

[LH08] Yunpeng Li and Daniel P. Huttenlocher. “Learning for Optical Flow Using Stochastic Optimization.” *ECCV*, 2008.

[LHO96] J. Larsen, L. Hansen, and C. Ohlsson. “Design and Regularization of Neural Networks: The Optimal Use of A Validation Set.” In *Neural Networks for Signal Processing*, 1996.

[Li95] S Z Li, editor. *Markov Random Field Modeling in Computer Vision*. Springer, 1995.

[LKF05] T. Lakoba, D. Kaup, and N. Finkelstein. “Modifications of the Helbing-Molnár-Farkas-Vicsek Social Force Model for Pedestian Evolution.” *SIMULATION*, 2005.

[LL03] I. Laptev and T. Lindeberg. “Interest Point Detection and Scale Selection in Space-Time.” *Scale Space Methods in Computer Vision*, 2003.

[LLS04] B. Leibe, A. Leonardis, and B. Schiele. “Combined Object Categorization and Segmentation with an Implicit Shape Model.” In *ECCV’04 Workshop on Statistical Learning in Computer Vision*, pp. 17–32, Prague, Czech Republic, May 2004.

[LMP01] John Lafferty, Andrew McCallum, and Fernando Pereira. “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.” In *Proc. 18th International Conf. on Machine Learning*, pp. 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[Low04] D. Lowe. “Distinctive image features from scale-invariant keypoints.” *IJCV*, 2004.

[LPM01] John Lafferty, Fernando Pereira, and Andrew McCallum. “Conditional random fields: Probabilistic models for segmenting and labeling sequence data.” In *ICML*, 2001.

[LSA96] Jan Larsen, Claus Svarer, Lars Nonboe Andersen, and Lars K. Hansen. “Adaptive Regularization in Neural Network Modeling.” In *Neural Networks: Tricks of the Trade*, pp. 113–132, 1996.

[LSP07] S. Lazebnik, C. Schmid, and J. Ponce. “Beyond Bags of Features: Spatial Pyramid Matching for Recognizing natural Scene Categories.” 2007. CVPR.

[LST10] M. Luber, J.A. Stork, G.D. Tipaldi, and K.O. Arras. “People tracking with human motion predictions from social forces.” In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 464 –469, may 2010.

[LSV07] B. Leibe, K. Schindler, and L. Van Gool. “Coupled Detection and Trajectory Estimation for Multi-Object Tracking.” In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1 –8, oct. 2007.

[LW06a] A Levin and Y Weiss. "Learning to Combine Bottom-Up and Top-Down Segmentation." In *European Conference on Computer Vision (ECCV)*, Graz, Austria, May 2006.

[LW06b] Anat Levin and Yair Weiss. "Learning to Combine Bottom-Up and Top-Down Segmentation." In *ECCV (4)*, pp. 581–594, 2006.

[MB02] Derek Magee and Roger Boyle. "Detecting lameness using 'Re-sampling Condensation' and 'multi-stream cyclic hidden Markov models'." *Image and Vision Computing*, **20(8)**:581–594, 2002.

[ME02] Dimitrios Makris and Tim Ellis. "Spatial and Probabilistic Modelling of Pedestrian Behaviour." In *British Machine Vision Conference 2002, vol.2*, pp. 557–566, 2002.

[ME05] D. Makris and T. Ellis. "Learning semantic scene models from observing activity in visual surveillance." *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, **35**(3):397 –408, june 2005.

[MFT01] D. Martin, C. Fowlkes, D. Tal, and J. Malik. "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics." In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pp. 416–423, July 2001.

[MLB10] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. "Anomaly detection in crowded scenes." *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, **0**:1975–1981, 2010.

[MMS10] Ramin Mehran, Brian Moore, and Mubarak Shah. "A Streakline Representation of Flow in Crowded Scenes." In *Computer Vision ECCV 2010*, volume 6313 of *Lecture Notes in Computer Science*, pp. 439–452. Springer Berlin Heidelberg, 2010.

[MPG10] Mehdi Moussaid, Niriaska Perozo, Simon Garnier, Dirk Helbing, and Guy Theraulaz. "The walking behaviour of pedestrian social groups and its impact on crowd dynamics." 2010.

[MS09] R. Mehran and M. Shah. "Abnormal Crowd Behavior Detection using Social Force Model." *CVPR*, 2009.

[NE86] H. Nagel and W. Enkelmann. "An investigation of smoothness constraint for the estimation of displacement vector fiels from images sequences." *IEEE Transactions of Pattern Analysis and Machine Intelligence*, **8**:565–593, 1986.

[Nea93] Radford M. Neal. "Probabilistic Inference Using Markov Chain Monte Carlo Methods." Technical Report CRG-TR-93-1, University of Toronto, Dept. of Computer Science, 1993.

[NWF06] J. Niebles, H. Wang, and L. Fei-Fei. "Unsupervised Learning of Human Action Categories Using Spatial-TemporalWords." *BMVC*, 2006.

[ORP00] N. Oliver, B Rosario, and A. Pentland. "A Bayesian Computer Vision System for Modeling Human Interactions." *PAMI*, 2000.

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, second edition, 1988.

[PEG10] S. Pellegrini, A. Ess, and L. van Gool. "Improving Data Association by Joint Modeling of Pedestrian Trajectories and Groupings." *ECCV*, 2010.

[PES09] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. "You'll Never Walk Alone: Modeling Social Behavior for Multi-target Tracking." *ICCV*, 2009.

[PET10] S. Pellegrini, A. Ess, M. Tanaskovic, and L. van Gool. "Wrong Turn - No Dead End: a Stochastic Pedestrian Motion Model." *International Workshop on Socially Intelligent Surveillance and Monitoring (SISM)*, 2010.

[PEV10] Stefano Pellegrini, Andreas Ess, and Luc Van Gool. "Improving Data Association by Joint Modeling of Pedestrian Trajectories and Groupings." In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision ECCV 2010*, volume 6311 of *Lecture Notes in Computer Science*, pp. 452–465. Springer Berlin Heidelberg, 2010.

[PSW03] J. Portilla, V. Strela, M. Wainwright, and E.P. Simoncelli. "Image Denoising Using Scale Mixtures of Gaussians in the Wavelet Domain." *IEEE Transactions on Image Processing*, **12**(11):1338–1351, November 2003.

[PT01] Alan Penn and Alasdair Turner. "Space syntax based agent simulation." In *in M. Schreckenberg and S. Sharma (Eds.), Pedestrian and Evacuation Dynamics, 99114*, pp. 99–114. Springer, 2001.

[PVB04] Patrick Perez, Jaco Vermaak, and Andrew Blake. "Data Fusion for Visual Tracking with Particles." In *Proceedings of the IEEE*, pp. 495–513, 2004.

[RA79] J. W. Roach and J. K. Aggarwal. "Computer Tracking of Objects Moving in Space." *PAMI*, 1979.

[RAK09] M. Rodriguez, S. Ali, and T. Kanade. "Tracking in unstructured crowded scenes." In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1389 –1396, 29 2009-oct. 2 2009.

[RB05a] S. Roth and M. Black. “Fields of experts: A framework for learning image priors.”, 2005.

[RB05b] Stefan Roth and Michael Black. “Field of Experts: A Framework for Learning Image Priors.” *CVPR*, 2005.

[RCM91] Anand Rangarajan, Rama Chellappa, and B. S. Manjunath. “Markov random fields and neural networks with applications to early vision problems.” In I. K. Sethi and A. K. Jain, editors, *Artificial Neural Networks and Statistical Pattern Recognition: Old and New Connections*, pp. 155–174. Elsevier Science Press, 1991.

[Rey87] C. Reynolds. “Flocks, Herds, and Schools: A Distributed Behavioral Model.” *ACM SIGGRAPH*, 1987.

[RM03] Xiaofeng Ren and Jitendra Malik. “Learning a classification model for segmentation.” In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pp. 10–17, 2003.

[RS99] R. Rosales and S. Sclaroff. “3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions.” In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pp. 2 vol. (xxiii+637+663), 1999.

[RTM05] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. “LabelMe: a Database and Web-Based Tool for Image Annotation.” *MIT AI Lab Memo AIM-2005-025*, 2005.

[SAS07] P. Scovanner, S. Ali, and M. Shah. “A 3-Dimensional SIFT Descriptor and its Application to Action Recognition.” *ACM Multimedia*, 2007.

[SBS09a] Fahad Shah, Philip Bell, and Gita Sukthankar. “Agent-Assisted Navigation for Virtual Worlds.” In *Proceedings of the 9th International Conference on Intelligent Virtual Agents*, IVA ’09, pp. 543–544, 2009.

[SBS09b] Fahad Shah, Philip Bell, and Gita Sukthankar. “Identifying User Destinations in Virtual Worlds.” *AAAI*, 2009.

[SC10] W. Sultani and J. Young Choi. “Abnormal Traffic Detection using Intelligent Driver Model.” *ICPR*, 2010.

[SDT07] Kegan G G Samuel, Craig V Dean, Marshall F Tappen, and David M Lyle. “Learning to Segment with Logistic Random Fields.” In Submission to Neural Information Processing Systems (NIPS) 2007, 2007.

[SG99] C. Stauffer and W. Grimson. "Adaptive background mixture models for real-time tracking." *CVPR*, 1999.

[SG00] C. Stauffer and W.E.L. Grimson. "Learning patterns of activity using real-time tracking." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **22**(8):747 –757, aug 2000.

[She] Jonathan Richard Shewchuk. "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain." Available at `http://www.cs.cmu.edu/~jrs/jrspapers.html`.

[SIF03] E. Sudderth, A. Ihler, W.T. Freeman, and A. Willsky. "Nonparametric Belief Propagation." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[SK01] S. Sundararajan and S. Sathiya Keerthi. "Predictive Approaches for Choosing Hyperparameters in Gaussian Processes." *Neural Computation*, **13**(5):1103–1118, 2001.

[Sla07] Matt Slaughter. "Tracking Pedestrians With Machine Vision." *National Instruments*, 2007.

[SLC04] C. Schuldt, I. Laptev, and B. Caputo. "Recognizing Human Actions: A Local SVM Approach." *ICPR*, 2004.

[SP07] Daniel Scharstein and Chris Pal. "Learning Conditional Random Fields for Stereo." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, June 2007.

[SS02] Daniel Scharstein and Richard Szeliski. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms." *International Journal of Computer Vision*, **47**(1/2/3), April-June 2002.

[SSS09] I. Saleemi, K. Shafique, and M. Shah. "Probabilistic Modeling of Scene Dynamics for Applications in Visual Surveillance." *PAMI*, 2009.

[ST07] W. Shao and D. Terzopoulos. "Autonomous Pedestrians." *Graphical Models*, 2007.

[ST09] P. Scovanner and M. F. Tappen. "Learning Pedestrian Dynamics from the Real World." *ICCV*, 2009.

[Sta03] Chris Stauffer. "Estimating Tracking Sources and Sinks." In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW '03. Conference on*, volume 4, p. 35, june 2003.

[SZS06a] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall F. Tappen, and Carsten Rother. "A Comparative Study of Energy Minimization Methods for Markov Random Fields." In *ECCV (2)*, pp. 16–29, 2006.

[SZS06b] Rick Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. "A Comparative Study of Energy Minimization Methods for Markov Random Fields." In *Seventh European Conference on Computer Vision (ECCV 2006)*, volume 2, pp. 16–29. Springer-Verlag, May 2006.

[TAF06] Marshall F. Tappen, Edward H. Adelson, and William T. Freeman. "Estimating Intrinsic Component Images using Non-Linear Regression." In *The Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pp. 1992–1999, 2006.

[Tap07a] M. F. Tappen. "Utilizing Variational Optimization to Learn Markov Random Fields." *CVPR*, 2007.

[Tap07b] M. F. Tappen. "Utilizing Variational Optimization to Learn Markov Random Fields." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR07)*, 2007.

[Tas04] B. Taskar. "Learning structured prediction models: A large margin approach.", 2004.

[TCK05] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. "Learning Structured Prediction Models: A Large Margin Approach." In *ICML*, 2005.

[TCP06] A. Treuille, S. Cooper, and Z. Popović. "Continuum Crowds." *ACM SIGGRAPH*, 2006.

[Ter86] D. Terzopoulos. "Regularization of inverse visual problems involving discontinuities." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**:413–424, 1986.

[TFA03] M. F. Tappen, W. T. Freeman, and E. H. Adelson. "Recovering intrinsic images from a single image." In *NIPS*, pp. 1343–1350, 2003.

[TFA05] Marshall F. Tappen, William T. Freeman, and Edward H. Adelson. "Recovering Intrinsic Images from a Single Image." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(9):1459–1472, September 2005.

[TFF05] Antonio Torralba, Li Fei-Fei, and Rob Fergus. "Recognizing and Learning Object Categories.", 2005.

[Thr02] Sebastian Thrun. “Probabilistic robotics.” *Commun. ACM*, **45**:52–57, March 2002.

[THS04] Tom Troscianko, Alison Holmes, Jennifer Stillman, Majid Mirmehdi, Daniel Wright, and Anna Wilson. “What happens next? The predictability of natural behaviour viewed through CCTV cameras.” *Perception*, **33**(1):87–101, 2004.

[TK10] P. Trautman and A. Krause. “Unfreezing the robot: Navigation in dense, interacting crowds.” In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 797 –803, oct. 2010.

[TLA07] M. F. Tappen, C. Liu, E. H. Adelson, and W. T. Freeman. “Learning Gaussian Conditional Random Fields for Low-Level Vision.” In *CVPR*, 2007.

[TLJ06] Ben Taskar, Simon Lacoste-Julien, and Michael Jordan. “Structured Prediction via the Extragradient Method.” In *Advances in Neural Information Processing Systems 18*, pp. 1345–1352. MIT Press, Cambridge, MA, 2006.

[TMF05] Antonio Torralba, Kevin P. Murphy, and William T. Freeman. “Contextual Models for Object Detection Using Boosted Random Fields.” In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pp. 1401–1408. MIT Press, Cambridge, MA, 2005.

[TRF04] Marshall F. Tappen, Bryan C. Russell, and William T. Freeman. “Efficient Graphical Models for Processing Images.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pp. 673–680, 2004.

[TS09] B. Tastan and G. Sukthankar. “Exploiting human steering models for path prediction.” In *Information Fusion, 2009. FUSION ’09. 12th International Conference on*, pp. 1722 –1729, july 2009.

[TS10] B. Tastan and G. Sukthankar. “Leveraging Human Behavior Models to Predict Paths in Indoor Environments.” *Pervasive and Mobile Computing*, 2010.

[TT10] Kuo-Shih Tseng and A.C.-W. Tang. “Goal-oriented and map-based people tracking using virtual force field.” In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 3410 –3415, oct. 2010.

[VSS06] S. Vishwanathan, N. Schraudolph, M. Schmidt, and K. Murphy. “Accelerated Training of Conditional Random Fields with Stochastic Meta-Descent.” In *International Conference on Machine Learning (ICML ’06)*, 2006.

[Wai06] Martin J Wainwright. “Estimating the “wrong” graphical model: Benefits in the computation-limited setting.” *Journal of Machine Learning Research*, **7**:1829–1859, September 2006.

[WBS04] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. “Image Quality Assessment: From Error Visibility to Structural Similarity.” *IEEE Trans. Image Processing*, **13**(4):600–612, April 2004.

[WJ03] M. Wainwright and M. Jordan. “Semidefinite relaxations for approximate inference on graphs with cycles.”, 2003.

[WJ05] J Winn and N Jojic. “LOCUS:learning object classes with unsupervised segmentation.” In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pp. 756–763 Vol. 1, 2005.

[WJW05a] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. “MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches.” *IEEE Transactions on Information Theory*, **51**(11):3697–3717, November 2005.

[WJW05b] M J Wainwright, T S Jaakkola, and A S Willsky. “A New Class of Upper Bounds on the Log Partition Function.” *IEEE Transactions on Information Theory*, **51**(7):2313–2335, July 2005.

[WMG09] X. Wang, X. Ma, and E. Grimson. “Unsupervised Activity Perception in Crowded and Complicated Scenes Using Hierarchical Bayesian Models.” *PAMI*, 2009.

[WN07] Bo Wu and Ram Nevatia. “Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors.” *International Journal of Computer Vision*, **75**:247–266, 2007. 10.1007/s11263-006-0027-7.

[YFW00] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. “Generalized Belief Propagation.” In *NIPS*, pp. 689–695, 2000.

[YJS06] A. Yilmaz, O. Javed, and M. Shah. “Object Tracking: A Survey.” *ACM Computing Surveys*, 2006.

[YYD07] Qingxiong Yang, Ruigang Yang, James Davis, and David Nister. “Spatial-Depth Super Resolution for Range Images.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR07)*, 2007.

[ZCM04] Shaohua Kevin Zhou, R. Chellappa, and B. Moghaddam. “Visual tracking and recognition using appearance-adaptive models in particle filters.” *Image Processing, IEEE Transactions on*, **13**(11):1491 –1506, nov. 2004.

[ZLN08] Li Zhang, Yuan Li, and R. Nevatia. “Global data association for multi-object tracking using network flows.” In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1 –8, june 2008.

[ZM97] Song Chun Zhu and David Mumford. “Prior Learning and Gibbs Reaction-Diffusion.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(11):1236–1250, November 1997.

[ZN03] Tao Zhao and Ram Nevatia. “Bayesian Human Segmentation in Crowded Situations.” *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, **2**:459, 2003.

[ZN04] Tao Zhao and R. Nevatia. “Tracking multiple humans in crowded environment.” In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pp. II–406 – II–413 Vol.2, june-2 july 2004.

[ZRG09] B.D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J.A. Bagnell, M. Hebert, A.K. Dey, and S. Srinivasa. “Planning-based prediction for pedestrians.” In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 3931 –3936, oct. 2009.