Low Latency Tracking and Anomaly Detection in Pedestrian Crowds from Video Data

A Dissertation Presented
by

Oliver Lehmann

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Electrical Engineering

Northeastern University
Boston, Massachusetts

January 2016
To my family and friends.
# Contents

Abstract vii

1 Introduction 1
   1.1 Background ......................................................... 2
   1.2 Present Contributions ............................................. 8
   1.3 Thesis Outline .................................................... 9

2 Anomaly Detection in Large, Dense Crowds 10
   2.1 Introduction ....................................................... 10
   2.2 Crowd Anomaly Detection Framework .............................. 13
   2.3 Crowd Model ...................................................... 17
   2.4 Crowd Model Parameter Estimation ............................... 35
   2.5 Crowd Behavior and Anomaly Detection .......................... 43
   2.6 Anomaly Detection Results ....................................... 47
   2.7 Concluding Remarks and Further Research Recommendations . 54

3 Accelerated Video Labeling for High Quality Ground Truth Generation 56
   3.1 Introduction ....................................................... 56
   3.2 Video Labeling Data Framework ................................. 58
   3.3 Operational Support .............................................. 60
   3.4 The Software Suite “ANchOVy” ................................. 65
   3.5 Concluding Remarks and Further Research Recommendations . 68

4 Trajectory Recovery from Short Sequences of Noise Corrupted Observations 70
   4.1 The Noisy Trajectory Recovery Problem ....................... 70
   4.2 Order-selective Weighted Hankel Total Least Norm ............ 72
   4.3 Performance Evaluation .......................................... 82
# Bibliography

## A Microscopic Crowd Simulation

A.1 Optimal Reciprocal Collision Avoidance Algorithm .......................... 108
A.2 Extension of the ORCA Algorithm ................................................. 111
A.3 Simulations .................................................................................. 118

## B Equality Constrained Least Squares

B.1 Exact Solution .............................................................................. 122
B.2 Approximate Solution ................................................................. 122
B.3 Iterative Solution ......................................................................... 124

## List of Acronyms

## Index
# List of Figures

1.1 Examples of large, dense crowds .................................................. 1
1.2 Surveillance cameras at JFK airport checkpoint ................................. 6

2.1 Crowd anomaly detection framework ............................................... 14
2.2 Example of a crowded scene .......................................................... 16
2.3 Knudsen number as a function of normalized density for pedestrian crowds .......................................................... 30
2.4 Temporal evolution of lane formation in a straight corridor .................. 32
2.5 Multiple lanes in a wide corridor ...................................................... 33
2.6 Temporal evolution of vortex formation on a pedestrian intersection ....... 34
2.7 Anomaly detection parameter estimation for a local area in a bidirectional crowd from high sample rate data .......................... 41
2.8 Anomaly detection parameter estimation for a local area in a corridor setup from low sample rate data .................................. 42
2.9 Distributed parameters of the crowd motion model for the two-lane corridor .......................................................... 43
2.10 Motion of a two-population crowd under external forcing ................. 46
2.11 Temporal evolution of estimated parameters of a bidirectional crowd under external forcing .......................................................... 47
2.12 Distributed estimated parameters of the crowd motion model of a bidirectional crowd under external forcing .......................... 48
2.13 Anomaly detection – Example “Egress” ............................................ 49
2.14 Anomaly detection – Example “Egress” – ROC .................................. 50
2.15 Anomaly detection – Example “External Forcing” .............................. 51
2.16 Anomaly detection – Example “External Forcing” – ROC ................. 52
2.17 Anomaly detection – Example “Small group” ..................................... 53
2.18 Anomaly detection – Example “Small group” – Anomaly score ............ 54

3.1 Video label classification and corresponding examples .......................... 59
List of Tables

2.1 Crowd motion model parameters ................................................. 31
A.1 ORCA crowd model parameters ............................................... 118
Acknowledgements

Looking back, I don’t think that I knew what was lying ahead of my when I arrived in Boston. If not for the assistance and support of a great number of people, this dissertation would not have seen the light of day. For their love, friendship, support, and professional guidance I gratefully acknowledge the following people.

First and foremost, I would like to thank my advisor, Dr. Gilead Tadmor. I was truly lucky to have a mentor with Gilead’s patience, understanding, and incredible depth of knowledge on my side. He was responsible for involving me in so many exciting projects in the first place, was always there to listen, to talk about research, philosophy, and life, and showed me different ways to approach a research problem and the need to be persistent to accomplish any goal. Gilead, your help, stimulating suggestions, and mentoring helped me beyond doing research and writing this thesis.

Special thanks go to Dr. Octavia Camps and Dr. Mario Sznaier for participating in my project committee. I thank you for your continuous support, patience, and for your invaluable suggestions on how to improve my research.

Thanks go to Dr. Michael Silevitch and John Beaty for their guidance and time, both of which they gave generously. I would also like to thank the faculty and staff at Northeastern’s ALERT center for their support, wisdom, and camaraderie. In particular, I thank Emel Bulat for her consulting, mentoring, and sharing her valuable business knowledge. I also owe Dianna Beirne a debt of gratitude for her efforts to support me in all matters of IT.

My thanks go also to my fellow students of the Robust System Lab hosted at Northeastern University for many valued discussions both on-topic and off-topic and my former colleagues involved the Video Analytics program – Rick Moore (Thanks so much for everything!), Tom Hebble, Fei Xiong, and Binlong Li – for their vision, dedication, and hard work that allowed our project to succeed. It was an honor to work with all of you.

I also express my thanks to the Point-blank Research (PBR) team for memorable evenings and their support in completing the present work. But most importantly, I would like to sincerely thank my family and friends here and back home for their support and understanding.

The author also acknowledges excellent working conditions at the Department of Homeland Security Center of Excellence for Awareness and Localization of Explosives-Related Threats (ALERT), which is funded by the U.S. Department of Homeland Security, Science and Technology Directorate, Office of University Programs, under Grant Awards 2013-ST-061-ED0001.

I recall thinking that the computer would never advance much further than this. Call me naïve, but I seemed to have underestimated the universal desire to sit in a hard plastic chair and stare at a screen until your eyes cross.

– David Sedaris
Abstract

Low Latency Tracking and Anomaly Detection in Pedestrian Crowds from Video Data

by

Oliver Lehmann

Doctor of Philosophy in Electrical and Computer Engineering

Northeastern University, January 2016

Dr. Gilead Tadmor, Adviser

A low latency behavior analysis of human crowds is important for the prevention of crowd disasters. It is critical, in particular, in a security context because of the opportunity to cause large damage through malicious actions as we have seen during the Boston Marathon on April 15, 2013. Ensuring safety through continuous crowd monitoring by humans is impractical in heavily utilized pedestrian scenes and a computer vision-based tracking of each individual in a large crowd is virtually impossible due to the combination of sheer numbers, the proximity between people, and partial occlusion.

Here we consider a method for real-time identification of motion anomalies in large, dense crowds. Our method addresses the prohibitive complexity of individual-based motion analysis methods within a vision of a layered, multi-scale framework. Specifically, here we focus on the first stage where coarse grain/macroscopic level analysis identifies and localizes suspected anomalies, allowing the subsequent focus of sensing and computational resources and enabling the use of finer grain methods, based on individual motion analysis, in small suspect areas.

To make low-latency, coarse grain motion analysis of a large and dense crowd computationally feasible we make a continuum assumption, modeling the entire crowd as a continuous fluid. By this approach, crowd dynamics are described by a set of parameterized partial differential equations (PDEs) inspired by the compressible Navier-Stokes equations and include terms for driving purposeful motion, density aversion, multiple desired paths, crowd viscosity effects, and a long observed natural stochasticity, such as in pedestrians collusion avoidance maneuvers. Using numerical simulations we demonstrate that, indeed, this model is capable of reproducing some well-known self-organization phenomena of human crowds.
The sought complexity reduction is achieved by characterized localized crowd behavior in terms of the values and temporal variations of the identified local parameters of governing PDEs. We suggest auxiliary, low order dynamical models for the evolution of those parameters and the use of such models for the identification of abrupt parameter changes which indicate a local crowd motion anomaly.

Algorithms employing video data, such as our crowd anomaly detection, require a comprehensive video footage database with accompanying high-accuracy labels for tuning and performance testing. Since the accurate labeling of videos is a highly cost intensive process, especially for large crowds, the development of sophisticated algorithms is typically limited to an economized label database to meet project budget constraints. To overcome this limitation, we developed methods to accelerate the generation of high quality video labels through the deployment of a flexible annotation framework with computer-aided labeling support.

This material is based upon work supported by the U.S. Department of Homeland Security, Science and Technology Directorate, Office of University Programs, under Grant Award 2013-ST-061-ED0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.
Introduction

With the increase of the world population and ongoing urbanization, human gatherings take place more frequently and crowds larger in size. This poses challenges in urban planning, crowd management, and public safety for the fact that large crowds can be uncontrollable, dangerous, or even deadly forces.

The gathering of large numbers of individuals for special events (e.g., sports, arts, political, or religious reasons), as shown in the examples presented in Fig. 1.1, can create situations that are unpredictable and volatile, potentially resulting in mass panics or stampedes. A 2014 study of mass casualty incidents involving crowds reports 162 such incidents over the cause of two decades worldwide, each resulting in injuries and the loss of many lives [1].

All crowd related disasters, spanning continents, cultures, and events, are caused by some trigger event. Many of these incidents are the result of selfish, uncoordinated behavior of a

Figure 1.1: Examples of crowds. Structured crowd at the Ganesh festival in Mumbai, India, (left) and an unstructured gathering of people on a square in Ulm, Germany (right).
1. INTRODUCTION

Single individual/small group, commonly triggered by perceived or real threats, fire, a simple fight, fear of endangerment, overcrowding, spatial limitations, extreme weather conditions, and crowd demographics. In response, adequate crowd management and effective security measures provide the means to prevent or at least mitigate the effects of panic situations.

Besides disasters there is also a potential security threat when large numbers of individuals gather. Human crowds are vulnerable because of the opportunity to cause large damage through malicious actions as we have seen during the Boston Marathon 2013 or the Paris attacks in 2015. The recurrence of this type of event drives the increased demand for public safety at spaces such as airports, train stations, stadiums, or shopping malls.

1.1 Background

Proactive security measures start with sufficient planning of public spaces and events and continue with the online monitoring of crowds. The latter requires a real-time understanding of large crowd behavior, which is provided through online crowd surveillance. In a typical surveillance situation, a large crowded scene is monitored by human observers, who simultaneously view a number of video signals, monitoring the crowd for abnormal situations. To allow for a timely response this task has to be conducted in real time. Humans are good at those monitoring task, but severely challenged when they are tasked to evaluate too much data in too little time [2].

To support preventative security measures, crowded scene analysis and the study of human behavior in densely crowded environments has been an active area of research, especially in the fields of system modeling and computer vision. Technological advances both in surveillance equipment and computational capacities are the foundation for the shift from passive to predictive surveillance. The current research is based on the common understanding that conventional techniques are insufficient for their application to dense crowd analysis due to complex dynamics, large variations of densities, and strong occlusions in the scene. As a result, there has been a shift to new, more advanced techniques that rely on a higher computing power, allowing for a crowd anomaly detection in large, dense crowds.

The tasks for automated crowd anomaly detection cover a wide range of research domains and employing a manifold of techniques. Those tasks are crowd sensing, crowd modeling, behavior modeling, classification/learning, and anomaly detection, all of which we describe subsequently. How those tasks are performed depends on the principle differentiation on what scale the crowd

---

1 Since the 170s, the computing power increased by about 7 orders of magnitude.
1. INTRODUCTION

is resolved. Typically, approaches are classified by their observed resolution as either micro-scale, meso-scale, or macro-scale. Where microscopic approaches recognize single individuals that make up a crowd, the macroscopic philosophy treats a crowd as a whole, using holistic properties to describe and analyze its behavior. Mesoscopic techniques combine the previous two approaches and may be characterized as small group models. The survey studies by Zhan et al. [3] and more recently Thida et al. [4] both present an overview of approaches focusing specifically on crowd video analysis.

Crowd Sensing

Crowd sensing is the task of determining the crowd’s properties, such as velocity, density, and flow, from sensed data. Since most crowd surveillance is conducted with CCTV cameras, this task employs computer vision techniques applied to videos of crowded scenes. Methods that work well for less crowded scenes, which are discussed in [5], are not applicable to densely populated scenes and specialized approaches are required.

Microscopic crowd sensing is accomplished through the detection of individuals in a crowd [6], tracking them [7], and analyzing the gathered trajectories to describe the crowd motion state. With respect to crowd dynamics modeling, one of the microscopic approaches’ benefits is that both velocity and density – together they make up the crowd motion state – are obtained. Extracting reliable trajectories from heavily crowded scenes is a difficult task not only because of the sheer number of tracking targets but also due to the interactions between individuals in the crowd and strong occlusions. Particle filter-based methods [8] using a wide variety of features for single-target and multi-target tracking are used most often. Improved results are achieved by using mesoscopic crowd properties that are obtained by an analysis of crowd dynamics. For example, contextual cues [9] and social interaction forces [10] may be used to supplement tracking algorithms as priors to predict motion patterns, improving tracking performance.

Macroscopic crowd sensing does not rely on feature extraction from single individuals making up the crowd. According to Thida’s survey [4], those methods can be categorized into two groups: Optical flow-based methods [11] and those based on local spatio-temporal gradients. The dense field of instantaneous displacements (or velocities) of pixels (or small patches) in consecutive video frames are called optical flow (OF) and may be obtained by any of the quasi-standard OF methods e.g. [12, 13, 14].

Macroscopic OF-based approaches employ the optical flow vectors to model pedestrian
1. INTRODUCTION

motion dynamics by clustering the vectors into dominant motion patterns. Again, many such methods were developed some of which use spatial, temporal, or spatio-temporal filtering others use statistical modeling and graph theory [15, 16]. More sophisticated ones are based on machine learning techniques to extract the desired crowd motion state [17].

Spatio-temporal gradient methods use shape, size, and texture as features, representing the crowd and its motion [18] and typically rely on image/video processing and pattern recognition techniques. For example, crowd density may be deduced from the texture of a surveillance image. This method assumes that images of low (high) density crowds tend to present coarse (fine) texture [19, 20].

Crowd Modeling and Simulations

While a crowd is defined as a gathering of individuals who share a purpose, crowd behavior is the behavior that is conducted by individuals who gather in a crowd. In other words, when entities belonging to the same group are acting in a coordinated behavior influenced by collective decisions rather than individual goals, those entities exhibit similar behavior and are acting as one crowd. The task of understanding how and why large groups of people behave the way they do requires a theory of crowd behavior for the benefit of, for example, public space planning, public safety management, human factors analysis in building egress, or even computer animation for entertainment purposes. There are many applications which require a tool to model virtual crowds [21].

Crowd simulations are based on models for the coordinated behavior of individuals in a crowd based on theories and methods of a variety of disciplines such as sociology, psychology, robotics, computer vision, and artificial intelligence. Again, those models can be grouped by their methodology into microscopic, mesoscopic, or macroscopic approaches. The choice of methodology is based on the simulation’s focus: Individual-centric models are generally microscopic to allow for the observation of/analysis of reactive behavior, different personalities, psychological elements, or decision-based navigation, whereas simulations of large crowds, neglecting each individual’s motion for the sake of a realistic overall crowd movement, are typically macroscopic.

The very first simulations were microscopic and of empirical nature based on observations rather than theory: Examples are transition matrix models [22], stochastic models [23], and pedestrian route choice [24, 25]. However, the descriptive and predictive power of those early models was rather limited and consequently more sophisticated methods were developed. Among those are models that consider the pedestrians directional movement choices, which are determined by techniques such as
rule-based models [26], cellular automata [27], finite state machines, and physical models. Among the latter class is the social force model that was introduced by Helbing and Molnár [28], which extended the social force field theory of Lewin [29] and uses social analogs of physical forces including attractive and repulsive interactions, friction, dissipation, and stochastic fluctuations. Helbing’s model treats individuals as particles and models their motion by solving Newton’s equations of motion driven by the social forces. According to [21], the social force approach fails to model natural human behavior in high-density crowds as the model overreacts in response to the numerous impinging forces exerted by the many neighbors of individuals in those type of crowds.

The use of a macro-scale crowd models reaches back to the early 1970s, when Henderson published the first fluid-dynamics based crowd models [30, 31] based on vehicular traffic observations traffic flow on long crowded roads published by Lighthill [32]. Henderson’s conjecture that pedestrian crowds behave similar to gases or fluids was the foundation most macroscopic crowd models. However, a realistic fluid-dynamic theory for human behavior in crowds must consider effects due to their particular interactions and physical forces as mentioned above [33]. We will describe macro-scale methods in more detail in the next chapter.

Crowd Anomaly Detection

The true value of data lies in its application to some meaningful purpose and how it is actioned in a preexisting context. The relevant portion of data needs to be extracted; anomaly detection can achieve that. In general terms, anomaly detection is the identification of data which do not conform to expected patterns, so called anomalies or outliers. In a wide field of applications, the detection of outliers is a key enabler to rendering data actionable [34]. For example, detecting anomalous sections in prenatal magnetic resonance images of paediatric brains allows the early diagnosis of serious brain diseases for timely medical intervention, reducing infant mortality [35].

The concept of anomalies and their detection was known as early as in the late 19-th century, but only recently automated anomaly detection became highly relevant as a method to reduce the vast amount of gathered. The abundance of surveillance cameras, as for example seen in Fig. 1.2, give the ability to capture more video data than ever, but it also poses the problem of how to analyze the gathered data. According to the results of a 2015 report by MeriTalk, a public-private partnership targeting improved government IT, more than 50% of video surveillance data gathered by the U.S. federal government goes unanalyzed and only 38% are actually used for anomaly detection, leaving a vast amount of information unused. In general, the low use percentage of such gathered
1. INTRODUCTION

Figure 1.2: One checkpoint at John F. Kennedy International Airport, New York City, USA, equipped with more than 50 surveillance cameras.

Data is attributed to the fact that video analysis is resource intensive and prohibitively expensive. Here, anomaly detection can act as a filter that excludes irrelevant data from further analysis. In the field of crowd surveillance, anomaly detection may be employed to reduce the burden of human operators monitoring pedestrians to detect anomalous behavior.

One of the main problems of anomalous behavior detection is the subjective definition of anomalies, their interpretation, and underlying assumptions [36]. Typically, anomalies are considered to be infrequently occurring events that are significantly different from “normal” data [4]. For example, an anomaly may be defined as data having low probability given a pre-learned “normal” model [37], exhibiting specific events over an extended period of time [38], or if the data is anomalous in a specific context. Methods using the latter definition exploit the structure in the data set, which is implicitly part of the anomaly’s assumptions. This poses the problem of defining a context, which in some cases may not be possible. Other methods declare anomalies by considering events with deviating position and/or trajectory content [39]. *A priori* definitions of anomalous events are another option that are, for example, applied in a security context to detect specific events such as an intrusion of a secure area [40] or leaving or picking up an item [41]. In respect to crowds, research efforts resulted in various approaches to identify different crowd events and/or to detect anomalous events and for us, the goal is not so much to analyze normal crowd behavior, but rather to detect deviations from it.

Identifying behavioral anomalies from surveillance videos is a difficult task in any framework. The task is particularly challenging in large crowds because of the complexity that is associated with the tracking of a large number of individuals and prevents analysis in real-time. The detection task here builds on a developed model such as those presented in the last section, using the
1. INTRODUCTION

sensed crowd state as input. Most methods deducing anomalies rely on machine learning and/or classification algorithms to distinguish between the normal or anomalous state of the observed behavior.

The reporting of results for any anomaly detection method is typically either a score value or a label. Binary labels may be assigned to test data instances, which flag the instances as either “normal” or anomalous. On the other hand, scoring techniques [42] produce a continuous anomaly score for each test data instance, reflecting to which degree it is considered to be anomalous. Scores permit the postulation of thresholds, separating anomalies from “normal” data, which allows for a ranking of detection results for a prioritized further processing, for example.

Video Labeling

Success in anomaly detection using the aforementioned crowd sensing, some type of (semi-)supervised learning, model identification, and/or anomaly definitions requires the availability of many data samples for tuning, testing, and verification of involved algorithms and models. In particular, such data samples need to be associated with labels such as the location of individuals for crowd sensing or anomaly descriptors for the crowd anomaly detection. More general, video labeling, as an extension to image labeling, is the association of descriptive labels to video data, generating semantic information, such as object identities, events, and activities. The actual definition of labels, also commonly known as annotations, depends on their purpose. For example, annotations used for establishing object identity may comprise a unique object identifier and bounding boxes, which are spatial labels for the location and extent (e.g. height and width), for each video frame in which an object is visible. Other labels may be descriptors for human activities such as “picking up object” and “leaving object behind”, or for traffic activities “turning left” and “parking”. In general, the information carried by the labels is only limited by the versatility of their definition.

The importance of large, comprehensive databases of labeled videos (and images) has been stressed many times in literature for automated algorithm benchmarking of tracking, segmentation, object recognition, activity analysis, etc. [43, 44, 45, 46]. Generating such labeled databases is tedious, error-prone, and often prohibitively expensive. This is due to the fact that labeling is conducted by human (expert) operators, involving repeated manual operations of associating meaning to presented images/videos.

Efficiently creating temporally consistent spatial annotations across video frames with little human interaction, crucial for low-cost video labeling, requires some algorithmic support. In
1. **INTRODUCTION**

1995, Flickner et al. introduced an image content query system that delegated simple, nonsemantic information retrieval tasks, such as texture and color descriptors, to a computer algorithm [45]. Building on advances in the fields of computer vision, databases, and machine learning, other computer support has been introduced over the years with the intent to reduce the labeling effort and/or to support operations online. Regarding the latter, Yuen stated in 2009 that tracking algorithms are too computationally expensive and may prevent the labeling operators from interacting with videos in real time [47]. While the general-purpose annotation system ViPER (*video performance evaluation resource*) [48] avoids computer vision assistance altogether [46], the Labelme framework [49] uses a low-latency but inferior method of employing a homography-preserving shape interpolation that is aided with global motion estimation to propagate annotations temporally. More recent advances focus on minimizing the labeling effort by using active learning techniques [50].

Crowdsourcing the labeling task is an option to distribute the work load to many operators [51, 52] with the potential of increasing label quality [53]. However, involving a large number of people is an approach only valid for certain data sets; It is inapplicable for handling proprietary, sensitive, or even classified data.

1.2 **Present Contributions**

Successful crowd anomaly detection requires low-latency understanding of large crowd behavior, which in turn is based on the many challenging tasks described in the last section. The present work is concerned with crowd modeling, crowd behavior models, crowd anomaly detection, and video labeling. In particular, this study makes the following specific contribution to the fields of system modeling and computer vision:

(a) **Crowd anomaly detection**

- Crowd anomaly detection framework, allowing low-latency decision support for the management of large crowds while minimizing the need for computational and human monitor resources;
- A Navier-Stokes-inspired model suitable for large, dense crowds, reducing the computational complexity of crowd state description to a manageable level, for extracting meaningful parameters that describe the crowd behavior locally;
1. **INTRODUCTION**

- Crowd behavior description with a small set of localized parameters, rendering the anomaly detection problem a dynamic estimation of few model parameters and defining anomalies as abrupt changes in these parameters; and

- Extending the optimal reciprocal collision avoidance method by incorporating crowd density avoidance, group dynamics, and density-speed relationship, generating an efficient microscopic simulation to test and verify crowd models.

**(b) Video labeling**

- General-purpose video labeling framework for efficient generation of high-quality ground truth data;

- Spatio-temporal trajectory recovery from short noisy trajectory segments by exploiting the dynamic cues captured in only sparsely available, short sequences of noise corrupted observations; and

- A software toolbox for cost-efficient, in-house annotation of videos.

1.3 **Thesis Outline**

The rest of this paper is organized as follows. In Chapter 2 we consider the task of identifying anomalies in dense crowds. We start by reviewing the crowd modeling methodology, present our proposed crowd anomaly detection framework, formally define our Navier-Stokes-inspired crowd motion model, and show examples of applied anomaly detection. The supporting Appendix A discusses microscopic crowd simulations and our model extensions that were used to generate crowd state data for testing the anomaly detection. In Chapter 3, we propose a method to accelerate the generation of high quality video labels through the deployment of a flexible annotation framework with computer-aided labeling support that we detail in Chapter 4.
Anomaly Detection in Large, Dense Crowds

2.1 Introduction

In this chapter we consider the task of identifying anomalies in dense crowds. Identifying behavioral anomalies from surveillance videos is a difficult task in any framework. The task is particularly challenging in large crowds because of the complexity that is associated with the tracking of a large number of individuals and prevents analysis in real-time.

However, real-time behavior analysis for dense crowds is important. It is of particular importance in a security context because of the opportunity to cause large damage through malicious actions as we have seen during the Boston Marathon on April 15, 2013.

Background

When entities belonging to the same group act in a coordinated behavior influenced by collective decisions rather than individual goals, those entities exhibit similar behavior and are acting as one crowd. For example, crowd behavior takes place when large groups of people are forced to navigate a given situation, and therefore act differently than they would have as individuals or small groups. A common outcome of this phenomenon is a crowd disaster, which occurs when a high percentage of individuals in a crowd experience real or perceived dangers, such as building fires, overcrowding in sport stadiums, or a bomb threat.

Unfortunately, crowd disasters – in particular crowd stampedes – are becoming more
common, and over the past ten years, more than 3,000 deaths have resulted from these phenomena [1].

Online crowd behavior analysis is crucial for managing ongoing crowd disasters to allow for an informed response of the authorities responsible for ensuring public safety.

Need for autonomous real-time anomaly detection method, suitable for large and dense crowds

A favorable source of observation data for the study of crowds are surveillance videos capturing the crowd’s motion and its behavior. Typically, to observe crowds of people in a specific area of interest, such as train stations and public areas, a closed-circuit television (CCTV) camera network is deployed. The received CCTV video data is then utilized to deduce and monitor the crowd behavior by, for example, the police or a security company’s crowd manager. Of particular interest is the aggregate modeling of crowd behavior which is observable through both the crowd’s motion and its density. However, continuous monitoring by humans is impractical in heavily utilized pedestrian scenes. The combination of sheer numbers, the proximity between people, and partial occlusion, makes tracking and analyzing the motion of each individual in a large inspected area virtually impossible. This is due not only to the possibility of human error, but also to the budget constraints that limit the number of personnel able to monitor CCTV. Because dependency on human observation carries with it so many flaws, the solution to this problem is to employ a computer automated, autonomous, real-time behavioral crowd anomaly detection method. This automated approach tracks individuals in densely occupied areas, with a large number of these individuals moving in multiple directions.

Application examples

An autonomous behavioral crowd anomaly detection, operating in real-time, is the key enabler for online crowd behavior monitoring. To our knowledge, no such system was deployed in real-world situations, so far. Our framework enables real-time decision support for crowd management in large and dense crowds and could be applied in, for example, busy airport and train terminals, or during large political rallies and mass gatherings. Our framework could ensure the

---

1Crowd disaster occur when a high percentage of individuals in a crowd experience real or perceived dangers, such as building fires, overcrowding in sport stadiums, or a bomb threat. For example, in 2003, 100 people died while escaping a building fire in West Warwick, Rhode Island, and in 2005, 1,000 pilgrims died in a mass panic in Baghdad, Iraq.

2In 1989, 96 spectators died in a football stadium in Sheffield, England after uninformed police actions exacerbated overcrowding in a narrow tunnel leading to the seating area. In a similarly caused stampede in 2010, 19 teenagers died during an outdoor festival in Duisburg, Germany. It is well known that insufficient knowledge about the crowd condition can lead to a reduced situational awareness and thus uninformed responses. Then, seemingly unimportant decisions, as for example opening or closing exits to moderately crowded areas, can trigger crowd stampedes.
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

safety of spectators visiting sport events or outdoor markets/festivals by preventing catastrophes through early warnings of an evolving crowd disaster and allowing an effective, informed response.

Crowd anomaly research history

Crowd behavior has been empirically studied for about half a century [54]. This analysis established the design of pedestrian facilities, and the need for crowd planning guidelines was addressed. After the mechanisms of normal crowd behavior was comprehended, the research focus shifted to crowd disasters and panic situations. Most research done through socio-psychological investigations, has concentrated on finding the reasons why crowd disasters happen in the first place, by analyzing individual behavior in a situational and social context. With the emergence of computer simulations in the 1970s, it became feasible to study emergency and evacuation situations, and in the last 15 years, researcher groups thus focused on the quantitative understanding of the observed panic phenomena.

Typically, crowd behavioral analysis uses mathematical models to describe the motion of crowds. Generally those methods can be characterized by how crowd motion is measured. Two methods are used: trajectory analysis and motion analysis. Trajectory analysis-based methods are based on tracking single objects. To operate reliably this method requires a low-crowd density to facilitate the tracking of individuals. Motion analysis, which analyzes patterns of crowd motion while ignoring the single individual, can circumvent this problem, and thus is better suited for heavily crowded scenes.

Some approaches for detecting anomalies in crowd behavior rely on the tracking of individuals or virtual particles seeded in the observed area and driven by a measured, spatial flow field. Those methods are typically based on a Boltzmann-like equation and are used in simulation-based, offline crowd analysis. The complexity of these approaches is prohibitive in very large crowds, and prevents their use in online analysis within short latency or even in real-time. However, real-time analysis is essential for timely security response.

Structure of this chapter

Each step is described in further detail as follows: First, we present the details of our proposed crowd anomaly detection framework in Sec. 2.2. In Sec. 2.3 we review the crowd modeling methodology, introduce our Navier-Stokes-inspired crowd motion model, and present examples showing our model’s ability to reproduce well-known crowd self-organization phenomena. In Sec. 2.4,
we introduce an estimation for crowd model parameters, in Sec. 2.5 we detail our framework’s method to detect crowd anomalies. We close this chapter by demonstrating simulated crowd anomalies and their detection using our framework in Sec. 2.6.

2.2 Crowd Anomaly Detection Framework

Addressing the prohibitive complexity of existing crowd anomaly detection methods when applied to large crowds, we propose a layered, multi-scale framework, which uses fluidic models to capture crowd movement and dynamical systems analysis to characterize crowd behavior [55]. This framework also allows a purposeful resource allocation, including the allocation of sensor, computational and human resources, to critical areas where anomalies are suspected.

In order to avoid the complexity of particle-based methods, we make a continuum assumption and model the entire crowd as a continuous fluid. Crowd motion is now modeled using a novel compressible, fluid-dynamics-inspired functions, which includes forces driving purposeful motion, density aversion, multiple desired paths and natural stochasticity. This model is capable of reproducing well-known self-organization phenomena of human crowds. The crowd motion state is then processed to monitor the crowd behavior and to detect deviations from normal behavior.

We characterize crowd behavior in terms of a few slowly varying spatially distributed parameters, which dramatically reduces the complexity of the problem. Dynamic models provide tractable, powerful tools to quickly analyze the spatial and temporal evolution of those parameters and thus for the characterization of behavioral changes. The anomaly detection layer of our framework is based on the characterization of normal behavior from learned features of an identified set of models, and an anomaly is then defined as a discrepancy between these learned characteristics and observed behavior. Specifically, the subsequently presented framework characterizes crowd motion on the sensed data, provides a set of local crowd models, determines parametric values of those local models, learns and adapts a set of auxiliary stochastic models approximating the temporal evolution of the parametric values for each crowd model, and identifies potential anomalies based on both the auxiliary stochastic models as well as the crowd motion model.

Crowd anomaly detection

Figure 2.1 illustrates the crowd anomaly detection process that the system performs for detecting anomalies in crowd behavior. The crowd behavior is evaluated by a succession of increased resolution refinements from crowd through group to the individual levels each triggered by detected
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

![Diagram of crowd anomaly detection framework]

Figure 2.1: Crowd anomaly detection framework.

anomalies in suspect areas, before a human observer is alerted. Advantage of this multi-layered scheme is that the allocation of sensor, computational, and human resources is dynamically optimized. This is done by a set of resolution-dependent models: The crowd level behavior monitoring is executed with coarse grain models requiring only less resources, whereas fine-resolution models with a large computational resource demand are focused on small suspicious areas only.

**Large-scale crowd anomaly analysis**

The large-scale crowd anomaly analysis module receives video data from a network of CCTV cameras and processes the received sensor data at the crowd level to derive the crowd motion state and subsequently the crowd behavior. Therefore a set of low computational cost dynamical models is employed to describe the evolution of crowd motion in both space and time. The crowd is modeled as a multi-phase fluid through a locally parameterized macro-scale, Navier-Stokes-like, stochastic motion model, which incorporates geometry constraints and fundamental crowd behavior characteristics. This allows us to capture the crowd behavior state with a few parameters. Behavior classification is embodied locally by both the slow dynamics of those parameters model parameters and the learned dynamic-rules of their normal spatio-temporal variations. Since crowds in abnormal situations, as for example brawls or stampedes, exhibit motion patterns which deviate significantly
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

from the normally observed behavior, our model parameters will change accordingly. Those spatio-temporal parameters changes identify indicators for detecting anomalies. Thus our motion model parameters, when learned in real-time, can be used to identify local areas where suspicious action takes place. We learn the temporal evolution of the crowd motion model parameters using a set of “stochastic parameter variation models” (subsequently referred to as SPV models) and, finally, detect anomalies by performing a validation/invalidation of those learned SPV models, which discovers parameter changes indicating a potential abnormal behavior.

Meso-scale crowd anomaly analysis

If the present framework identifies a potential anomaly in an area at the large-scale crowd level, the system focuses both additional sensor and computational resources to verify the potential anomaly in the suspect area, and further at the individual scale. The meso-scale anomaly analysis module allocates video sensor resources, and processes resulting received video data at the group level to validate the potential anomaly. Since the number of individuals in a suspect area is small in respect to the whole crowd, exiting agent-based motion models can be used in a manageable computational complexity.

Individual anomaly analysis

After verifying an identified potential anomaly at the group level, the framework further concentrates resources on individuals in the area associated with an identified potential anomaly. The final arbiter who in fact determines the criticality of a flagged anomaly may be a human operator, who will be able to focus only on areas deemed suspect by the multi-layered analysis.

Advantages of proposed framework

The proposed framework enables real-time decision support for crowd management of large crowds while minimizing the need for computational and human monitor resources. The complexity of crowd behavior anomaly detection is reduced to a level manageable for real-time reaction by rendering the anomaly detection problem a dynamic estimation of few model parameters and defining anomalies as abrupt changes in these parameters. In other words, instead of tracking the motion of a large number of individuals, this method requires tracking only a few model parameters. A multi-layered structure confines fine grain analysis, and eventually human monitoring, to small areas, where high-resolution, cost-expensive analysis becomes feasible. The fact that an anomaly
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Figure 2.2: Example of crowded scene applicable to proposed anomaly detection framework.

may include both the deviation from continuously learned patterns, and possibly predefined patterns, prevents the low reliability of methods based solely on a predefined collection of suspect behaviors.

Framework requirements

The above defined crowd anomaly detection is meant for large, dense crowds and is applicable to pedestrian crowds with density of more than 2 people per square meter. Since we rely on knowledge about the crowd motion state, we require sensor input, which can be used to derive the crowd motion state in a short latency. Here, we use video streams provided by a single camera or a network of cameras, monitoring the area of interest. In the case of multiple cameras, an extra processing step needs to be performed, which fuses the crowd motion state together. The camera location(s) and their video resolution must be appropriate to unambiguously derive the crowds motion state, which is constituted by its spatially distributed density and velocity fields. We further assume that the spatial geometry of the observed area, which can be variable and complex, must be known in advance.

Video data appropriate for our framework is not published often in the public domain. This is mainly due to the nature of crowd disasters and the reluctant attitude of authorities to disclose details about those disasters and their causes. However, some surveillance footage of the 2010 “Love Parade” electronic dance music festival in Duisburg, Germany, was made available online by the event organizer and has been our prime use case example to test our anomaly detection framework. Figure 2.2 shows a exemplary video snapshot of the main entry/exit area to the festival’s venue about
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

one hour before the shown area became overcrowded and the tragedy occurred. It is noted that for
the reason of reverence, no surveillance material of the actual stampede was made public.

2.3 Crowd Model

In this section we review the crowd modeling methodology, then we present our Navier-
Stokes-inspired crowd motion model in Sec. 2.3.2. Sec. 2.3.3 concerns the numerical method of
solving the crowd motion model and we present computational details as well as examples showing
our model’s ability to reproduce crowd self-organization phenomena.

2.3.1 Crowd Model Methodology

About 50 years of crowd behavior research resulted in a vast catalog of crowd motion
models. When analyzing crowd motion, the principle differentiation is on what scale crowd is
resolved, and typically, models are classified by their observed resolution as either micro-scale,
meso-scale, or macro-scale.

Micro-scale crowd models consider each individual person in the observed crowd and
create a unique motion representation for that individual based on Newton’s second law of motion.

Meso-scale crowd models lump individuals into larger groups and concentrate on modeling
phenomena on the group level, e.g. spontaneous lane formation for 100 individuals. Those models
either employ micro-scale analysis to evaluate the motion of single individuals, which are representa-
tives of their group [28, 56, 57, 58] or use Boltzmann-like, gas-kinetic models for crowd motion
that do not focus on individuals but rather describe members of a group and their characteristic
motion in aggregate terms (e.g., velocity distributions) [59]. However, this type of models are
typically improved by applying particle discretization methods to numerically solve the gas-kinetic
equations, where each particle represents one individual in the crowd [60]. Another modeling option
on the meso-scale are lattice-gas models for crowd motion, which mimic individuals by gas particles
performing a random walk biased by exogenous forces [61, 62].

The macro-scale analysis of crowds, on the other hand, is neglecting single individuals and
rather concerns large scale phenomena of the entire crowd. For this model class, a crowd motion
representation is derived from observed macro-scale data, which measures the crowd motion state
as a whole. For example, assume that the motion state for a large crowd moving from point $A$ to
point $B$ is known, then this state will not change if a single individual walks in the opposite direction

17
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

from $B$ to $A$. Example macro-scale quantities are average spatial density and velocity. Typically, microscopic crowd representations are based on mathematical equations describing the physical principles of the crowd’s motion, i.e. the spatio-temporal evolution of those collective quantities. The use of a macro-scale crowd models reaches back to the early 1970s, when Henderson published the first fluid-dynamics based crowd models [30, 31] based on vehicular traffic models of the same theories [32].

2.3.2 Continuous Navier-Stokes-inspired Crowd Model

Fluid models provide the framework we use for modeling crowds. Addressing the complexity of Boltzmann-like methods, we model an entire crowd as a continuous fluid under the continuum assumption. This means that even though a pedestrian crowd is composed of individuals, which interact with each other, the crowd is considered to be continuous. As used by other macroscopic models, we make the assumption that many pedestrians exhibit a similar behavior [30]. This means that those individuals share the same properties for at least a short time and over a small area. That is, properties such as density and velocity are considered to be well-defined at infinitesimal small areas. This assumption, even though it is a simplification, is valid under the continuum hypothesis and thus a large group of pedestrians can be treated as one crowd.

Following the continuum assumption, we propose a crowd motion model based on a variant of the Navier-Stokes model for the dynamics of compressible, multi-phase flows. That variant incorporates forces that represent the tendency of individuals to avoid congestion and collision, the pulling effect exerted on an individual by the motion of the immediately surrounding crowd, the tendency of members of each sub-populations to follow the characteristic motion patterns of that sub-population, and stochastic acceleration effects.

Crowd sub-populations

The procedure of crowd level anomaly detection comprising the union of $N$ distinct sub-populations, indexed by $i \in \mathcal{I} \overset{\text{def}}{=} \{1, \ldots, N\}$. Each sub-population is defined by a characteristic statistical description of the pattern of motion of its members, which distinguishes it from any other sub-population. For example, each sub-population may be characterized by the statistical distribution of paths traversed by its members (for example, from a specified train platform to a specified exit gate in a subway station), the velocity of traversing these paths, and the stochastic behavioral pattern (for example, grocery shoppers versus window-shoppers in an open air pedestrian mall).
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Surveilled domain

The procedure is focused on a preassigned grid of points that discretizes the entire surveillance area. That grid is denoted here by $\Omega$ and let $\partial \Omega$ denote its boundary. Detection is based on the real-time analysis of restrictions of sensor data streams to each local neighborhood in an ensemble of local neighborhoods of the points in $\Omega$, denoted $\{\Omega(x)\}_{x \in \Omega}$.

Continuous functions describing crowd motion

Under the continuum assumption we ignore the existence and distinct behavior of the individuals. Instead, we consider characterize crowd mass occupying each local neighborhood by smooth functions of position and time, as described below. This proves to be much simpler than considering the dynamics of the individuals and this is the so-called continuum hypothesis.

An ideal description of the motion of each sub-population is provided by a smooth function of space and time. A generic example is the use of the velocity vector field and the density field, representing the local velocity and density of agents, such as humans, in a crowd per unit area, respectively. For sub-population $i$, $i \in \mathcal{I}$, these fields are denoted by $u_i(x, t)$ and $\rho_i(x, t)$. Here, we denote time by $t$ and a location in space on a flat surface by $x = (x, y)^T$, respectively. Subsequently, without loss of generality, we restrict the analysis to a two-dimensional space, i.e. $x \in \mathbb{R}^2$, and denote a spatial location by $x$. The velocity field is a vector-field with values of the same dimension as the spatial location vector, $x$. That is, if $x \in \mathbb{R}^2$, $u_i(x, t) = (u_i, v_i)^T$ denotes the $i$-th sub-population’s velocity in direction of the $x$- and $y$-axes by $u_i$ and $v_i$, respectively. Since a crowd comprises of a collection of individuals and is therefore inherently discrete, (a) the continuous description may represent a spatially filtered version, as defined by a smoothing filter,

$$\rho_i(x, t) \overset{def}{=} \int G_i(x - \xi) \rho_i^{(d)}(\xi, t) \, d\xi,$$

where the discrete density $\rho_i^{(d)}$ is convolved with a smoothing kernel $G_i$, which is supported over a small area around the spatial origin;\(^3\) and (b) each local neighborhood may contain multiple sub-populations.

\(^3\)Commonly used kernels for smoothing include, polynomials, Epanechnikov, and Gaussian kernels. Here and (unless otherwise noted) throughout this section, we use the Epanechnikov kernel with bandwidth $\omega$: $K(x) \overset{def}{=} \frac{3}{4a} \max \{0, \frac{3}{4a} \left(1 - \|x\|^2/\omega\right)\}$, where the scaling factor $a$ is such that $\int K_\omega(x) \, dx = 1$ holds.
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

**Total density, pressure, total velocity**

The total density is defined as the sum of all the sub-population densities in the local area at a given point in space and time, i.e.

\[ \rho(x, t) \overset{\text{def}}{=} \sum_{i=1}^{N} \rho_i(x, t). \]  \hspace{1cm} (2.1)

An upper bound on the total feasible or tolerable density is denoted \(\rho_{\text{max}}\). \(\rho_{\text{max}}\) may be based, for example, on long-term observations of crowd motion patterns in a particular location, time of day, day of week, season, etc., on safe bounds or areas set by the designer of a public area under inspection, or derived from moving averages over long-time windows. We define the “crowd pressure” or “social pressure” as

\[ p(x, t) \overset{\text{def}}{=} \frac{\rho(x, t)}{\rho_{\text{max}} - \rho(x, t)}, \]  \hspace{1cm} (2.2)

modeling the pressure which individuals in a crowd may feel when they are more tightly or less tightly in the presence of one another, and the tendency of individuals to avoid densely-packed areas. Conversely, individuals exhibit a tendency to move towards less densely-packed areas. The relative sub-population and total densities, denoted \(\rho_{i,\text{norm}} \overset{\text{def}}{=} \frac{\rho_i}{\rho_{\text{max}}}\) and \(\rho_{\text{norm}} \overset{\text{def}}{=} \frac{\rho}{\rho_{\text{max}}}\), respectively, may be introduced to simplify the expression for the pressures, as

\[ p(x, t) \overset{\text{def}}{=} \frac{\rho_{\text{norm}}(x, t)}{1 - \rho_{\text{norm}}(x, t)}. \]

Furthermore, we define a velocity, \(\bar{u}\) on \(\Omega\), which we call “mean velocity” or “observed velocity”, as discussed later, which represents a local average of the velocities of all contributing sub-populations. That is, the observed velocity is determined over a small area and therefore related to the densities and velocities of the sub-populations at a given point in space and time, i.e.

\[ \bar{u}(x, t) \overset{\text{def}}{=} \sum_{i=1}^{N} \frac{\rho_i(x, t)}{\rho(x, t)} u_i(x, t). \]  \hspace{1cm} (2.3)

**Crowd motion state**

In what follows we refer to the collection of continuous functions representing crowd motion as the *crowd motion state* and denoted the state by \(\phi(x, t)\). For example, when the chosen
state comprises the densities and velocities of the sub-populations that together make up the crowd, we use the abbreviation

\[ \phi \overset{\text{def}}{=} [\rho_1, \ldots, \rho_N, u_1, \ldots, u_N] . \]

**Dynamic process for the crowd motion state**

At the heart of the anomaly detection, the large crowd is described by a parameterized set of models, each approximating the dynamic evolution of the crowd model state, over one of the neighborhoods \( \tilde{\Omega}(x), x \in \Omega \). The parameterized dynamic model can be written in the form

\[
\partial_t \phi(x, t) + F(\phi(x, t), \theta) = 0 ,
\]

where \( F \) is a nonlinear operator that may include spatial derivatives of \( \phi \), and where \( \theta \) is a vector of adjustable model parameters. The parameters are used to fit the model to the observed dynamics in each of the local neighborhoods, as discussed later.

**Dynamic process for the crowd mass**

The equations governing the crowd dynamics are inspired by the *Navier-Stokes equations* (NSE) for (compressible) flows, which are derived from the basic principles of conservation of mass, momentum, and energy. Based on the conservation of mass, the so called continuity equation for a single sub-population is given by

\[
\partial_t (\rho_i) + \nabla \cdot (\rho_i u_i) = 0 , \quad i \in I ,
\]

where \( \nabla \) denotes the Nabla operator. Here mass is meant as the area/volume integral of the density, \( \rho_i \). Note that, by definition of the densities \( \rho_i \), the mass at a time \( t \) of sub-population \( i \) in an area \( \tilde{\Omega} \in \Omega \), within the observed area \( \Omega \), is the spatial integral \( \int_{\tilde{\Omega}} \rho(\xi, t) \, d\xi \). The mass thus represents the locally smoothed number of individuals of the \( i \)-th sub-population, that is contained within \( \tilde{\Omega} \). In these terms, the continuity equation states that if, at a certain time \( t_0 \), a portion of the sub-population \( i \) is contained in a certain area \( \tilde{\Omega}(t_0) \), and if one tracks the time evolution of points in \( \tilde{\Omega}(t_0) \), as the points move in space and time according to the velocity field \( u_i(x, t) \), to form an area \( \tilde{\Omega}(t_1) \), at a later time, \( t_1 > t_0 \), then the number of persons of the sub-population \( i \) that occupy \( \tilde{\Omega}(t_1) \) remains equal to the number of those that occupied \( \tilde{\Omega}(t_0) \).

\footnote{Starting with this expression and on, we sometime suppress the dependence of each state entry on the space and time variables. We do so in favor of notational simplicity.}
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Dynamic process for the crowd momentum

The counterpart of the Navier-Stokes momentum equation in \((2.4)\) governs the evolution of the momentum of the mass in an infinitesimal area, and is written in the form

\[
\partial_t (\rho_i \mathbf{u}_i) + C_i - D_i - M_i + P_i - F_i - S_i = 0, \quad i \in \mathcal{I}. \tag{2.5b}
\]

The definition of each of the terms on the left-hand side of \((2.5b)\) is detailed and motivated in what follows.

**Convection term**

The convection term,

\[
C_i \overset{\text{def}}{=} \nabla \cdot (\rho_i \mathbf{u}_i) \otimes \mathbf{u}_i, \tag{2.5c}
\]

models the convective acceleration of the crowd and is a standard term in all NSEs.

**Viscosity term**

Viscosity effects are represented by the terms \(D_i\) and \(M_i\). The first term,

\[
D_i \overset{\text{def}}{=} \frac{1}{\tau^v \rho} \sum_{j \in \mathcal{I}, j \neq i} \rho_i \rho_j \mathbf{u}_j - \rho_j \rho_i \mathbf{u}_i, \tag{2.5d}
\]

models the force exerted by the motion of those immediately surrounding an individual, on that individual. Namely, it represents the difficulty of an individual to move at a velocity and orientation that are different than those of the average motion, around that individual.

The second viscosity term,

\[
M_i \overset{\text{def}}{=} \frac{\rho_i}{\rho} \triangle \mu_i \bar{\mathbf{u}}, \tag{2.5e}
\]

is patterned after the viscous term in the standard Navier-Stokes momentum equation where it is derived from the conservation of momentum due to fluid particles exchange in the direction transverse to the orientation of motion. In our crowd motion model, \(M_i\) models the friction effect of velocity differences in the neighborhood of an individual. In Eq. \((2.5e)\), \(\triangle\) denotes the Laplace operator and the positive scaling constant \(\mu_i\) represents the sub-population’s dynamic viscosity and is a measure of
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

resistance to shearing in the presence of velocities differences. Thus, $M_i$ can be seen as a point-wise momentum transfer in space.

**Pressure term**

The pressure term,

$$P_i \overset{\text{def}}{=} -\frac{\rho_i}{\rho} \beta_i \nabla p_i,$$

(2.5f)

where $\beta_i < 0$ is the scaling pressure coefficient, models the tendency of the moving crowd to prefer motion orientation from high to low density, hence along the negative pressure gradient. In other words, the pressure term represents a congestion repelling force, as individuals in a crowd avoid highly packed areas and instead prefer less crowded areas. In the extreme, this term represents a collision avoidance force. The parameter $\beta_i$ scales this force.

**Extraneous forces and desired path/velocity**

Extraneous forces are acting on the crowd imposed by, e. g., the layout of the inspected area (here, for example, entries and exists to the inspected area), attracting areas (store fronts when observing a human crowd in a shopping mall, etc.) [28]. Concluding, those forces may be modeled by the single term,

$$F_i \overset{\text{def}}{=} \frac{\rho_i}{\tau_i^\circ} (1 + \epsilon^\circ \|u_i^\triangle\|_2^2) u_i^\triangle,$$

(2.5g)

where the vector field $u_i^\triangle$ denotes the deviation from some desired equilibrium motion $u_i^\circ d_i$, $\tau_i^\circ$ is the equilibrium relaxation time, and $\epsilon^\circ$ acts as a modulating nonlinear gain as explained in the following.

The scalar field $u_i^\circ$ is the desired traveling speed of individuals belonging to sub-population $i$. This parameter is depending on many factors, e. g., location, special events, and in case of human crowds also on, for example, culture, traveling intent, time of day, etc. Due to this, the literature reports various fundamental diagrams explaining those dependencies. Fang et al. provides a comprehensive insight in [63]. In general, it is recognized that the traveling speed is depending on the crowd density and an upper velocity limit: $u_\infty$, which is the free-stream, equilibrium velocity. Hence, we write $u_i^\circ = u_i^\circ(x, t, \rho; u_\infty)$.

In our crowd model, we use a linear equilibrium velocity definition

$$u_i^\circ \overset{\text{def}}{=} u_\infty \left(1 - \frac{\rho}{\rho_{\text{max}}} \right),$$

(2.5h)
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

following the empirical observation of an almost linear density-speed relationship [64, 65, 66]. Bellomo reports in [67] that this model does not account for the quality of the environment. However, in cases where (2.5h) is insufficiently fitting observations, higher-order models can be readily substituted.

 Determined by the particular geometry of the inspected area, individuals may have specific motivation to move from their current location to a specific target. Those motion targets are introduced to the crowd model by the vector field $d_i$, modeling both attractive and repulsive effects in the inspected area. Here $d_i$ is a two-dimensional vector field pointing in the direction a crowd will move if not disturbed by other forces. For a sub-population $i$ moving from any $x$ to a target area $Ω_d ∈ Ω$, we derive the desired motion direction, $d_i$, by solving the potential flow equation:

$$0 = \nabla \theta(x),$$

$$\theta(x) = \begin{cases} 
1, & \text{if } x ∈ Ω_d \\
0, & \text{otherwise} 
\end{cases}.$$ 

Now, the motion direction to reach $Ω_d$ in an optimal fashion is found through the relationship $d_i = \frac{∇θ}{∥∇θ∥^2}$.

 Returning to (2.5g), the desired path relaxation time, $τ_i^{\rho}$, is a time constant characterizing the response of a crowd not moving according to their desired/intended path (i. e., direction) and/or velocity. Consequently, the vector field $u_i^{\Delta} \overset{def}{=} u_i^\rho d_i - u_i$ represents the actual deviation from the desired motion $u_i^\rho d_i$.

**Stochastic motion**

 Lastly, we model random variations of the crowd behavior with a Langevin approach by adding the stochastic acceleration term,

$$S_i \overset{def}{=} ρ_i ζ_i,$$ 

where $ζ_i$ is a stochastic vector field with a small density-dependent amplitude $∥ζ_i∥$ and orientation $∠ζ_i$ drawn from a Gaussian distribution with possibly non-zero mean, $μ_ζ$, and standard deviation $σ_ζ ≥ 0$. The term $S_i$ adds small randomness to the motion of individuals in the crowd behavior model. The stochastic acceleration represents an intensifying randomness, to avoid congestion and collision, as an individual approaches a high-density area in a crowd, and reduced randomness during
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

the individual’s motion through low-traffic areas in the crowd. For example, the dependence of the vector-valued stochastic field $\zeta_i$ on the total density $\rho$ may reflect a tendency of individuals in the crowd to increase random sideways motion, to avoid approaching congestion or collision. Similarly, the converse may act to reduce such motion in areas which are less congested or more sparsely populated.

2.3.3 Solution of the Crowd Model

The model of pedestrian crowd dynamics developed in Sec. 2.3.2 has been simulated for a variety of situations is various geometries to demonstrate its abilities. Therefore a computational method based fluid dynamics principles was derived, which accurately solves the crowd motion model equations (2.5) [68]. This method was implemented in the computational fluid dynamics toolbox “Field Operation And Manipulation” (OpenFOAM) [69].

In Sec. 2.3.3.1 we present the numerical method and computational details of solving the crowd motion model. And in Sec. 2.3.3.2 the crowd model’s ability to reproduce self-organization phenomena is demonstrated.

2.3.3.1 Numerical Solution

When solving partial differential equations (PDEs), it is possible to obtain a unique, explicit formula for certain specific examples such as the wave equation. However, most PDEs used in practice such as our crowd model (2.5) have no explicit solution, and can only be solved by numerical methods. Those methods are algorithms that use numerical approximations to solve mathematical problems such as PDEs.

For a continuous problem to be solved numerically, it must be represented by an equivalent discrete problem, sufficiently approximating the continuous problem. Since (2.5) is continuously defined on the entire surveillance area, which is bounded by $\partial \Omega$, $\Omega$ must be discretized. Depending on $\Omega$’s geometry, we choose a finite number of points defined as the knots of either a rectangular or triangular mesh, discretizing the domain $\Omega$.

There are the three classical methods to solve discrete approximations of PDEs: finite difference method (FDM), finite element method (FEM), and finite volume method (FVM). FDM is based on local Taylor expansions to approximate the differential equations using finite differences on a square grid. Since this is a strong limitation of the possible geometries of computational domain,
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

FEM and FVM were developed. They use finite elements and volumes, respectively, to evaluate discrete PDEs.

**Finite volume method**

The discretization of the typical governing equations of continuum mechanics for compressible fluids for a scalar field \( \phi = \phi(x, t) \), in differential form,

\[
\partial_t (\rho \phi) + \nabla \cdot (\rho \mathbf{u} \phi) - \nabla \cdot (\rho \Gamma \phi \nabla \phi) = S_\phi(\phi),
\]

yields the discretized approximation of our crowd model, since (2.5a, 2.5b) can be written in form of (2.6), which is a conservation law for \( \phi \), balancing the temporal derivative, convection term, and diffusion term with the source term \( S_\phi(\phi) \). Since (2.6) includes a second order derivative term, its order of discretization must be at least two to yield appropriate results.

**Discretization of Eq. (2.5)**

In order to preserve a good approximation, the discretization error must be small, which is achieved by posing a smoothness condition on \( \phi \). As mentioned above, we require at least a second-order approximation of (2.6), and thus state that the variation of the transported scalar \( \phi \) can be sufficiently approximated by the truncated Taylor series

\[
\phi(\cdot, t + \Delta t) = \phi(\cdot, t) + \Delta t \, \partial_t \phi(\cdot, t)|_{\phi=\phi(\cdot, t)},
\]

\[
\phi(x, \cdot) = \phi(x', \cdot) + (x - x') \cdot \nabla \phi|_{\phi=\phi(x', \cdot)}.
\]

This means that \( \phi \)'s variations are at most linear in both time and space, which is a safe assumption, since we implicitly used those smoothness properties for the derivation of our motion model.

The finite volume method requires that equations of type (2.6) are satisfied over small cells, or control volumes, \( V_p \), around points \( p \in \Omega \) defined by a given computational mesh. Those volumes can be of any shape – though not all shapes are generating high-accuracy results – and must be both contiguous and completely fill the domain \( \Omega \). Applying the finite volume method to (2.6) leads to the
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

integral equation

\[
\int_{t}^{t+\Delta t} \left[ \partial_t \int_{V_p} \rho \phi \, dV + \int_{V_p} \nabla \cdot (\rho \mathbf{u} \phi) \, dV - \int_{V_p} \nabla \cdot (\rho \Gamma \nabla \phi) \, dV \right] \, dt
\]

\[
= \int_{t}^{t+\Delta t} \left[ \int_{V_p} S_\phi(\phi) \, dV \right] \, dt. \tag{2.9}
\]

The spatial terms of the latter equation are discretized by the application of the divergence theorem, more commonly known as Gauss’s theorem, and its identities

\[
\int_V \nabla \cdot \mathbf{F} \, dV = \oint_{\partial V} \mathbf{F} \cdot \mathbf{n} \, d\mathbf{n},
\]

\[
\int_V \nabla \phi \, dV = \oint_{\partial V} \phi \, d\mathbf{n},
\]

\[
\int_V \nabla \mathbf{F} \, dV = \oint_{\partial V} \mathbf{a} \, d\mathbf{n},
\]

for some generic vector field \( \mathbf{F} \) and scalar field \( \phi \), where \( V \)’s boundary, represented by \( \partial V \), is composed of infinitesimal surface elements, \( d\mathbf{n} \), with outward pointing normals. The discretization yields a set of linear algebraic equations of form

\[
Ax = b, \tag{2.10}
\]

where \( A \) is a square matrix obtained by the discretization, and \( x \) and \( b \) are the principle variables and source term in vectorized form, respectively. Here we refer the reader to Chapter 3 of [69] for a detailed discussion of the underlying mathematical principles used to transform (2.9) into (2.10) and continue our discussion with the specific implications of this discretization scheme.

**Time stepping**

The temporal derivative of (2.6) is solved numerically with an implicit time-stepping using the backward-Euler scheme. The benefit of this method is that larger time steps can be used to integrate the governing equations (2.5a, 2.5b). The time step size is variable and solved for online through thresholding the Courant-Friedrichs-Lewy condition,

\[
C_{\text{max}} \geq C \overset{\text{def}}{=} \max_{i,f} \frac{\rho_{i,f} \mathbf{u}_{i,f} \cdot \mathbf{d}}{\Delta t},
\]

where the mass flux \( \rho_{i,f} \mathbf{u}_{i,f} \cdot \mathbf{d} \) through the cell faces \( f \) with their respective normal vector \( \mathbf{d} \) is used. Then we select the maximal admissible time step \( \Delta t \).
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Spatial discretization

The computational mesh, defining the location of each cell \( V_p \), can be structured or unstructured. The distinguishing feature of structured meshes is that all control volumes are congruent, whereas in unstructured meshes the mesh nodes \( p \) can be arbitrarily placed and thus the cells shapes may vary. Here, we use an former type of grids. The spacing between two mesh points is depending on the expected gradient of the variables \( \phi \) and \( u \). To satisfy the condition stated in (2.7), the grid spacing is small in areas of steep gradient changes. The domain boundary, \( \partial \Omega \), is resolved by the set of cell faces, \( \partial \Omega_{\cdot,j} \), coinciding with \( \partial \Omega \) and is given by the set addition

\[
\partial \Omega \stackrel{\text{def}}{=} \partial \Omega_w \cap \partial \Omega_{\text{in}} \cap \partial \Omega_{\text{out}} ,
\]

using

\[
\partial \Omega_{\cdot,j} \stackrel{\text{def}}{=} \bigcap_{j=1}^{N_w} \partial \Omega_{\text{wall},j} , \quad \partial \Omega_{\text{in},j} \stackrel{\text{def}}{=} \bigcap_{j=1}^{N_{\text{in}}} \partial \Omega_{\text{in},j} , \quad \partial \Omega_{\text{out},j} \stackrel{\text{def}}{=} \bigcap_{j=1}^{N_{\text{out}}} \partial \Omega_{\text{out},j} .
\]

Here we distinguish between wall, inflow, and outflow boundary faces, denoted by \( \partial \Omega_{w,j} \), \( \partial \Omega_{\text{in},j} \), \( \partial \Omega_{\text{out},j} \), respectively. This differentiation is necessary as we apply different conditions to each boundary to accurately model the physical properties of pedestrian crowds.

Boundary conditions

Boundary conditions are required to uniquely solve the mathematical crowd motion model. We therefore define boundary conditions for all principle variables used in (2.5), i.e. \( \rho_i \) and \( u_i \), \( i \in \mathcal{I} \).

There are two kinds of boundary conditions named Dirichlet and Neumann boundary conditions. Conditions of the former class prescribe the principle variable’s value on the boundary, whereas those of the later type define the gradient \( \nabla_n \phi \big|_{x' = \phi(x')} \), \( x' \in \partial \Omega \) in boundary-normal direction \( n \).

For all particle-based systems, walls are considered impermeable. This leads to the following boundary condition for all \( x' \) on \( \partial \Omega_{w,j} \):

\[
\nabla_n \rho_i \big|_{x' = \phi(x')} = 0 , \quad \text{and} \quad u_i = 0 .
\]

Note that the Knudsen correction on wall boundaries [59] is not necessary as we are not concerned with small densities.

Outflow boundary conditions are applied to regions where flow exits the simulated domain and model principle variables which are unknown prior to solving the partial differential equations.
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Since the simulated crowd model is a compressible flow, we cannot solve for the density and instead have to prescribe it on all boundary faces in \( \partial \Omega_{\text{out}} \). In fluid dynamics, the modeled fluid is leaving the computational domain through outflow boundaries. This means that a backflow, i.e. a fluid flux through the boundary inside the domain, is not allowed. However, to model areas of infinite length, we assume a pedestrian backflow is possible. Thus we propose the following mixed outflow boundary condition for \( u_i \). If the flux through an outlet boundary face is positive, i.e. the velocity vector, \( u_i \), points out of the domain, then the boundary condition will be of the Neumann type: \( \nabla_n u_i |_{\phi=\phi(x')} = 0 \). For boundary faces in backflow condition, meaning pedestrians are returning to the observed domain (negative flux), we can imagine that the outflow has to be considered an inflow, and therefore the Dirichlet-type boundary condition, \( \nabla_n u_i |_{\phi=\phi(x')} = \bar{u} \), is applied. In both cases, the density fields are treated with a zero-gradient Neumann condition.

All crowds enter the domain through the inflow boundaries, \( \{ \partial \Omega_{\text{in},j} \}_{j=1,N_{\text{in}}} \). On those cell faces we prescribe Dirichlet conditions for both density and velocity fields. Sub-population \( i \)'s density is evaluated depending on the total mass flux through face \( f \), \( \psi_f = \sum_{i=I} \psi_{i,f} \) as follows:

\[
\begin{cases} 
\bar{\rho}_f + \rho'_f, & \text{if } \psi_f < 0 \\
\rho_{*,f}, & \text{otherwise}
\end{cases}
\]

where \( \bar{\rho}_f \) is a deterministic density value, \( \rho'_f \) represents a stochastic density fluctuation, and \( \rho_{*,f} \) is selected s.t. \( 0 = \nabla_n \rho |_{\rho=\rho(x_f)} \) holds. This essentially models a mixed boundary condition for the density to allow for both inflow as well as outflow. Finally, we close the inflow boundary condition by solving (2.5h) for all cell faces \( f \) given the now known density values.

Initial conditions

Initial conditions for all principle variables are required to solve for a unique solution of (2.5). Here, we simply chose a homogeneously mixed state with a zero velocity component, i.e. \( u_i = \mathbf{0} \) and \( \rho_i = k \rho_{\text{max}} \forall i \), where \( 0 < k < 1 \) is the filling degree.

Admissible densities

We close this section with a short discussion about a limitation of our crowd motion model. Modeling a particle system such as pedestrian crowds using the continuum assumption (see Sec. 2.3) can become inaccurate at very low densities. This is because with decreasing density, the
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Figure 2.3: Knudsen number as a function of normalized density for pedestrian crowds. The dashed line indicates the maximum admissible Knudsen number which allows a particle system to be approximated by continuous, Navier-Stokes-like equations.

The linear transport terms in (2.5) are becoming exceedingly inaccurate. A formal expression of this is expressed with the Knudsen number,

\[ K \text{ def } = \frac{\lambda}{L}. \tag{2.12} \]

where \( \lambda \) is the mean free path between collisions within the particle system, and \( L \) is the characteristic length scale in the system. \(^5\)

Systems with \( K \) near or greater than 0.2 \(^70\) should not be modeled by PDFs of type (2.5) and instead statistical methods such as particle simulations must be used. For pedestrian crowds, kinetic theory yields a mean free path between collisions of \( \lambda = 1/\sqrt{4.2\pi\rho d} \) \(^30\). Here, \( d \) is the diameter of a pedestrian modeled as disk. It is well known that this quantity is depending on the pedestrian’s velocity and thus \( d = d(|u|) \). Following the data reported in \(^71\), the pedestrian extent may be modeled by a rectangle with width \( d_1 = d_{10} \frac{|u|}{\sqrt{\infty}} \) and length \( d_2 = d_{20} + 1.06 |u| + (2.05|u| - 0.35) (\frac{|u|}{\sqrt{\infty}})^{10} \), we simplify and state \( d = \frac{1}{2} (d_1 + d_2) \). Considering only crowds in equilibrium state, we use (2.5h) to close the equations and show the resulting \( \rho \)-dependent Knudsen number for pedestrian crowds in Fig. 2.3. To ensure that (2.5) is a valid approximation of pedestrian crowds, i.e. \( K < 0.2 \), we require a normalized total density s.t. \( \rho_{\text{norm}} > 0.02 \).

\(^5\)The characteristic length scale in a system is not uniquely defined. Here we follow \(^67\) and chose \( L \) as the largest dimension if the observed domain, \( \Omega \), is bounded. In unbounded systems, \( L \) is selected as the largest dimension of the domain containing the crowd at the initial time, \( t_0 \).
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Table 2.1: Parameters of the crowd motion model (2.5) used in the simulation of the self-organization phenomena for the two sub-populations, $i = 1, 2$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum density $\rho_{\text{max}}$</td>
<td>5 [72, 73]</td>
</tr>
<tr>
<td>Relaxation time of $D_i$ $\tau_v$</td>
<td>3</td>
</tr>
<tr>
<td>Dynamic momentum transfer viscosity $\mu_i$</td>
<td>0.25</td>
</tr>
<tr>
<td>Pressure gradient modifier $\beta_i$</td>
<td>0.75</td>
</tr>
<tr>
<td>Relaxation time of $F_i$ $\tau^{\circ}$</td>
<td>1</td>
</tr>
<tr>
<td>Nonlinear gain of $F_i$ $\epsilon$</td>
<td>0.5</td>
</tr>
<tr>
<td>Intended free-stream velocity $u_{\infty}$</td>
<td>1.34 [31]</td>
</tr>
</tbody>
</table>

2.3.3.2 Self-Organization Phenomena

The model of pedestrian crowd dynamics developed in Sec. 2.3.2 has been simulated for a variety of situations is various geometries. Despite its simplicity, our crowd model can reproduce various self-organization phenomena observed in human crowds. Two examples of those spatio-temporal behavior patterns are presented in the following. It is noted that those patterns are a result of intrinsic collective phenomena of the whole pedestrian crowd and not motivated by external influences.

Self-organization phenomena simulation

We model two sub-populations, $i = 1, 2$, both sharing the same properties as detailed in Table 2.1. The subsequently presented crowd motion state fields, $\rho_i$ and $u_i$, are normalized by $\rho_{\text{max}}$ and $u_{\infty}$, respectively.

Lane formation

Empirical observation

Pedestrians walking with the same desired walking direction prefer to walk in lanes and separate themselves from other pedestrian walking in an opposite direction. In boundless spaces, both the structure and location of those lanes are varying in time. It was observed that the number of lanes depends on the width of the street, crowd density, as well as the degree of present stochastic motion. Helbing interprets the lane formation phenomena with the assumption of humans not preferring a special side when avoiding obstacles [28] whereas [74] observes in experiments that the left side is preferred in Japan.
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Figure 2.4: Temporal development of lane formation in a bidirectional crowd in a straight corridor at $t = 15, 50, 100, 200, 400$ (top to bottom) displayed with $\rho_1 u_1 + \rho_2 u_2$ (left) and $\rho_2 - \rho_1$ (right).

Numeric simulation results

Here, the observed domain $\Omega$ is a segment of an infinite-length, rectangular corridor of width $w$ and length $l$, which is bounded by impenetrable walls on the top and bottom. The left and right boundary are the inflow and outflow areas for two sub-populations, i.e., in this scenario we write $I = \{1, 2\}$. Sub-populations $i = 1, 2$ have a desired path from right to left and vice versa.

Integrating the crowd motion model on $\Omega$ shows that our model (2.5) reproduces lane formation for all but low densities ($\rho_{\text{in}} \overset{\text{def}}{=} \rho(x) \leq 0.1, x \in \partial \Omega_{\text{in}}$). Even though [74] reports that in experiments lanes emerge already at low densities, this result was expected as the model in Section 2.3.2 is not designed to handle low densities. Interestingly, Helbing’s social force model [28] as well as our crowd motion model exhibits the same property and instead both models converge to a non-homogeneous mixing state.

With increasing density, the mixing state gets unstable and lanes start to form. At medium inflow densities, $\rho_{\text{in}} > 0.3$, the segregation by walking direction is very pronounced. Fig. 2.4 shows the formation of two strongly separated lanes for an inflow density $\rho_i = 0.6, i = 1, 2$. 

32
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Figure 2.5: Multiple lanes evolve in a bidirectional crowd constrained by a corridor with width \( w = 10 \) (top) and \( w = 25 \) (bottom). The asymmetry of the lanes is attributed to the influence of the stochastic term, \( S_i \), and the discontinuity of the lanes near the corridor walls are, in fact, the result of our boundary conditions. The data is presented as in Fig. 2.4.

Number of lanes

Helbing reports in [28] that the number of lanes, \( N_L \), emerging in his microscopic, social force model scale linearly with the corridor’s width, \( w \), following

\[
N_L(w) = 0.36w^{-1} + 0.59.
\]  
(2.13)

Using our crowd motion model, we observe a similar lane width scaling in respect to \( w \); two examples are shown in Fig. 2.5 for \( w = 10 \) (top row) and \( w = 25 \) (bottom row). In both cases the formation of numerous lanes is observed.

Vortex formation

Empirical observation

At intersections of pedestrian paths, the motion pattern is changing often. Assuming a two-way intersection in a shape of a cross, there may be times at which the intersection is crossed only crossed in one direction. During those times, the motion pattern is determined by the dominantly used travel direction. But there might be also phases at which all paths are heavily utilized. Then it can be observed that the crowd behavior is exceedingly stronger organized and a vortex motion pattern occurs in the middle of the intersection.
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Figure 2.6: Temporal evolution of vortex formation on a four-way pedestrian intersection at $t = 5, 10, 25, 50$ displayed with $\sum_{i \in I} \rho_i u_i$. The color-coded arrows represent the velocity field of the four sub-populations.

It is well known that the likelihood of vortex forming is increased by adding an obstacle in the intersection’s center. Pedestrian intersections with those improvements exhibit a traffic pattern, which is similar to road traffic ring junctions. Even though the motion path in this vortex is longer than the direct path across the intersection, the collective motion becomes more efficient, which stabilizes this phenomena. When entering the intersection, pedestrians only need to find a gap in the pedestrian flow on the intersection, they do not always perform a full stop and their deceleration is minimized. As a result, by keeping a part of their momentum, the average speed and thus flow through the intersection will increase.
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Numeric simulation results

In this case, the observed domain $\Omega$ is a cross-shaped two-way intersection. Each feeding corridor has a width of $w = 5$ and length $l = 5$. The ends of the corridor branches are the in- and outflow areas for 4 sub-populations, which want to cross the intersection: sub-population 1 (3) from right (left) to left (right) and sub-population 2 (4) from top (bottom) to bottom (top). All other boundaries are consisting of impenetrable walls.

Fig. 2.6 shows the temporal evolution of the vortex formation phenomena.

2.4 Crowd Model Parameter Estimation

The crowd motion model, presented in Sec. 2.3.2, above, includes a set of parameters quantifying the effect of crowd viscosity, crowd pressure, extraneous forces, and random motion that characterize local motion of each sub-population. The deterministic part of the crowd motion model for a single sub-population $i$ is parameterized by six variables, which are accumulated in the vector

$$\theta_i \overset{\text{def}}{=} (\tau^v_i, \mu_i, \beta_i, \tau^o_i, \epsilon^o_i, u_\infty)^T,$$

and we further write $\theta \overset{\text{def}}{=} \{\theta_i\}_{i \in I}$ as the aggregated parameter vector for all $N$ sub-populations.

It is emphasized that (2.14) does not comprise the model parameters representing target behavior, $d_i$. Target behavior may be primarily based on long-time observations of the crowd and such $d_i$ is not a parameter of interest to describe the short-term temporal evolution of crowd behavior. Long-term observations can be gathered from sensor data, such as video data, of an area under surveillance for a long time. Example time intervals can include minutes, hours, days, weeks, months, seasons, or even years and depend on the surveillance purpose. Accordingly, the parameters representing target behavior may depend on the location relative to the global spatial geometry, or on time of day, day of week, season, etc. These target behavior parameters may also include a functional dependence on the state of the flow, representing long-time averaged shifts in sub-population path, in response to varying levels of traffic density of the crowd. We further note, that this argument also holds for the maximum admissible velocity amplitude, $u_\infty$, which could thus also be excluded.

---

6 As an illustration, the first- and second-order statistics of the routes and velocities of the crowd moving from one particular subway platform to another during rush-hour on a working week day in autumn, could be determined from observation over an entire month or two, and be expected to remain valid under similar conditions.
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

from all $\theta_i$ vectors. In other variants, heavily penalized small real-time adaptations of these a priori estimates can be included in $\theta_i$.

Parameters estimation

During normal crowd behavior, it is safe to assume that the crowd motion, characterized by the $N$ pairs $(u_i, \rho_i)$, is essentially band limited. We interpret this assumption in terms of a time scale $\tau \gg \frac{L}{u_{\infty}}$, (where $L$ is the characteristic length scale of the system) and the following smooth dependencies:

$$\theta = \theta(x, \frac{t}{\tau}).$$

(2.15)

By this assumption, the parameter vector remains essentially constant over a short time window around $t$.

As described earlier, the crowd anomaly detection framework characterizes motion in each local area by determining a set of real-time estimates of distributed sub-population motion states, $(u_i, \rho_i)$, found in the local area. To determine the density field, real-time density estimates may be directly computed from local data only for the total density $\rho$. To determine the velocity field, a locally averaged velocity of the combined sub-population can be determined first, i.e., $\bar{u}$. This might be accomplished by, for example, using optical flow field estimates.

After determining the total density and locally averaged velocity, the present framework can then determine estimates for velocities and densities for each sub-population based on the distributed estimates of $\rho$ and $\bar{u}$, over small areas such as the local areas partitioned by the spatial grid used to discretize $\Omega$. For example, under the assumption (2.15), local variations in $\rho_i(x, t)$ and $u_i(x, t)$ can be described by low-order polynomial representations. The present system can estimate coefficients of the polynomials from direct estimates of $\rho_i$ and $u_i$ at multiple grid points in space, and several successive samples in time, such as consecutive frames of a video stream at small neighborhood of all tuples $(x, t)$, with $x \in \Omega$ and to bounded by some times $t_0$ and $t_1$, $t_0 < t_1$.

In particular, the crowd model parameter estimation should perform a robust estimation of the parameter vector, $\theta$, as (2.5) is stochastically excited through the terms $\{S_i\}_{i \in \mathcal{I}}$. This step of the anomaly detection process is taking the real-time estimated crowd motion state (i.e., densities $\rho_i$ and velocity fields $u_i$) on $\Omega$ as an input and returns $\theta$. 
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Parameters estimation of distributed variables

Estimation methods that do not rely on an integration of the crowd motion model (2.5a, 2.5b) are commonly based on solving a minimization problem, employing the residual error of said equations evaluated at the measured states, i.e.

\[ \mathcal{R}_i(x, t; \theta_i) \overset{\text{def}}{=} \partial_t \rho_i(x, t) - \mathcal{F}(\rho_i(x, t), u_i(x, t); \theta_i), \quad (2.16) \]

as a cost function. The parameter estimation conducted over a short-time horizon of length \(2T\), i.e. \([t - T, t + T]\), and, as discussed earlier, in local neighborhoods, \(\tilde{\Omega}(x_j)\) around points \(x_j\), can then be expressed by the general form

\[
\theta_i(x_j, t) = \arg \min_{\theta_i} \int_{\tilde{\Omega}} \int_{-T}^{T} \mathcal{R}_i(x, t - \tau; \theta_i)^2 d\tau \, dx \\
+ \lambda_t \| \partial_t \theta_i(x, t) \|^2 + \lambda_x \| \partial_x \theta_i(x, t) \|^2,
\]

where \(\| \cdot \|\) and \(\partial_x\) denote some suitable norm and the partial derivative with respect to \(x\), respectively.

A regularization to enforce smooth estimated values, \(\theta_i\), in time and space is introduced to (2.17) through the second and third term, respectively, and controlled by the constants \(\lambda_t, \lambda_x \geq 0\). \(^7\)

In case it is viable to integrate the crowd model, estimation methods may take the model’s solution into account and estimate \(\theta\) by minimization of regularized least squares. For example,

\[
\theta_i(x_j, t) = \arg \min_{\theta_i} \int_{\tilde{\Omega}} [\tilde{\rho}_i - \rho_i(\cdot; \theta_i)]^2 + \lambda_u [\tilde{u}_i - u_i(\cdot; \theta_i)]^2 dx \\
+ \lambda_x \mathcal{R}_i(\cdot; \theta_i)^2, \quad \lambda_u, \lambda_x \geq 0 .
\]

Here we assume that the state variables, \(\phi\), are corrupted by an independent and identically distributed measurement error, \(\epsilon\), and only \(\tilde{\phi} = \phi + \epsilon\) can be measured. The error is assumed to be drawn from a zero-mean Gaussian distribution with some variance \(\sigma^2_\epsilon\): \(\epsilon \sim N(0, \sigma^2_\epsilon)\). Since sensor locations and their derived variables are intrinsically discrete, we can rewrite the two terms in the integral expression

\(^7\)Any penalization through \(\lambda_t, \lambda_x\) should respect the expected scale of \(\theta_i\)’s entries such that all entries of \(\theta_i\) have the same “influence” on the penalization.
of (2.18) as discrete sums over all sensors \( s = 1, 2, \ldots \) at their discrete locations \( x_s \in \tilde{\Omega}(x_j), \) i.e.
\[
\theta_i(x_j, t) = \arg\min_{\theta_i} \sum_s [\hat{\rho}_i(x_s, t) - \rho_i(x_s, t; \theta_i)]^2 \\
+ \lambda_u \sum_s [\hat{u}_i(x_s, t) - u_i(x_s, t; \theta_i)]^2 \\
+ \int_{\tilde{\Omega}} \lambda_x [R_i(x, t; \theta_i)]^2 dx, \quad \lambda_u, \lambda_x \geq 0.
\] (2.19)

Suitable estimation methods to solve the optimization (2.17)-(2.19) may be based on classical estimation methods (e.g., minimum variance unbiased estimation, maximum likelihood estimation, least squares estimation) [75], Bayesian estimation (e.g., minimum mean-square estimation, maximum a posteriori estimation, optimal filtering, wiener filtering, Kalman filtering) [76]. Since we target a low-latency framework for anomaly detection, we employ the simplest parameter estimation methods, sacrificing estimation quality for computational speed, that still yield suitable results.

In the following, we consider the momentum equation (2.5b) as our crowd motion model with unknown stochastic terms, \( S_i \). We simplify the equilibrium motion term (2.5g) by setting the nonlinear gain \( \epsilon^o \) to zero and employ the linear velocity-density relationship (2.5h), which yields
\[
F_{i*}^a \overset{\text{def}}{=} \frac{u^\infty}{\tau_i^o} (1 - \rho_{\text{norm}}) \rho_i d_i - \frac{1}{\tau_i^o} \rho_i u_i.
\] (2.20)

This allows us to write the residual (2.16) as
\[
R_i(x, t; \theta_i) \overset{\text{def}}{=} T_i - C_i + D_i + M_i + P_i - F_{i,a}^* + F_{i,b}^* + S_i,
\] (2.21)
where \( T_i \overset{\text{def}}{=} \partial_t (\rho_i u_i) \) denotes the temporal derivate term, or expressed in matrix-vector form,
\[
R_i(x, t; \theta_i) \overset{\text{def}}{=} A_i(\rho_i, u_i) \hat{\theta}_i - b_i(\rho_i, u_i),
\] (2.22a)
\[
A_i \overset{\text{def}}{=} \begin{bmatrix} d_i & m_i & p_i & -f_{i,a}^* & -f_{i,b}^* \end{bmatrix},
\] (2.22b)
\[
b_i \overset{\text{def}}{=} T_i - C_i + S_i
\] (2.22c)
\[
\hat{\theta}_i \overset{\text{def}}{=} \left( \begin{array}{c} \frac{1}{\tau_i^o} \mu_i, \beta_i, \frac{u^\infty}{\tau_i^o}, \frac{1}{\tau_i^o} \end{array} \right)^T.
\] (2.22d)
introducing the modified parameter vector $\hat{\theta}_i$. Here, given $N_x$ discrete points in $\Omega$, $b_i \in \mathbb{R}^{2N_x}$ is the sum of all terms independent of $\hat{\theta}_i$ and the $2N_x$-length columns of matrix $A_i$ hold all terms of (2.21) dependent on $\hat{\theta}_i$, written in vectorized form. For example, $A_i$’s rows $j$ and $j + 1$ are defined by their entries

$$
\begin{pmatrix}
\frac{1}{\rho} \sum_{j \in I, j \neq i} \rho_j (\rho_j u_j) + \rho_j (\rho_i u_i) \\
\frac{\rho_i}{\rho} \bar{u} \\
-\frac{\rho_i}{\rho} \nabla p_i \\
(1 - \rho_{i,\text{norm}}) d_i \\
\rho_i u_i
\end{pmatrix}^T,
$$

where all spatially distributed variables are evaluated at $x = x_j$.

The classical least squares estimation can be used to solve (2.17) using the residual as defined in (2.22) as follows. Considering the unregularized estimation, i.e. $\lambda_t = \lambda_x = 0$, we substitute (2.22a) into (2.17) for all discrete times $t = [-T, T]$ with a time step of $\Delta t$, and solve the quadratic minimization problem

$$
\hat{\theta}_i^* = \arg \min_{\hat{\theta}_i} \| A_i \hat{\theta}_i - b_i \|^2, \tag{2.23a}
$$

$$
A_i \overset{def}{=} \begin{bmatrix}
A_i(t - T) \\
A_i(t - T + \Delta t) \\
\vdots \\
A_i(t + T - \Delta t) \\
A_i(t + T)
\end{bmatrix}, \quad b_i \overset{def}{=} \begin{bmatrix}
b_i(t - T) \\
b_i(t - T + \Delta t) \\
\vdots \\
b_i(t + T - \Delta t) \\
b_i(t + T)
\end{bmatrix}, \tag{2.23b}
$$

in the least squares sense by solving the normal equations

$$
(A_i^T A_i) \hat{\theta}_i^* = A_i^T b_i. \tag{2.24}
$$

Back-substitution of $\hat{\theta}_i$ into (2.22d) yields the desired parameter vector $\theta_i$. The parameters of all $N$
sub-populations result from evaluating (2.24) for all $i \in I$ or by concurrently estimating all $\hat{\theta}_i$:

$$
\hat{\theta}^* = \arg \min_{\hat{\theta}} \| A\hat{\theta} - b \|^2,
$$

$$
A \defeq \begin{bmatrix} A_1 \\ \cdots \\ A_N \end{bmatrix}, \quad b \defeq \begin{bmatrix} b_1 \\ \cdots \\ b_N \end{bmatrix}.
$$

Finally, we note that (2.22b) implicitly requires that the discrete points $x_j$ are equally spaced in $\tilde{\Omega}$. If this requirement does not hold, (2.22b) can be improved by weighting the $j$-th right-hand side row with a factor $a_j / \sum_j a_j$, where $a_j$ is the area of the $j$-th patch at location $x_j$.

**Noise filtering**

The solution of the crowd model parameter estimation, such as (2.24), involves both temporal and spatial derivatives of the motion states introduced in (2.22a). In the practical case of noisy state measurements, those derivatives amplify the noise, which is not captured by the motion model, and this would result in erroneous parameters. In order to increase the estimation robustness against noise, methods taking into account the noisy nature of the input data are the most appropriate techniques to conduct the parameter estimation. Here, we assume again that zero-mean noise, $\epsilon$, is corrupting the state variables,

$$
\tilde{\phi} = \hat{\phi} + \epsilon, \quad (2.25)
$$

and approximate the underlying, real state variables by a linear combination of $N$ basis functions:

$$
\hat{\phi}(x, t) \defeq \sum_{j,k=1}^{N} \alpha_{j,k}(t)b_j(x)b_k(x) = b^T(x)\alpha b(x), \quad (2.26)
$$

where $b = (b_1, \ldots, b_N)^T$, and $\alpha$ is the matrix of basis coefficients with entries $\alpha = \alpha_{i,j}$. Since we require that the state $\phi$ is a sufficiently smooth function on $\Omega$, we employ B-splines of degree $N$ as the basis functions, $\{b_j\}_{j=1}^N$, in (2.26) [77]. Since some terms of (2.5b) comprise of first and second derivatives in space, we choose cubic splines ($N = 3$) to achieve a smooth approximation of those terms by (2.26). Another benefit of using B-splines as the basis functions is their compact support, i.e. the functions are non-zero only on a small compact subset, which makes for an efficient use.

---

8For steady-state problems, the temporal derivative, $t_i$, evaluates to zero. However, transient problems have to include this term in (2.24).
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

An approximating spline interpolation may be used to smooth the state measurements. To match the added noise assumption in (2.25), we do not require that the approximation matches the measurements in $x_j$, i.e. $0 = \Delta \phi(x_j, t) \overset{def}{=} \| \phi(x_j, t) - \hat{\phi}(x_j, t) \|$ might not hold for all $x_j$. In this case it is useful to define a measure of the approximation accuracy to steer the expected values of $\Delta \phi(x, t)$. For this purpose, we define the cost function consisting of squared weighted residuals

$$S(\alpha) \overset{def}{=} \sum_{j=1}^{N_x} w_j^2 \left( \phi(x_j) - \hat{\phi}; \alpha(x_j) \right)^2,$$

where the weights $w_j$ reflect prior knowledge about the measurement accuracy at location $x_j$. A least square optimal spline approximation, $\hat{\phi}$, is found by minimizing the $S(\alpha)$ in respect its coefficients $\alpha$. Additional penalty functions that aim to impose smoothness to avoid overfitting may be added if needed. Furthermore, another penalization term might be added to incorporate the crowd motion model structure as shown in [77]. The resulting $\hat{\phi}$ is then used to find the crowd model parameters by an optimization such as given by (2.24).
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Crowd Model Parameter Estimation From Artificial, Noisy Data Generated by the Examples

This section concerns a proof of concept showing that the estimation method presented above is suitable for application to distributed parameters as, for example, used in (2.5).

The first parameter study shows that a relatively sparse temporal discretization is sufficient to resolve the temporal effect of the motion model states, \( \phi \).

We sampled our simulated two sub-population, bidirectional corridor crowd as presented in Sec. 2.3.3.2 by applying (2.25) to the simulated motion states with both 100 and 10Hz sampling frequency. The added noise field \( \epsilon \) was generated by a random zero-mean Gaussian process with variable standard deviation \( \sigma_\epsilon = 0.1 \| \phi \|_2 \). The parameter estimation was then conducted by employing (2.24) at each discrete time \( t_k \) over a short-time horizon of length \( 2T = 10 \).

Figures 2.7 and 2.8 display four example parameters for the 100 and 10Hz sample frequency. As expected, it is apparent that the higher sampling rate results in both a smaller standard deviation if the estimation error, \( \hat{\theta} - \theta \), and a better resolved transient. However, Figure 2.8 leads to the conclusion that a measurement sampling rate of 10Hz yields a sufficient estimation quality.

Fig. 2.9 shows parameter fields estimated from a not fully converged crowd motion state at \( t_k = 5 \). The estimated parameters are close to their true values (green color code) and the largest
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Figure 2.9: Estimated distributed parameters of the crowd motion model for the two-lane corridor configuration. From top to bottom: $\beta_i, \tau^0_i, \mu_i, u_{\infty_i}, \tau^0$ for sub-population $i = 1, 2$ shown in the left and right column, respectively. The bottom line shows the corresponding residual error $R_i(x, t; \theta_i)$. The color encodes the parameter values, where green represents estimates in accordance to the true value.

The color encodes the parameter values, where green represents estimates in accordance to the true value.

deviation is found near the wall boundaries, $\partial \Omega_w$. This is not surprising as the forces within the crowd are not yet fully balanced and transient effects due to the boundaries are not resolved by the motion model (2.21) used in the estimation.

2.5 Crowd Behavior and Anomaly Detection

2.5.1 Crowd Behavior Model

This step of the crowd anomaly detection process learns both the typical short-time dynamics and statistical moments of the crowd model parameters accumulated in the vector $\theta$. Learning methods may model the temporal evolution of $\theta$ over a short time by employing a second set auxiliary models, which we label *stochastic parameter variation models*. The stochastic parameter variation models, denoted by the tuple $(F, \gamma)$, are suitable low-dimensional functions of form,

$$\frac{d}{dt} \theta = F(\theta; \gamma),$$
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

where \( F \) is a possibly nonlinear, vector-valued function of both \( \theta \) and a second set of parameters, \( \gamma \), of appropriate dimension, \( M \).

The stochastic parameter variation model will be continuously updated as in any control and estimation paradigm. Albeit, adaptive model updates will take place at a time constant that is considerably longer (for example, an order of magnitude) than those of the crowd motion state estimation. Accordingly, the required predictive power of the model should be only over a comparable interval in the future. Moreover, even the simplest example of a linear formulation corrected with an additive weighted error term, \( e \), of form

\[
\frac{d}{dt} \theta = A(\gamma) \theta + B(\gamma) e
\]

may well suffice.

This learning process involves the learning of both the stochastic parameter variation model’s structure, \( F \), and its parameters, \( \gamma \), respectively. A general formulation for a robust identification of \((F, \gamma)\) is given by

\[
\gamma \overset{def}{=} \arg \min_{\gamma} \left\| \frac{d}{dt} \theta - F(\theta; \gamma) \right\|^2 + \lambda_\gamma \| \gamma \|^2 + \lambda_M f(M), \tag{2.28}
\]

where \( f \) is any suitable function of \( M \),\(^9\) and \( \lambda_\gamma, \lambda_M \geq 0 \) are possible penalization parameters to ensure smoothness of \( \gamma \) and to control the complexity of \( F \), respectively.

The stochastic parameter variation model, \((F, \gamma)\), is learned during an initial training period, and continually updated during system operation. Examples for the learning may be based on: classical nonlinear system identification methods [78, 79], principal components analysis (PCA) [80], nonlinear discriminant analysis (LDA) [81], subspace learning (e.g., local linear embedding) [82], Galerkin methods [83, 84, 85], or machine learning algorithms (for example, support vector machines, Bayesian networks, evolutionary/genetic algorithms [86], neural networks) [87, 88, 89].

2.5.2 Crowd Anomaly Definition

The task of anomaly detection, independent of its application to crowds, requires the challenging and critical definition of an anomaly based on assumptions that may vary with the task’s purpose and/or its application [36]. Those assumptions affect all aspects of the task, such as the choice of sensors, the measurements, feature extraction, modeling as well as the definition of an anomaly itself.

In general and by definition, anomalies are events that are important to a given task, they are abnormal and thus have a low-probability of occurrence. This allows the typical formulation of anomaly detection as an outlier detection problem. Given some measurements and an associated

\(^9\)For example, \( f(M) = \log(M) \).
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

model explaining the measurements in a normal condition, anomalies are defined as all measurements that cannot be explained by the “normal condition” model. For example, a statistical model may assign probabilities to a range of measurements and any measurement with a probability below some threshold is considered an anomaly. In order for those approaches to operate satisfactorily, anomalous events must have a signature (in measurement space) that distinguishes them from all normal events. As said before, the signature differences may occur in space and/or time and they may vary in space and/or time as well. For example, a crowd moving into a stadium will behave differently from a crowd exiting and understandably the anomaly signature will vary too.

Some methods for monitoring crowd behavior include detecting anomalies in a crowd, such as individuals avoiding a certain area or people changing directions sharply, or even stampeding in response to a real or perceived danger. Some former approaches to tracking crowds rely on a library of normal behavioral patterns. The problem with this is that the formal characterization of what is normal is subject to personal bias and if the definition of abnormal events is a-priori unknown a complete cataloging of benign behavior can not be compiled.

Here, we consider the task of anomaly detection from localized data that is gathered from features describing the crowd motion state. We derive the crowd state through measurements recorded with static CCTV cameras. However, other types of sensors, including multiple spatially distributed sensing devices, may be used here, such as infrared video cameras, radar systems, photoelectric sensors, pressure-sensitive sensors, wireless non-contact systems, acoustic sensors such as microphones, optical sensors, laser-based systems, or systems employing localization of mobile phones, or wireless non-contact systems [55].

2.5.3 Crowd Anomaly Detection

The final step of the anomaly detection in crowds is to decide whether or not the crowd in observed in \( \Omega \) is behaving deviant from the “typical” behavioral pattern. Therefore we define a mapping from both the augmented state variables collected over a short-time history and the stochastic parameter variation model, \( \Gamma = \{ u, \rho, \theta, F, \gamma \} \), to the anomaly detection score \( a = a(x_j), x_j \in \Omega \), reflecting to which degree the crowd around location \( x_j \) is considered to be anomalous. If \( a \) is a real number scalar field with \( 0 \leq a(x_j) \leq 1 \), then \( a \) is the probability of the event “an anomaly is occurring in the neighborhood of \( x \)”, otherwise \( a(x_j) \in \{0, 1\} \) is considered a binary label, where the values 0 and 1 represent the states “no anomaly detected” and “anomaly detected”, respectively.

All applicable methods to find the map from a short-time history of \( \Gamma \) to \( a \) are categorized
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Figure 2.10: Motion of a bidirectional, two-population crowd under external forcing in the corridor setup, as described in the text, is displayed. The top row shows the crowd’s reorganization reaction shortly after \((t_0 + 1, \text{left})\) and at the end \((t_1, \text{right})\) of the external forcing. The bottom image depicts the crowd state 2 seconds after the forcing stopped \((t_1 + 2)\). Here, we display \(\Delta \rho = \rho_2 - \rho_1\) for a better visibility of the disturbance caused by the external force field. The colors red over green to blue encode \(\Delta \rho > 0, \Delta \rho = 0, \Delta \rho < 0\), respectively. The velocity of the sub-population moving to the right, \(u_1\), is visualized with a line integral convolution (LIC) vector field visualization [90].

as follows: Classical cost function-based methods could employ first- and/or higher-order statistics of the augmented state \(\Gamma\). If any statistical moment or their combination exceeds a predefined threshold, an anomaly is considered to be detected. A different approach is based on the analysis of the dynamics of the stochastic parameter variation models described by the pair \((\mathbf{F}, \gamma)\). An anomaly is indicated if the dynamics, described by the model structure \((\mathbf{F})\) and its variables \((\gamma)\), are changing significantly. Deterministic prediction through precursors learned from previously detected anomalies are the third class of mapping functions presented here. Here we assume that a specific, short-time pattern of observations, \(\Gamma_a\), which is known to typically represent and/or precede an anomaly can be described by a set of variables, \(V_\Gamma\), around \(\Gamma_a\). This means, an anomaly state is triggered if the real-time system state \(\Gamma\) is element of any such anomaly-state set, \(V_\Gamma\). Methods to find a low-dimensional approximation of the anomaly-state set might be appropriate to reduce the computational burden related to this solution method. The latter approach is also suitable to handle predefined heuristic rules describing anomalies, such as heavy motion in a particular direction, strong rotational motion, high crowd density, shock waves, etc. In this case, we would employ the measured motion state variables, \((u, \rho)\), to derive the anomaly indicator function.
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Figure 2.11: Temporal evolution of estimated parameters $\theta_i$ of two sub-populations $i = 1, 2$ (red, blue curves) for a local area in the neighborhood $x = (16, 3)^T$ in reaction to an artificial external body force acting on the crowd at all times in the interval $[t_0, t_1]$.

2.6 Anomaly Detection Results

This section concerns a proof of concept, showing that the crowd motion model parameters of model (2.5) indeed reflect anomalous crowd behavior.

2.6.1 Direct Simulation

Here, we employ the bidirectional corridor crowd simulation introduced in Sec. 2.4 in conjunction with the parameter estimation as described in Sec. 2.4. Starting from a fully converged two sub-population crowd, we simulated an event, which causes the crowd to avoid a certain part of the observed area. An example for such a repelling local event might be a brawl or an obstacle in the path of the crowd.

In a numerical simulation, we model such repelling events with an external body force field, $R_i = \frac{\rho_i}{\rho} F$, which is added to the right-hand side of (2.5b). In the simulation here, the force acts radially on the crowd: Denoting the forcing amplitude and center with $A_F(x, t) > 0$ and $x_F = (x_F, y_F)^T \in \Omega$, respectively, then $F(x, t) = A_F(x, t)(\cos(\alpha), \sin(\alpha))^T$, where $\alpha$ is the angle from $x_F$ to location $x$.

Using this setup, a numerical simulation was conducted as described in Sec. 2.3.3.1 with the parameters presented in Table 2.1. Starting at time $t_0 = 350$, an external force field, $F$, centered at $x_F = (15, 0)^T$ is acting on the crowd. At $t > t_1 = 400$ the external forcing term is set equal to zero, restoring the initial force balance. Note that the forcing term is graded (second order smooth)
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Figure 2.12: Estimated distributed parameters of the crowd motion model for the two-lane corridor configuration under external forcing. It is clearly visible that all parameters exhibit a strong but localized reaction to the external force field. Here we display the parameter fields corresponding to \( t_0 + 20 \) of the sequences presented in Fig. 2.10. From top to bottom: \( \beta_i, \tau_i^0, \mu_i, u_\infty, \tau^0 \) for subpopulation \( i = 1, 2 \) shown in the left and right column, respectively. Since the parameter variations are large in respect to their true values, we display the log-scaled version \( \text{sign}(\theta) \log(|\theta|) \). The color encodes the parameter values, where green represents estimates in accordance to the true value.

in space to not violate the requirements for the numerical solution; for the same reason, we also ramp the forcing amplitude when switching the force on/off.

Fig. 2.10 displays a converged bidirectional corridor crowd flow, organized in two distinct lanes, under external forcing. The corresponding estimated parameters for a small area around the location \( (16, 3)^T \), shown in Fig. 2.11, exhibit an almost immediate reaction to the disturbance. The time delay results from the fact that the disturbance effect (a) is ramped and (b) needs to travel from its source to the sensed location. The estimated parameters exhibit a strong deviation from the observed behavior when the force is activated \( (t_0) \) and deactivated \( (t_1) \), which is due to the fact that the crowd motion state induced by the forcing is not captured by the range of the crowd model. It is noted that after the two displayed transition phases at \( t_0 \) and \( t_1 \), the estimated parameters converge back to their true values.
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

2.6.2 Crowd Simulation

In this section, we present three examples highlighting the performance of our anomaly detection framework. In particular, we present results based on crowd simulations that employ the microscopic algorithm that is presented in Appendix A. The input to our anomaly detection framework is prepared by Eq. (A.9).

**Egress**

Scenario: Unstructured crowd with subsequent egress.

![Figure 2.13: An unstructured crowd that performs an egress. Visualization of the crowd as in Fig. A.2 (left column) and anomaly score (right column).](image)

We model an unstructured crowd in a confined space (e.g. a room) with one sub-population. Each agent is assigned a random target within the room that she/he moves toward with a random preferred speed (c.f. Sec. A.1); after reaching its target, the agent receives a new random target and
so on. After a period of time, the whole crowd performs an egress through an exit located on the space’s boundary. Fig. 2.13 displays three samples of the simulation (left column, top to bottom row): (a) unstructured motion pattern, (b) shortly after the egress begins, and (c) at a later time.

In this scenario, we define anomalies as all behavior that is deviant from the unstructured crowd pattern. Additionally, we impose the constraint that no a priori knowledge of the crowd motion parameters during egress is available. However, since egress behavior is highly structured and such in clear contrast to the “normal” motion pattern, anomalous data is assumed to be represented by a significantly different set of parameters. Those behavioral parameters, $\theta$, are extracted on small 2-by-2 meter neighborhoods centered around the points of a discrete grid that spans the entire simulated space.

The scenario constraints restrict the applicable anomaly detection methods and here we use a one-class classification-based anomaly detection technique that assumes that all training instances have just one class label (here this is “normal” behavior) [91]. In general, this type of approach learns a discriminative boundary containing instances of the single class using classification algorithms such as one-class support vector machines, Gaussian (mixture) models, or one-class kernel Fisher discriminants. A test data sample is considered anomalous if it does not fall within the learned boundary.

A three-layer feed-forward neural network architecture is constructed that has the same number of input and output neurons matching the dimension, $N_n$, of the aggregated parameters vector, $\theta$. This type of network, also known as replicator neural network, serves the function to

---

**Figure 2.14:** Receiver operating characteristic curve for the “egress” scenario.

---

We refer the reader to [92] for a more detailed discussion of one-class classification methods.
reproduce input data at its output with minimal error [93]:

\[ \epsilon_{\text{MSE}}(\theta) \overset{\text{def}}{=} \frac{1}{N} \| F_N(\theta) - \theta \|_2^2, \]

where we denote the network’s input-output transfer function as \( F_N \). Appropriate network performance is achieved through training during which the network is optimized such that the mean square error for all training data is minimized. We follow Hawkins’ network design suggestions, covered in [93], to build and train the neural network.

Assuming that anomalies differ significantly from “normal” data, it is highly likely that the network will not reconstruct samples from unseen motion patterns well and consequently \( \epsilon_{\text{MSE}} \) will be higher for such data. Therefore, the reconstruction error of one sample is the basis for its anomaly score:

\[ a = 1 - \exp(-\kappa \epsilon_{\text{MSE}}(\theta)), \]

where the scaling parameter \( \kappa > 0 \) is selected such that training data is classified as “normal”. Fig. 2.13 (right column) shows the distributed score \( a \), which is close to zero during the unstructured motion pattern (top row) and close to one when the crowd is conducting the egress.\(^{11}\)

To evaluate the anomaly detection performance, we processed multiple “egress” scenario simulations using different initial conditions, egress behavior, and exit locations. With omniscient knowledge, which was not used in the actual detection, we compared the anomaly score classifier with the ground truth and display the receiver operating characteristic (ROC) curve in Fig. 2.14. In conclusion of this discussion, we note that the anomalous egress behavior is also detectable by analyzing the crowd motion state directly, possibly similarly well performing. However, this scenario provides a proof of concept for our claim that the crowd behavior is reflected in the crowd model parameters and that anomalies are indicated by their change.

**External Forcing**

Scenario: A repelling force disturbs a bidirectional corridor crowd.

![Figure 2.15: Average anomaly score calculated over all times the external force is acting on the crowd.](image)

\(^{11}\)Regions with insufficient density are not covered by our method and are assigned a zero anomaly score.
This scenario uses a similar setup as in Sec. 2.6.1, above. A repelling force is acting on a fully converged bidirectional corridor crowd, disturbing its highly structured behavior which in turn alters the crowd motion parameters locally. Those parameters are calculated over small 2-by-2 meter neighborhoods centered around the nodes of a discrete grid that spans the surveilled domain.

Here, we extract the anomaly score with our anomaly framework that is based on the local parameters to as follows. We resort to artificial neural networks to find the anomalously behaving parameters (c.f. Sec. 2.5.1). While there is a wide variety of architectures that could be used to address this problem [87, 88], we selected a feed-forward neural network architecture because of its modeling flexibility.

In particular, the net’s input is the the short parameter sequence $(\theta(t - \tau)^T, \ldots, \theta(t)^T)$, where $\tau = 1$, sampled at 10 Hz; and we define the continuous net output $a \in [0, 1]$ as the anomaly score, where a value of 0 and 1 represent “normal” and “anomalous” behavior, respectively. The network features one hidden layer with a hyperbolic tangent sigmoid activation function and the output layer’s transfer function is a log-sigmoid. Bayesian regularization training is used to prevent overfitting and for its better generalization capability [94, 95].

In a Monte Carlo simulation, we generated 20 simulations with varying initial conditions and different force locations, $x_F$, to test the versatility of both the model and the framework. The training requires labeled parameter/anomaly score data samples. We trained our parameter change detection model (neural network) with just one simulation, where we labeled the disturbed crowd as

---

The hyperbolic tangent sigmoid function, $y = \frac{2}{1 + \exp(-2x)} - 1$, up to numerical difference mathematically equivalent to $y = \tanh(x)$, maps smoothly from $\mathbb{R}$ to its real range $[-1, 1]$. The log-sigmoid transfer function, $y = \frac{1}{1 + \exp(-x)}$, maps the whole real domain to $[0, 1]$.
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Figure 2.17: A bidirectional corridor crowd disturbed by a group of 5 individuals as described in the text. Left column: Visualization of the crowd as in Fig. A.2. Right column: Anomaly score. The disturbing group is visualized by black circles and the trailing lines show the group’s path over a short time horizon (10 seconds).

“anomalous” and all other times as “normal”, by minimizing the error between the network’s output and the training labels. For one simulation, Fig. 2.15 shows the average anomaly score calculated over all times the anomaly is present. The spatial distribution of the anomaly score reflects the crowds response to the disturbance, which is not limited to just the force’s spatial support. Fig. 2.16 presents the anomaly classification result over all 20 simulations for all times and inside force’s spatial support.

Small Group Disturbance

Scenario: A small, aggressive group disturbs a bidirectional corridor crowd.

The baseline for this simulation is a fully converged bidirectional corridor crowd. We insert a group of 5 individuals that performs an anomalous motion pattern, which is not aligned with the motion of any sub-population. Using the extension described in Sec. A.2.2, we assign an aggressive behavior to the group members, which we interpret as a high priority.

The group alters the crowd state which in turn alters the crowd motion parameters locally. Again, those parameters are calculated over a small 2-by-2 meter neighborhood on a discrete grid
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

Figure 2.18: Temporal evolution of the anomaly score of the bidirectional corridor crowd disturbed by a group of 5 individuals as described in the text (left) and cumulative histogram of the score (right). At more than 85% of all times, the anomaly score correct, i.e. it is above the threshold of 0.5 (dashed line).

that spans the surveilled domain. We define anomalous behavior using the parameter model defined for the “external forcing” case and use that case’s neural network to calculate the anomaly score. Even though the crowd state is only altered slightly (c.f. left column in Fig. 2.17), the extracted anomaly score exhibits the desired correlation between the group’s position and the associated anomaly (c.f. right column in Fig. 2.17). The spatial extent of the detected anomaly is a result of the crowds’ reaction to the disturbing influence, which is not limited to the immediate neighborhood of the small, anomalously behaving group. Fig. 2.18 presents the temporal evolution of the anomaly score. We note that we detect the anomaly correctly at more than 85% of all times. At all other times, the group is not altering the crowd behavior in a way that is detectable with our crowd motion model.

2.7 Concluding Remarks and Further Research Recommendations

In this chapter we described a framework to detect anomalies in large, dense crowds. We proposed a parameterized model structure capable of capturing dense multi-population crowd motion. Additionally, we suggested a method to estimate the crowd motion model parameters from known state variables.

The spline smoothing applied to the measurements to generate the states $\phi$ is known to produce unphysical results due to it neglecting the structure of the generating process (2.5) and the necessary knowledge of the weights $w_j$. This lack of information is corrected by transforming (2.27)
2. ANOMALY DETECTION IN LARGE, DENSE CROWDS

into a regularized least squares cost function by adding a roughness penalty term which is evaluated on the residual (2.21) using the spline approximation (2.26) [77]:

\[ S'(\alpha) \overset{def}{=} \sum_{j=1}^{N_x} (\phi(x_j) - \hat{\phi}(x_j; \alpha))^2 + \lambda \int_{\tilde{\Omega}} \left[ R_i(x, t; \hat{\theta}_i) \right]^2 \, dx. \]

The second term reintroduces the physical meaning of the generating process into the smoothing operation and thus it is expected to produce more appropriate results.

Another area of interest proposed here is the estimation of the principle variables from video data. The crowd anomaly detection relies on the knowledge of the distributed motion state estimated online. This topic was not addressed yet as appropriate video material was – as of now – unavailable.

To solve the state estimation step, we suggest the following procedure taking continuous video feeds as an input to estimate \( \phi \). As delineated above, the distributed states of the large-scale crowd model comprise of the densities, \( \rho_i \), and the corresponding velocities, \( u_i \). The observed velocity is a weighted average of a subset of sub-population velocities, and the observed density is the sum of the densities of the participating sub-populations. Even though that the states are continuous functions in space, in reality, these quantities are discretized over a spatial grid, and are defined by smoothing filters over small neighborhoods of each point. Note, in theory, if the smoothing filters’ extent is made infinitesimally small, each point will only feature (at most) one non-zero density and thus velocity of a single sub-population. The velocity states are estimated effective preexisting methods. For example, observed velocities can be estimated from the optical flow. On the other hand, generating a reliable density estimate for the proposed method is more difficult. State of the art methods are not applicable to high-density crowds as those methods typically rely on counting individuals in a crowd. However, considering strong occlusions and a single individual only represented by a hand full of pixels, those methods tend to fail. We thus suggest that density may be viewed as a texture property of the video data and that \( \rho \) can be estimated from small-area statistical properties of wavelet coefficients, resolving a range of length scales.

Lastly, the most promising model learning process needs to be selected from the list given in Sec. 2.5.1. Applying real-world video data to the proposed framework scheme will prove its validity for detecting anomalies in large, dense crowds.
3

Accelerated Video Labeling for High Quality Ground Truth Generation

The accurate annotation of videos is a cost intensive process. In this chapter, we propose methods to accelerate the generation of high quality video labels through the deployment of a flexible annotation framework with computer-aided labeling support presented in an effective, user-friendly graphical user interface.

In the following Sec. 3.1, we give a brief introduction, Sec. 3.2 presents a flexible annotation framework, which allows the creation of virtually any desired label. Our solution for a computer-aided labeling support with its main focus on spatial bounding box annotations, is described in Sec. 3.3. We close this chapter with Sec. 3.4, reviewing the implementation of the video labeling software suite, ANchOVy.

3.1 Introduction

Modern computer vision-based analysis such as medical and security research depends typically on sophisticated machine learning algorithms, which employ statistical reasoning [89] to find approximate solutions for explaining, classifying, and predicting visual data. An example at hand in surveillance applications is the automatized task to determine if a given individual has been previously observed over a network of surveillance cameras. Such problems can be approached by using probabilistic machine learning or pattern recognition algorithms as, for example, supervised and semi-supervised learning [88]. (Semi-)supervised learning algorithms infer models by analyzing labeled training data, which can be used to map previously unseen data.
3. ACCELERATED VIDEO LABELING FOR GROUND TRUTH GENERATION

The development and use of such algorithms requires a comprehensive video footage database of long video sequences with high-accuracy labels, which are also named ground truth (label) data. These labels serve as the trustworthy “gold standard”, which is the highest quality test data available under reasonable conditions, for the validation of the hypotheses of statistical algorithms, testing, tuning, and rigorous performance evaluation. The type of labels and the combination of different labels depends on the research objectives as well the methods for the performance evaluation of those objectives.

It is well known that the accuracy and cost of the labeling are correlated and that the process of annotating videos can be tiresome, which leads to operator mistakes and a degraded sensitivity and specificity. On the other hand, surveillance videos often contain long sequences without a single object of interest present.

Video labeling methodology

Conventionally, video labeling was executed by annotating every single frame manually. This typically leads to a decreased labeling quality if the videos contain complex motion and/or complicated setups. The labeling process was accelerated by exploiting the assumption that objects move smoothly and thus only a sparse subset of frames is labeled and some interpolation is used to infer the labels of the unannotated sequences. This approach is pursued by the “ViPER Ground Truth Authoring System”, which was presented as an annotation tool to be employed for video information extraction ranging from object detection to media understanding task such as event analysis [46].

One strategy to achieve accurately annotated visual content is the use of crowdsourcing. When solving a problem by crowdsourcing, a project is divided into smaller tasks, which are executed by a large number of people enlisted through crowdsource marketplaces [52]. The video annotation tool “Vatic” is using this approach to annotate publicly available videos [53]. However, crowdsourcing is inapplicable for handling proprietary, sensitive, or even classified videos. For example, the sharing of sensitive security information (SSI) is governed by Title 49 Code of Federal Regulations (CFR) parts 15 and 1520, which states that those data cannot be shared with the general public. Another application example, where crowdsourcing is not viable, is the labeling of medical images [96], which requires experts of specially trained personnel to generate reliable video labels.

To overcome those limitations, which become a prohibitive burden in medical and security research, we propose a flexible annotation framework with computer-aided labeling support. Its deployment, presented in an effective, user-friendly graphical user interface operating with a short
3. ACCELERATED VIDEO LABELING FOR GROUND TRUTH GENERATION

latency, will accelerate video label processes. As part of this research effort, we implemented this strategy and are developing a human operator-supervised, computer-aided software package for cost-effective photo and video footage labeling and metadata generation, which we named “Annotation Of Objects In Videos (ANchOVy)”.

3.2 Video Labeling Data Framework

Subsequently we will define a general-purpose design goal for video labels and devise a structure of label types to accommodate a wide field of applications.

Videos record a sequence of visual images, which contain a wide range of information of interest to the many research communities. Such information ranges from low-level characterizations, as for example the category of the recorded scene (e.g., urban, mountains, human heart, etc.), to high-level descriptions, such as the semantics of complex activities present in the video. The information recorded in videos is used in a wide spectrum of tasks. For example, the medium-level vision task of detecting semantic objects of a certain class (e.g. tree, vehicle, or building) requires knowledge about the presence of such objects in the video. In order to extract this knowledge, video information must be defined by a set of properties, which is then used to classify the content of the visual images. The choice and combination of those properties, which we call labels, varies by the application of video information and include a wide range of categories, such as the position and type of objects present in the video, which can be used in object identification, or the labeling of coordinated actions between people, which is of interest in security applications for the detection of terrorist activities. This means in particular that the labeling framework must be flexible enough to support such diverse tasks. Having this design goal in mind, we devised a framework of label types to accommodate a wide field of applications.

Flexible label framework

Primarily, we distinguish between two label classes: (a) Sequence-labels concerning certain sequences or the whole video; and (b) Object-labels representing properties of an object entity present in the video. For the latter class we further distinguish between time-constant property labels, such as gender or size, which have a one-to-one relationship with the associated object, and time-varying annotations such as position, extent, or motion type. Since spatial position and extent annotations are the most commonly used labels in computer vision research, special care was taken in crafting these
3. ACCELERATED VIDEO LABELING FOR GROUND TRUTH GENERATION

Label classes. We support rectangular, ellipsoid, and arbitrarily shaped extent annotations. The latter sub-class approximates the bounding contour with a snake using an active contour model [97].

A selection of typical low-level annotation types is listed in Figure 3.1. The association of arbitrary combinations of attribute and annotation labels to object entities (or groups of entities) is then creating more complex label types, which can be grouped together in semantic blocks.

![Label classification and corresponding examples.](image)

**Figure 3.1: Label classification and corresponding examples.**

**Label scripting language**

A label classification scheme can be defined by recognizing that every label can be constructed by a combination of describing properties. Example label properties are the label type and description, the data class used to represent this label type, the number of values, the allowed range or set of data values, their temporal propagation, possible constraints, as well as the label’s representation in a graphical user interface, edit ability, etc. We utilized this abstract label definition scheme to derive a simple scripting language used to create both labels and label blocks via the combination of predefined properties. This allows us to define almost any set of labels tailored to a specific annotation task.

In order to accelerate a video labeling process, we implemented a text file parser, which translates such previously stored label definitions into readily applicable video labels to be used during the actual labeling process with ANchOVy, where an instructed operator assigns them to a video. Storing and reusing label definitions also removes the possibility of ambiguously defined video labels and thus creating a trustworthy video information database.
3. ACCELERATED VIDEO LABELING FOR GROUND TRUTH GENERATION

3.3 Operational Support

The cost driver of video labeling is related to the length of the video and the number of labels to assign. Assuming a fully manual process of assigning $K$ labels to $N$ entities visible in a $T$ seconds long video recorded with $F$ frames per second, an operator would have to conduct at least $KNTF$ user operations to fulfill this task. From our experience we know that one such user interaction takes on average about 7 seconds. This means that the extraction of information captured by a single label for one entity would amount to a time investment of 3.5 hours for each video minute. However, more complex information annotation might require even more time.

The key enabler to mitigate the annotation costs while maintaining accurateness is a computer-aided annotation process supervised and corrected by human operators. For example, targets can be clearly visible, un-occluded for several consecutive frames. In these cases, it is possible to reduce the burden on the annotator by simply propagating the labeling and asking the operator to label again only when there is ambiguity; for example, by detecting a large change in appearance or proximity to other moving targets. This reduces the operator task to initially label key frames only, to monitor the label quality, and to take corrective action through label correction until the desired label quality is reached. In cases where only a small workforce can be employed, accelerating the video labeling process is the only viable option to generate minimal-cost annotations.

The implementation of such accelerating strategies is, as far as we know, limited to the propagation of position and extent annotations across video frames. Those bounding box labels represent the location and size of an entity in the video frame. While the ViPER tool is only supporting linear interpolation of bounding boxes \cite{46}, the developers of Vatic use a two-stage propagation scheme reported in \cite{53}, which consists of an online phase of key frame labeling and a path interpolation of unlabeled frames conducted offline using computationally expensive algorithm. However, all other types of labels are not propagated from one frame to the next. With the implementation of the ANchOVy framework we support different kinds of label propagation contributing to the acceleration of video information extraction. Specifically, the propagation of label values is considered to be a label property in the framework detailed in Sec. 3.2. For all logical and numeric label types we support constant, linear, and spline interpolation. All spatial annotation, such as bounding boxes, can optionally be propagated using the algorithm presented in Sec. 3.3.1. We are currently investigating a special case of label propagation, which uses constraints to conditionally assign labels. An example use would be a binary (true/false) label for a pedestrian walking into a predefined area, which would be set to true if the spatial position annotation is placed inside that area.
3. ACCELERATED VIDEO LABELING FOR GROUND TRUTH GENERATION

3.3.1 Bounding Box Propagation

A high-quality temporal propagation of position and extent annotations, such as the bounding box label, in a sparsely annotated video sequence for a single object is addressed in this section.

Without loss of generality, let \( t \in \mathcal{T} \equiv \{0, 1, \ldots, T\} \) denote the video frame number between the first \( (t = 0) \) and last \( (t = T) \) annotated frame, and write \( \mathbf{p}_t = \mathbf{p}(t) \) for the position and extent of a bounding box, \( \mathbf{p} = (x, y, w, h) \), measured in image coordinates. We define \( \mathcal{T}^\circ = [t_1, \ldots, t_N] \subseteq \mathcal{T} \) as the \( N \)-length set of those frame numbers, which contain an operator annotated bounding box \( \mathbf{p}_t^\circ \) from the set of all assigned labels \( \mathcal{P}^\circ = \{\mathbf{p}_t^\circ\}_{t \in \mathcal{T}^\circ} \). Then the trajectory, \( \mathcal{P} \), of the investigated object is given by one bounding box at all frames \( t \in \mathcal{T} \), i.e. \( \mathcal{P} = \{\mathbf{p}_t\}_{t \in \mathcal{T}} \) and we write \( \mathcal{P}[a, b] \) for a trajectory segment between the times \( a \) and \( b \), \( a < b \).

The bounding box propagation task is to find the path \( \mathcal{P} \) under the constraint that \( \mathbf{p}_t = \mathbf{p}_t^\circ \ \forall t \in \mathcal{T}^\circ \), i.e. that the propagated task must pass through the assigned labels. We relax this constraint to account for small annotation errors in the set \( \mathcal{P}^\circ \), by requiring that the interpolated path, \( \mathcal{P} \), must hold

\[
\frac{1}{N} \sum_{t \in \mathcal{T}^\circ} \| \mathbf{p}_t - \mathbf{p}_t^\circ \|^2 \leq \mu,
\]

where \( \mu \geq 0 \) is the maximum allowed average error in the sense of some appropriate norm, \( \| \cdot \| \).

This problem is readily solved by employing a piecewise linear interpolation, which would result in \( \mathbf{p}_t = \mathbf{p}_k + \frac{\mathbf{p}_{k+1} - \mathbf{p}_k}{t_{k+1} - t_k} (t - t_k), \ t \in [t_k, t_{k+1}] \), and \( t_k \in \mathcal{T}^\circ \). However, the assumption of linearity for the temporal evolution of the bounding box is known to be faulty on long time scales and only holds in short time intervals, which would require many manually labeled frames and thus increase the annotation project costs.

Yuen states in [47] that suitable tracking algorithms supporting the labeling process are too computationally expensive for efficient, low-latency use, preventing the labeling operators from interacting with videos in real time; Yuen’s method employs therefore the low-latency but inferior method of using a homography-preserving shape interpolation to propagate annotations temporally that is aided with global motion estimation. To address this problem, we propose a two stage trajectory extraction method to infer the object location, using a fast implementation of a featureless tracking-by-detection combined with a dynamical system motion model identified through the analysis of Hankel matrices. The first stage of the interpolation is an unsupervised preprocessing of the video and extracts the trajectories of moving objects visible in the video. The second, online stage of the interpolation is using the previously collected trajectories and combines them with the additional knowledge introduced by the sparse annotation set to find the interpolated trajectory.
3. ACCELERATED VIDEO LABELING FOR GROUND TRUTH GENERATION

Since computation time is virtually for free when compared to human labor cost, the suggested method transfers computationally expensive operations of the spatial interpolation from the actual labeling process involving a human operator to a preprocessing stage, which is completed without supervision before the labeling process starts. Furthermore, we suggest to store those preprocessed trajectories in a database using a \( k \)-d tree, which minimize their retrieval during the online stage. An annotation software employing the proposed interpolation scheme allows a near real-time use.

3.3.2 Online Bounding Box Trajectory Interpolation

The online path interpolation takes a consecutive, ordered set of video frames, \( \{I_t\}_{t \in T} \) and a sparse label set, \( P^o \), as input and returns the interpolated bounding box trajectory, \( P \), for the object. This task is executed by employing appearance-based tracking-by-detection and a parameter-free motion model. This interpolation reduces to a typical tracking task, if the cardinality of \( P^o \) equals one, which is a challenging problem [7]. However, in the interpolation case, we can exploit the knowledge of the object’s location and appearance at some time instances introduced by both \( P^o \) and \( \{I_t\}_{t \in T^o} \), respectively.

Tracking-by-detection

The first step in our proposed interpolation scheme is the extraction of trajectory segments starting at all known object locations \( P^o \) with a tracking-by-detection approach. We used the techniques described in [98], which we named “circulant tracker”, to implement this tracking task. This method approaches tracking as an object detection problem and uses an efficiently online learned classifier to discriminate the object from all other objects. The discriminative classifier is initially learned and continuously updated with a dense sampling strategy. At each frame, this strategy collects a set of strongly overlapping samples, which are sub-windows of constant size of the frame, around the target’s position. In contrast to classical linear classifier learning, such as methods employing support vector machines, where samples collected near the object of interest are labeled positive and all others are negative, Henriques et al. earns the classifier with Kernel Regularized Least Squares, which allows for continuous label values [98]. The main contribution of [98] is its theoretical framework, which allows for Kernel classifiers based on dense sampling and their closed form solutions for a most efficient training and evaluation using the Fast Fourier Transform. This
3. ACCELERATED VIDEO LABELING FOR GROUND TRUTH GENERATION

method is capable of handling partial occlusion, which occurs often in CCTV footage recorded from a low angle or in crowded areas.

At each frame, we assess the tracking confidence measured by means of both the maximum value and the peak-to-sidelobe ratio of the “circulant tracker” response. The former tests the degree to which the object classifier matches the tracked region, whereas the latter measurement provides the information about the certainty of the spatial correctness of the tracking result. The tracking is discontinued if the maximum response value is lower than a given threshold. The tracking is also terminated if the peak-to-sidelobe value falls below a predefined threshold [99] or if its temporal evolution is dramatically changing dynamics.

For each known object location, $p_k$, we employ this tracking algorithm both forward and backwards in time to create a continuous bounding box sequences, $\hat{P}_k \overset{def}{=} p|_{t_k-\tau_k^{-}, t_k+\tau_k^{+}}$. This partitions the whole trajectory $P$ into sequences, named tracklets in [100], which are known and unknown in alternation:

$$P \overset{def}{=} \{ \hat{P}_0, \hat{P}_0, \hat{P}_1, \hat{P}_1, \ldots, \hat{P}_{N-1}, \hat{P}_{N-1}, \hat{P}_N \}, \quad (3.1)$$

where the unknown trajectory segments, $\tilde{P}_k$, join the known adjacent segments $\hat{P}_k$ and $\hat{P}_{k+1}$ and are defined for times $t_{k+\tau_k^{+}} + 1$ through $t_{k+1} - \tau_{k+1}^{-} - 1$.

Incorporating the object’s dynamics

To fill the remaining gaps, $\tilde{P} \overset{def}{=} \{ \tilde{P}_k \}_{k=0}^{N}$, in the trajectory $P$, we propose to use motion dynamics estimated from all known sequences $\hat{P} \overset{def}{=} \{ \hat{P}_k \}_{k=0}^{N}$ without an explicitly stated dynamical model. To this account, we assume that the known sequences, $\hat{P}$, are generated by an autoregressive model whose order and structure remain essentially constant over $T$. This allows us to use an algorithm that is based on the iterative Hankel total least squares (IHTLS) algorithm suggested by Dicle et al. to reconstruct the missing values in $\tilde{P} [101]$. We detail our take on IHTLS in Ch. 4.

The sole parameter required for the IHTLS is the maximum allowed, average error, $\mu_{\text{max}}$, which, loosely formulated, regulates the amount of noise modeled. Lower values of $\mu_{\text{max}}$ will result in a lower-dimensional approximation of $P$, whereas IHTLS tends to overfit random errors and noise if $\mu_{\text{max}}$ is too large. Having said this, it is important to recognize that $\mu_{\text{max}}$ must reflect the accuracy of the coordinates in $\hat{P}$, and wrong choices of $\mu_{\text{max}}$ will lead to unphysical results.
3. ACCELERATED VIDEO LABELING FOR GROUND TRUTH GENERATION

3.3.3 Accelerating Preprocessing Stage

The described online path interpolation relies on the knowledge of short trajectories of the object of interest and can thus be accelerated by a preprocessing stage extracting those trajectories from the video beforehand.

This is implemented through an unsupervised tracking process extracting a set of short tracklets for all moving objects visible in the video. We consider the process of tracklet extraction an unrestricted visual search and track task, i.e., we assume that there exists no prior knowledge about both the location where objects will appear and their visual appearance. However, preexisting knowledge about the surveilled area, such as clearly defined entry and exit areas, can reduce the computational expense of the subsequently described method.

As mentioned before, surveillance videos often contain long sequences without any object of interest present. This can be exploited to further decrease the computational costs segmenting the video into blobs of interest (foreground) and static scene elements (background). To this account, the preprocessing stage learns and maintains a background model for the low-level segmentation of foreground blobs. Assuming a slowly varying background, continuously updated Gaussian mixture models offer the best trade-off between sensitivity and specificity for this task. We improve the segmentation quality with a morphological region processing and topological operations – a small area erosion followed by a dilation, to be precise – both resulting in noise reduction. This limits the tracking task to a significantly smaller subset compared to analyzing the whole video frame as we only need to track foreground. The object tracking is implemented in an oversampled manner by deploying multiple, overlapping trackers per foreground blob.

We note that most recent object trackers rely on strong assumptions on the target [7] and heavy optimization methods (e.g. [102]) rendering them fragile and not readily applicable to the greatly differing nature of CCTV footage. Thus we decided to employ the “circulant tracker” [98] to calculate short-term tracklets of all moving objects contained in the video. This is a prohibitively expensive task if conventional implementations of this tracker are used. Having said this, we removed this constraint through the implementation of a heavily parallelized version of the “circulant tracker” operating on graphics processing units and thereby allowing us to process videos in more than real-time.

The accelerating preprocessing stage processes the whole video or a selected portion of it and generates a $K$ short trajectories $P^i_k = \{p_t\}_t$ for all times $t \in [t_k^-, t_k^+]$. All generated tracklets are finally stored as tuples, $(t_k, P^i_k)$, for later use in the online path interpolation, where
3. ACCELERATED VIDEO LABELING FOR GROUND TRUTH GENERATION

they will be employed as the \( \hat{P} \). During the online phase we are given a point in image coordinates, \((x_r, y_r)^T\) at time \( t_r \), and select the best matching short trajectory from all \( \{P_k^{\dagger}\}_{k=1}^K \). Here, we define “best matching” in the sense of finding the \( k \)-th among all \( K \) short trajectory for which the following equation holds

\[
\min_k \|AP_{t_r} - (x_r, y_r)^T\|_2^2, \quad \text{s.t. } t_r \in t_k,
\]

\[
A \equiv \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}.
\]

3.4 The Software Suite “ANchOVy”

The software suite “Annotation Of Objects In Videos (ANchOVy)” is a software package for cost-effective CCTV surveillance footage labeling employing the aforementioned accelerating measures \[103\]. ANchOVy features a unified graphical user interface (GUI) shown in Fig. 3.2, which was designed for easy-to-learn and fast operational video labeling procedures. It features the flexible annotation framework introduced in Sec. 3.2 and the accelerating operational support (c.f. Sec. 3.3). In addition, ANchOVy is currently supported on all Mac OS X and Linux-based platforms due to its implementation in Java, and allows for a near real-time use by coding computationally complex methods in C++. Since, ANchOVy’s core routines make use of some Matlab libraries, we require an installation of Matlab’s runtime libraries which are at no cost.

ANchOVy also implements a role-based content control concept to accommodate the protection of data by defining three different user roles (supervisor, operator and user; c.f. Figure 3.3). Each role offers individual user rights; users, for example, have a read-only limited access to annotation data. The user identification, set by the operation systems mandatory access control, can also be used for annotation quality control as well as a cost analysis when combined with operational statistics, which ANchOVy collects at all times.

The ANchOVy software suite consists of multiple jointly operating modules as displayed in Figure 3.3, which are explained in the following.

**Sensor data input module**

The sensor data input module (SDIM) prepares the data to be analyzed by the software. Optical surveillance data is in essence a sequence of still images recorded at discrete times. Digital recording systems, typically used in CCTV frameworks, present the film footage in a container.
3. ACCELERATED VIDEO LABELING FOR GROUND TRUTH GENERATION

Figure 3.2: ANchOVy’s graphical user interface.

format following the AVI or MPEG standards. Since publicly available video reading libraries are not meant to access single frames, they implement simplified procedures to seek frames in videos. Thus, ANchOVy employs its own video accessing module, which ensures time-accurate frame grabbing and seeking.

To reduce the required data storage space, the video frames are compressed by lossy compression algorithms such as the MPEG or H.264 codecs, which, depending on the compression degree, introduce artifacts which impede the annotation analysis by exacerbating the already difficult task of supporting the spatial labeling process. The software provides an option to reduce those artifacts as well as to remove high-frequency noise while minimizing blending artifacts employing a combination of adaptive image gradient smoothing, unsharpening and a 3-way low-pass filter.

ANchOVy features reading of interlaced video formats. Because each interlaced video frame contains two images captured at different times, those frames can exhibit motion artifacts known as combing. Those interlacing effects are pronounced for fast moving objects impeding the exact definition of their shape, which makes bounding box annotation errors more likely. To avoid this source of mistakes, we optionally allow to rearrange the video frames into progressively
3. ACCELERATED VIDEO LABELING FOR GROUND TRUTH GENERATION

formatted video, which stores one field in one video frame.

Since ANchOVy’s annotation methods are in principle also applicable to pictures too, we also support the loading of still images saved in common picture format such as JPEG.

Figure 3.3: ANchOVy’s architecture.

Moving object recognition pipeline

The moving object recognition pipeline (MORP) with automated preprocessor implements the accelerating strategy described in Sec. 3.3.3 as an efficient, online labeling support. The MORP’s automated preprocessor computes spatial position annotation data for all visible, moving objects in the surveilled area, which can be employed by the online labeling engine, in an unsupervised, offline preprocessing task.

Labeling engine

The labeling process engine (LPR) is the object-labeling component directed and supervised by a human operator. With the use of ANchOVy’s GUI, the operator is easily assigning all types
3. ACCELERATED VIDEO LABELING FOR GROUND TRUTH GENERATION

of annotations. Since video annotation is repetitive and thus tiresome, we emphasized an ergonomic GUI design and included features to safeguard against operator errors (automated label propagation (depending on the type of annotation), continuous auto-save function, etc.). During the annotation workflow for bounding box labeling, the operator is inspecting the result of the MORP stage and can take corrective action. If such action is necessary, the bounding box annotations are updated and the object track is recalculated by the method described in Sec. 3.3.1, above. This procedure is repeated until a satisfactory annotation quality is achieved. Furthermore, to facilitate multi-scene projects and target re-identification research, the system maintains an object database with metadata annotations and cross-references between videos.

**Labeling data format**

The annotation data can be exported in extensible markup language (XML) and comma-separated value (CSV) format. Currently, ANchOVy ships with corresponding annotation data reading routines only for Matlab. However, saved annotation data is readily imported into other programming languages by using standard parser implementations for XML- and CSV-formatted text files.

3.5 Concluding Remarks and Further Research Recommendations

The label framework presented in Sec. 3.2 was implemented in ANchOVy, creating a general-purpose image and video labeling tool readily applicable to a variety of research fields and was used in surveillance and security studies, and recently in the field of breast cancer research [96]. ANchOVy was also used to label more than 600 passengers traveling through the Cleveland Hopkins International Airport (CLE). Out of a total number of about 250,000 frames annotated in this project, about 91.46% were derived algorithmically while still meeting the desired high-quality ground truth project requirement.

We currently investigate the application of a structured query language interface to access stored annotations by stating queries, which would allow the user to describe desired annotation data through label properties and values, for example: “Select all objects from ‘the CLE project’ after Jan 1, 2014 where the object wears black clothes and is carrying at least on piece of luggage.” This would allow for an automatized selection of, for example, training data tailored to the specific needs of sophisticated algorithms such as activity detection.
The scoring method used in the accelerated bounding box labeling support, described in Sec. 3.3, is employed by ANchOVy to indicate trajectory segments, which possess a low label quality and which should be corrected by the human operator. Currently, we compute this score with statistics of both the “circulant tracker” response matrix and the noise level returned by the IHTLS algorithm. Here we envision an optimized approach, which takes an anticipated label quality increase resulting from the operator correction into account. This would further decrease the labeling effort and associated costs. Lastly, we suggest to re-code the graphical user interface of ANchOVy in the programming language C++ to abandon ANchOVy’s legacy dependence on Matlab.
4

Trajectory Recovery from Short Sequences of Noise Corrupted Observations

This chapter concerns the recovery of missing segments of an object’s spatio-temporal trajectory in video data, leveraging the dynamics of the object’s motion by performing rank minimization of a Hankel matrix constructed from the available – possibly noise-corrupted – data.

This chapter is structured as follows. We formally describe the noisy trajectory recovery problem in Sec. 4.1, and propose to solve this problem by the means of a parameter estimation methods based on the Hankel total least norm framework in Sec. 4.2. Subsequently, we employ the notation introduced in Chapter 3.

4.1 The Noisy Trajectory Recovery Problem

Here, we consider the motion of an object as a stationary, vector-valued dynamical process generating the bounding box annotations, $P$, and describe such a process with the autoregressive model of low order $n$:

$$ p_t = a^T p_{[t-n,t-1]} + c , \quad t - n \geq 1 , \quad p_t \in \mathbb{R}^d , $$

where $a = (a_n, a_{n-1}, \ldots, a_1)^T$ are the parameters depending on the dynamics of $P$, and $c$ is some constant of the same dimension as $p_t$. Subsequently, we expect that $p_t$ is zero-mean over time and
4. TRAJECTORY RECOVERY

thus set \( c = 0 \).

To allow the following discussion, we assume that the object’s dynamics are similar over \( T \) and interpret this assumption in terms of a time scale \( \tau \gg T \) and the smooth dependency \( n = n(\frac{T}{\tau}) \). By this assumption, the system order, \( n \), remains essentially constant over \( T \) and thus for all tracklets used in the partition (3.1).

In the absence of noise, \( n \) in (4.1) corresponds to the system order or McMillan degree, which is a measure of the system’s complexity [104], and can be recovered by the Hankel matrix\(^1\) associated with \( P \),

\[
H_{P}^{(s,r)} \overset{\text{def}}{=} \begin{bmatrix} P_0 & P_1 & \cdots & P_s \\ P_1 & P_2 & \cdots & P_{s+1} \\ \vdots & \vdots & \ddots & \vdots \\ P_r & P_{r+1} & \cdots & P_{s+r} \end{bmatrix},
\]

(4.2)

whose rank equals to \( n \), i.e. \( n = \text{rank}(H_{P}^{(s,r)}) \) for some \( s, r \geq n \) [105].

We constrain the shape of \( H_{P}^{(s,r)} \) such that \( s \geq n \) and recover the number of rows from the relationship \( T = r + s - 1 \), where we use the known trajectory length \( T \). Simplifying the subsequent notation, we will write \( H_{P}^{(s)} \) for \( H_{P}^{(s,r)} \in \mathcal{H} \).

Applying (3.1) to (4.2) prohibits the explicit calculation of the system order, \( n \), due to \( H_{P}^{(s)} \)'s incomplete structure. In such cases – and also in the presence of noisy bounding box annotations in \( P \) – the rank of \( H_{P}^{(s)} \) must be estimated, which is a hard problem.

To close such type of problems, Dicle et al. suggest in [101] the repeated use of a Hankel total least norm (HTLN) algorithm [106], which reconstructs missing data while estimating \( n \). The optimal rank \( n^* \) is found by a repeated application of the HTLN algorithm with increasing rank until some average approximation error is less than some maximum allowed error. In the following, we propose an novel algorithm based on Dicle’s iterative Hankel total least squares (IHTLS) algorithm [101] to recover the unknown trajectory segments by exploiting the dynamics information captured in the known noisy tracklets while incorporating uncertainty information associated with each tracklet sample point.

\(^1\)A matrix \( H \) is said to be a Hankel matrix, if \( H \) has block-constant skew-diagonals, i.e. \( H_{i,j} = H_{i-1,j+1} \). The set of all matrices fulfilling this constraint is denoted by \( \mathcal{H} \).
4. TRAJECTORY RECOVERY

4.2 Order-selective Weighted Hankel Total Least Norm

In this section, we present a method to solve the rank estimation problem for a given
time series, $p_{[1:T]}$, as described above. The proposed algorithm, belongs to the class of parameter
estimation methods and as such is an extension of the total least norm algorithms for real-valued,
Hankel-structured problems. The two main components of the subsequently proposed method are (a)
the approximate solution to an overdetermined linear system of form $Ax \approx b$ which follows from
$p_{[1:T]}$, while preserving the Hankel-structure of perturbations in $[A|b]$, and (b) to find the best model
which explains the data, $p_{[1:T]}$, in the most efficient way. Finally, we present the order-selective,
weighted Hankel total least norm (wosHTLN) algorithm in Sec. 4.2.4, and study its performance in
Sec. 4.3.

4.2.1 Hankel Total Least Norm Formulation

Notation and requirements

As commonly used in structured total least norm formulations, we consider additive
noise in the principle variables, $P$, using the decomposition $P = P' + \eta$, i.e. for time $t$ we write
$p_t = p'_t + \eta_t$, where $P'$ and $p'_t$ denote the noise free trajectory and instantaneous value, respectively.
The noise part $\eta = \eta_{[1,T]}$ is recovered as follows.

For the sake of clarity, we introduce the notation used in the HTLN framework for scalar
signals, i.e. $d = 1$. The generalization to the multi-dimensional case is straight forward. For a given
system order $n$, let $b$ and $f$ be two vectors such that the two augmented matrices, $[A|b] = H_P^{(n+1)}$
and $[E|f] = -H_\eta^{(n+1)}$ be two Hankel matrices for sequences $P$ and $\eta$, respectively. Thus it follows
that the “cleaned sequence”, $P'$, is obtained as the first column and last row of $[A + E|b + f]$.

It is apparent that $P$ and $A$ are equivalent in the sense that given $P$ one can uniquely define
$A$ ($E$), and vice versa; the same is valid for equivalent $\eta$ and $E$, both defined by $\eta$. Note that $E$
is only depending on the subsequence $(\eta_1, \ldots, \eta_{T-1}) = R_0\eta$, where $R_0 \defeq [I_{T-1\times T-1} 0_{T-1\times 1}]$,
whereas the right-hand side perturbation can be expressed by $f = R_1\eta$ with $R_1 \defeq [0_{r\times s-1} I_{r\times r}]$, and dimensions $s = n + 1$ and $r = T - s$.

Now, since $[A|b], [E|f] \in \mathcal{H}$, any block matrices, such as $A$ and $E$, are also Hankel
structured and it follows from (4.1) that there exists a regressor $a$ such that $(A + E)a = b + f$ [107].
In the presence of noise, the latter equation may not be exact and is thus rewritten in residual form as
$r(a, \eta) = b + f - (A + E)a$, where $r(a, \eta)$ is the structured residual, which is not only a function of
4. TRAJECTORY RECOVERY

a but also of noise, $\eta$, due to the established fact that the perturbation matrix, $E$, is generated by $\eta$.

Furthermore, we note that there exists a relationship between $a$ and $\eta$ through the identity $Ea = XR_0\eta$, introducing the $r$-by-$T-1$ matrix $X$. $X$’s elements are generated by the entries of $a$ and its structure is determined by the following rule: If the element $E_{ij}$ is $\eta_k$, then the element $X_{ik}$ is $a_j$. Details of this matter can be found in, e.g., [106, 108].

Hankel Total Least Norm Formulation

The HTLN method finds a minimum-norm solution of structured perturbations $[E|f]$ for the overdetermined system $Aa \approx b$ [106, 108] such that $(A + E)a = b + f$ holds. This objective is formulated as

$$\min_{a,E,f} \|E|f\|_F^2, \quad \text{s.t.} \ (A + E)a = b + f, \text{ and } A, E \in \mathcal{H},$$

(4.3)

where $\| \cdot \|_F$ denotes the Frobenius norm, which is defined as $\|B\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n |B_{ij}|^2$ for an $m$-by-$n$ matrix $B$.

Employing both the established fact that $\eta$ is uniquely defining $[E|f]$ and the definition of the Frobenius norm, (4.3) can be expressed by [106]

$$\min_{a,\eta} \|D\eta\|_2^2, \quad \text{s.t.} \ r(a, \eta) = 0,$$

(4.4)

where $D$ is a diagonal matrix with positive integer-valued weights, accounting for the number of occurrences of each $\eta_t$ in $[E|f]$, i.e. $D$’s diagonal entries are limited by $0 < D_{ii} \leq s, i \in [1, T]$. Eq. (4.4) follows from (4.3), by the simple fact that if a Hankel matrix $H$, as defined in (4.2), is generated from a $T$-long vector $h$, then $\|H\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n |H_{ij}|^2 = \|Dh\|_2^2$ holds for $D = \text{diag}(d)$ with the generator $d = (1, 2, \ldots, n - 1, n, \ldots, n, n - 1, \ldots, 1)^T \in \mathbb{R}^T$.

Illustrating Example

We illustrate the HTLN formulation by the means of the short series $p_{[1,4]} = (1, 2, 3, 4)$. Let $n = 2$, then the corresponding Hankel matrix and perturbations are

$$[A|b] = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix}, \quad \text{and} \quad [E|f] = \begin{bmatrix} \eta_1 & \eta_2 & \eta_3 \\ \eta_2 & \eta_3 & \eta_4 \end{bmatrix},$$
4. TRAJECTORY RECOVERY

respectively, since \( s = n + 1 = 3 \) and thus \( r = T - s + 1 = 2 \). The augmented perturbations matrix \([E|f]\) determines the \( \eta \)-distribution matrices:

\[
R_0 = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}, \quad \text{and} \quad R_1 = \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\]

and the relationship between \( a \) and \( \eta \) is established by

\[
X = \begin{bmatrix}
x_1 & x_2 & 0 \\
0 & x_1 & x_2
\end{bmatrix}.
\]

Finally, we form the weight matrix in \((4.4)\): \( D = \text{diag}(1, 2, 2, 1) \). This concludes the illustrating example and \((4.4)\) can be solved with the methods discussed below.

4.2.2 Weighted Hankel Total Least Norm Formulation

In order to include the knowledge of \( P \)'s incomplete structure, the weighted Hankel total least Norm (wHTLN) method considers an indicator function, \( \omega_{[1:T]} \), where the \( t \)-th entry indicates the availability of data (\( \omega_t = 1 \)) or missing data (\( \omega_t = 0 \)).

Where the IHTLS algorithm is implicitly only using a binary \( \omega \), the weighted HTLN method, on the other hand, recognizes that the indicator function, \( \omega_t \), can be interpreted as an individual weight of the entries in the vector \( \eta \), and thus naturally expands \( \omega_t \) to be a nonnegative constant associated with the observation at time \( t \). The value of the weight, \( \omega_t \), indicates the importance of the associated observation in the optimization criterion. Now, optimizing the weighted problem \((4.6)\) to find the principle variable estimates allows the weights to determine the contribution of each observation to the final estimates. Note that each \( \omega_t \) is given relative to the weights of the other observations; thus different sets of absolute weights can have identical effects. To resolve this ambiguity for the problem at hand, we limit the weight range such that \( 0 \leq \omega_t \leq 1 \), \( \forall t \).

In the applied case of our trajectory recovery, \( \omega_t \) can encode the precision of the information contained in the \( t \)-th observation, and the weight equals to zero for missing data, i.e. \( \omega_t = 0 \) if \( p_t \neq \{ \tilde{P}_k \}_{k=[1,N]} \). Assume, for example, that an object is tracked through multiple occlusions with a tracking algorithm that not only returns the estimated object position over time but also calculates an uncertainty measure associated with each estimate, then it is reasonable to give little consideration to highly uncertain observations when attempting to recover the trajectory. The weighted HTLN
4. TRAJECTORY RECOVERY

framework allows to incorporate the uncertainty of tracking results through the weighting function $\omega$ (c.f. Sec. 4.3).

Applying the concept of weights to the original HTLN method leads to the weighted Hankel total least norm problem, and (4.3) becomes

$$
\min_{a,E,f} \| \Omega \circ [E|f] \|_F^2 , \quad \text{s.t.} \quad (A + E)a = b + f, \text{ and } A, E \in \mathcal{H},
$$

(4.5)

where $\Omega = H_\omega^{(n'+1)}$ is the Hankel matrix associated with $\omega_{[1:T]}$, and ‘$\circ$’ denotes the Hadamard product. Equivalently, (4.4) is modified and now written as

$$
\min_{a,\eta} \| WD\eta \|_2^2 , \quad \text{s.t.} \quad r(a,\eta) = 0,
$$

(4.6)

where $W \overset{\text{def}}{=} \text{diag}(\omega_{[1:T]})$.

Eq. (4.6) differs from the original HTLN problem in the fact that $\Omega$ modulates the norm while respecting the structural constraints of Hankel matrices. In other words, the enabling main concept of wHLTS is that it removes the unknown segments, $\{\tilde{P}_k\}_{k=1,N}$, from the optimization cost in (4.7) and rather implicitly uses their reconstructed values in the optimization constraints to ensure that the dynamics relationship holds at all times, i.e. the residual is vanishing at all times ($r(a,\eta) \approx 0$).

Solution of the Weighted HTLN Problem

As a preliminary step and prior to solving the wHTLN problem, we scale each of the $d$ channels of the vector-valued input, such that they are mean-free and their sample standard deviation equals one. This normalization eliminates the effects of gross influences of signals of widely varying scales. After having obtained the solution, the normalization is reversed such that the original signal statistics are restored.

Even though (4.6) is a convex, quadratic optimization problem, the set of nonlinear equality constraints ($r = 0$) render it a hard to solve problem.

In Appendix B, we discuss three methods to solve the HTLN problem (4.5): an exact method employing a generalized singular value decomposition, an approximate method using the QR factorization, and an iterative scheme based on Newton’s method applied to Karush-Kuhn-Tucker (KKT) conditions. We discuss the approximate solution for two types of QR decompositions based on Householder transformations [109] and Givens-rotations [110].
4. TRAJECTORY RECOVERY

The benefit of the iterative Newton-KKT method is that it solves (4.6) directly rather than using the approximation (4.7), which avoids the inaccuracies introduced by the method of weights (see below). In Sec. 4.3, below, we show the performance of the approximate method as well as the Newton-KKT scheme, and compare it to the original IHTLS algorithm.

Most commonly, the wHTLN problem (4.6) is solved by using the method of weights, which we will revisit and discuss next.

Solution of the Weighted HTLN Problem via the Method of Weights

Solution of Eq. (4.6)

The method of weights transforms the optimization problem in Eq. (4.6) into an unconstrained problem by including the equality constraints in the cost function using a penalization of constraint violation \[110, 111\]:

\[
\min_{a, \eta} \left\| \begin{pmatrix}
\pi r(a, \eta) \\
WD\eta
\end{pmatrix} \right\|_2^2,
\]

where \(\pi \gg 1\) is some large constant, and for \(\pi \to \infty\), the difference of the solutions to (4.7) and (4.6) tends to zero.

In solving the problem (4.7), a first order Taylor expansion of the nonlinear term \(r(a, \eta)\) around its argument \((a, \eta)\) is used. Let \(\triangle a\) and \(\triangle \eta\) represent small perturbations of \(a\) and \(\eta\), respectively, and let \(\triangle E\) denote corresponding changes in \(E\), we obtain:

\[
r(a + \triangle a, \eta + \triangle \eta) = b + R_1(\eta + \triangle \eta) - (A + E + \triangle E)(a + \triangle a)
\approx r(a, \eta) + R_1 \triangle \eta - (A + E) \triangle a - \triangle E a
= r(a, \eta) + (R_1 - XR_0) \triangle \eta - (A + E) \triangle a.
\]

Now, by substituting the linearization of \(r\) into (4.7), the wHTLN problem becomes the unconstrained linear least norm minimization:

\[
\min_{\triangle a, \triangle \eta} \left\| M \begin{pmatrix}
\triangle \eta \\
\triangle a
\end{pmatrix} + \begin{pmatrix}
-\pi r \\
WD\eta
\end{pmatrix} \right\|_2^2,
\]

\[
M \overset{\text{def}}{=} \begin{bmatrix}
\pi(XR_0 - R_1) & \pi(A + E) \\
WD & 0
\end{bmatrix},
\]

(4.8a)

(4.8b)
4. **TRAJECTORY RECOVERY**

which can be solved iteratively, starting at $\eta=0$ and $a$ set to the least norm solution

$$a_{\text{LSE}} = \arg \min_a \| b - Aa \|_2 .$$

At each iteration, (4.8) is solved with updated $a \leftarrow a + \Delta a$, $\eta \leftarrow \eta + \Delta \eta$ and corrected depending variables $r, X, E$. The structured total least norm solution, $(\eta^*, a^*)$, is the argument of (4.8) when the norms $\| \Delta a \|, \| \Delta \eta \|$ fall below the given small thresholds, $\delta_a, \delta_\eta > 0$.

**Convergence of (4.8)**

For $R_1 = 0, W = 1$, and $\pi = 1$, the convergence of this algorithm was proven in [108] by establishing that a solution of (4.8) is, in effect, a Gauss-Newton step applied to the gradient of the cost function

$$\phi(a, \eta) \equiv \frac{1}{2} \| \pi r(a, \eta) \|_2^2 + \frac{1}{2} \| WD\eta \|_2^2 ,$$

which is just (4.7) written in differentiable function form. The argument of [108] is generalized next to variable $R_1, W$, and $\pi$.

The tuple $(a, \eta)$ is a local optimum of $\phi(a, \eta)$ if all gradients $\nabla_a \phi$ and $\nabla_\eta \phi$ vanish. These are the first-order optimality conditions, which, using the established relations, become

$$\nabla_\eta \phi = -\pi^2 (XR_0 - R_1)^T r + (WD)'(WD)\eta = 0 , \quad \text{(4.9a)}$$

$$\nabla_a \phi = -\pi^2 (A + E)^T r = 0 . \quad \text{(4.9b)}$$

The latter equations are related to the argument of (4.8a) through

$$- \begin{pmatrix} \nabla_\eta \phi \\ \nabla_a \phi \end{pmatrix} = MT \begin{pmatrix} \pi r \\ -WD\eta \end{pmatrix} = MTM \begin{pmatrix} \Delta \eta \\ \Delta a \end{pmatrix} , \quad \text{(4.10)}$$

where the last equality follows from considering the normal equations of the least-squares solution of (4.8). $(\Delta \eta, \Delta a)$, the solution of (4.10), is unique if $MTM$ is positive definite, which is true if $M$ has full rank. The latter requirement is satisfied if $A + E$ exhibits full column rank. Now, the optimality conditions (4.9) are satisfied if the left-hand side of (4.10) vanishes, which holds in the limit for a converging solution of (4.8), i.e. $(\Delta a, \Delta \eta) \approx 0$. Furthermore, this shows that (4.8) is a Gauss-Newton step which approximates $\phi(a, \eta)$’s Hessian by $MTM$ [108].

Last, we note that (4.7) and the associated function $\phi$ are non-convex, and thus any solution
4. TRAJECTORY RECOVERY

satisfying (4.9) is a stationary point of $\phi$ but not necessarily a global optimum. However, generally (4.8) will converge to the local minimum closest to the starting conditions.

For the unconstrained problem (4.7) to be asymptotically equivalent to the constrained minimization (4.6), large weights, $\pi$, need to be introduced [112], yielding stiff, ill-conditioned matrices and thus inaccurate results when solving (4.8). For example, a solution via the normal equations using the Cholesky factorization of $M^T M$ may yield erroneous results. However, by applying carefully chosen methods this ill-conditioning is overcome. In Appendix B, we discuss this issue as well as possible strategies to accurately solve the wHTLN problem.

4.2.3 Model Order Selection and Validation

As we explained above, the wHTLN, i.e. the solution to (4.7) for some observed time series, yields both the estimated $P'(n)$ and corresponding noise $\eta(n)$ depending on a given model order, $n$, as well as the corresponding autoregressor parameters, $a$.

Now, we turn our attention to the model order, as the remaining question is: Which model describes the time series in the best way and in the most efficient manner? Due to the fact that the model order is, in general, not known \textit{a priori}, we describe two options to quantify and evaluate this model order choice.

Assume we are given a time series with nonzero additive noise $P = P' + \eta$ (c.f. Sec. 4.2.1), then, obviously, the best approximation will not be the model yielding a vanishing $\eta$. In other words, it is not the goal to minimize (4.7) s.t. $\eta=0$, but rather to drive the perturbations to the true noise level/power. However, this problem is not readily closed because the true signal is unknown and thus the true noise power cannot be identified. Therefore a surrogate for the true noise power is necessary. One possibility is to use the $\Omega$-weighted estimated perturbations to calculate a weighted error,

$$
\mu_\eta(n) \overset{\text{def}}{=} \frac{1}{\sqrt{d_1 \| \omega \|_1}} \sum_{t=1}^{T} \omega_t \| \eta_t(n) \|_2.
$$

Dicle et al. circumvent the noise level estimation problem by rendering the noise power an input parameter to their IHTLS algorithm [101]. Now, one IHTLS-iteration solves an HTLN problem for a given order $n=1, 2, \ldots$ and uses the recovered noise sequence, $\eta(n)$, to compute an average error. The IHTLS algorithm exits with a solution when the average error falls below a some predetermined threshold. This approach bears the problem that a correct threshold must be known \textit{a priori}. For example, if the error bound is too large, the estimated model order will be too large.
4. TRAJECTORY RECOVERY

and the recovered “noise-free” series, \( P' \), will indeed model parts of the noise. This means that in cases where the uncertainty of the threshold for the noise power is high, as for example use case of our noisy trajectory recovery, other means of finding the optimal model order are required. Here, we resort to a combined model order selection and model validation scheme to close the problem of selecting the correct model, which we explain subsequently.

4.2.3.1 Model Order Selection Scheme

Criteria for a sensible determination of the optimal model order are statistical measures describing the quality of a model for an observed set of data, and most commonly are expressed as a function of the form [113]:

\[
C_p(n) = f_p(n) + n \frac{c_T}{T},
\]

where a goodness of model fit term, \( f_p(n) \), is balanced with a \( c_T \)-weighted term penalizing the model’s complexity, which increases monotonically with \( n \). In other words, in order for a large \( n \) to minimize \( C_p(n) \), a higher model complexity must be merited by a significant fitting quality improvement. Let \( n^* \) be the model order optimal in the sense of (4.11), then \( C_p(n^*) \) is minimal over the set of all feasible \( n \). According to Paulsen [114], only criteria of form (4.11) can achieve consistency in the sense that, as \( T \to \infty \), the selected \( n^* \) will be the true model order with probability one, if \( c_T \) is a function of \( T \) that fulfills the properties \( c_T \to \infty \) and \( c_T/T \to 0 \) as \( T \to \infty \).

Over the years, many criteria fulfilling the Paulsen requirement were developed, each targeting a different specific area of application, and we suggest the comprehensive report given in [115] for in-depth information. One such criterion is the Schwarz criterion (SC), which is also known as the Bayesian information criterion. Here, we define it for some \( d \)-dimensional vector autoregressive time-series, \( p_{[1:T]} \), model (4.1), and order \( n \) as [116]

\[
C_p^{SC}(n) \overset{\text{def}}{=} \log |\Sigma_p(n)| + c \frac{\log T}{T},
\]

where \( c = d^2 n \) represents the number of model parameters,\(^2\) and \( \Sigma_p(n) \) is the unbiased, weighted covariance matrix,

\[
\Sigma_p(n) \overset{\text{def}}{=} \frac{1}{\sum_{t=1}^T \omega_t} \sum_{t=1}^T \omega_t \epsilon_t(n) \epsilon_t^T(n),
\]

of the multivariate residual series, \( \epsilon(n) = \{\epsilon_t(n)\}_{t=1}^T, \epsilon_t(n) = p_t - p_t(n) \) between the \( n \)-th order

\(^2\) Since our vector-valued model (4.1) is only employing \( n \) parameters, we simply set \( c = n \).
4. TRAJECTORY RECOVERY

approximation, \( p_t(n) \), of the reference \( p_t \). In the case of our trajectory recovery problem, the residual at time \( t \), \( \epsilon_t(n) \), is given by the estimated noise component \( \eta_t \) for the model of order \( n \). Now, the optimal model order in the sense of (4.12) is found by minimizing \( C_p^{SC}(n) \) over the set of feasible orders \( n=1, 2, \ldots \), i.e.

\[
n^* \overset{\text{def}}{=} \arg \min_n C_p^{SC}(n).
\] (4.13)

In addition to using the covariance matrix, we point out that parameter estimation algorithms based on likelihood maximization typically employ and explicitly calculate the log-likelihood for the parameters given \( p \). In those cases, an alternative form of (4.12) may be convenient where the first term is replaced by the maximized log-likelihood.

In closing this short discussion, we address the issue of selecting the minimal value of \( C_p^{SC}(n) \). In simple scenarios, the sequence \( \{C_p^{SC}(n)\}_{n \geq 1} \) will be well behaved and may exhibit a single pronounced minimum, which determines the optimal order, \( n^* \). However, this may not hold true for more challenging situations with considerable signal noise, such as the trajectory recovery. Therefore, we substitute (4.13) with the online knee-point detection algorithm of [117]. In employing this algorithm, we select the “optimal” model order as the \( n \), where the curvature\(^3\) of \( C_p^{SC}(n) \) is maximal.

4.2.3.2 Model Validation Scheme

Model validation is complimentary to the model order selection, discussed above. It is generally considered to be the validation that the predictions of some numerical model approximate the underlying physics of the data being modeled to some acceptable degree of accuracy [118]. Here, statistical tests of the residual, \( \epsilon \), are examined. We recall from the requirements established in Sec. 4.2.1 that the noise component, \( \eta \), is assumed to be mean-free Gaussian white noise with some unknown variance. Thus it follows that if the estimated model is indeed close the true model, the series \( \epsilon \) should behave like a random variable and its statistics will be similar to the statistics of the true \( \eta \). A variety of statistical tests for the degree of whiteness of signals have been developed and here we select the following method based on an investigation of the residual’s autocorrelation.

For a discrete process, the biased autocorrelation estimate for lag \( \tau \geq 0 \) is given by [119]

\[
\hat{R}_n(\tau) \overset{\text{def}}{=} \frac{1}{T} \sum_{t=1}^{T-\tau} \epsilon_{t+\tau}(n) \epsilon_t(n).
\]

\(^3\)Here we use the term “curvature” in the approximate sense of the curvature definition applied to discrete data as discussed in [117].
4. TRAJECTORY RECOVERY

If the whiteness hypothesis for the residual holds true,

$$\hat{\chi}^2_{M,n} \overset{\text{def}}{=} \frac{T}{R^2_n(0)} \sum_{\tau=1}^M \hat{R}^2_n(\tau)$$

(4.14)

will be asymptotically $\chi^2_{M}$-distributed, where $\chi^2_M$ denotes the $\chi^2$ distribution with $M$ degrees of freedom [78]. We denote this test as $\hat{\chi}^2_{M,n} \overset{\alpha}{\sim} \chi^2_{M}$, which compares $\hat{\chi}^2_{M,n}$ with the $\chi^2_{M}$-statistic; the hypothesis is accepted at the commonly used $\alpha = 95\%$ confidence level.

4.2.4 The Order-selective Weighted Hankel Total Least Norm Algorithm

Using the above, we now propose the order-selective, weighted Hankel total least norm (wosHTLN) algorithm, which estimates the rank of a noisy Hankel matrix with uncertainty weights through an iterative solution of weighted HTLN problems (c.f. Sec. 4.2.2) and a combined model order selection and validation scheme (c.f. Sec. 4.2.3), yielding the “optimal” rank, $n^\star$. The wosHTLN algorithm is shown as pseudo-code in Fig. 4.1.

The wosHTLN algorithm requires a time series, $p[1:T]$, with associated weights, $\omega[1:T]$, as input and returns the estimated model order $n^\star$, noise-free time series $p'[1:T](n^\star)$, noise component $\eta[1:T](n^\star)$, and regressor parameters $a(n^\star)$. The algorithm solves the wHTLN problem (4.6) iteratively for model orders $n = 1, 2, \ldots$ until a valid model with sensible model order is found. The model order selection is based on the “Schwarz criterion (SC)” (4.12) and the knee-point detection algorithm (c.f. Sec. 4.2.3.2). Any model that passed the the former test are then validated by testing the residual’s whiteness level through (4.14). If the hypothesis is accepted, the wosHTLN algorithm terminates, otherwise it attempts to find a better model with a larger model order.

In cases where a reliable maximum average error threshold, $\mu_{\text{max}}$, can be supplied, the additional termination criteria, $\mu(n) \leq \mu_{\text{max}}$, may be installed, which tests if the weighted average noise, $\mu(n)$ is small enough.

The proposed algorithm, belongs to the class of parameter estimation methods and as such is an extension of the total least norm algorithms for real-valued, Hankel-structured problems. It obtains an approximate solution to an overdetermined linear system of form $Ax \approx b$, while preserving the Hankel-structure of perturbations in $[A|b]$. The underlying dynamics of $[A|b]$ are captured with an autoregressive model, where a combined model order selection and validation scheme based on the “Schwarz criterion (SC)” and a $\chi^2$ test is used to estimated the model order. The wosHTLN method for estimating the rank of an incomplete, noisy Hankel matrix can be seen as
4. TRAJECTORY RECOVERY

Input: time series $\mathbf{p}_{[1:T]}$, weights $\omega_{[1:T]}$

Output: estimated model order $n^*$, noise-free time series $\mathbf{p}'_{[1:T]}(n^*)$, noise component $\eta_{[1:T]}(n^*)$, regressor parameters $a(n^*)$

for $n = 1, 2, \ldots$ do
  Form $\Omega = H(n+1)$, $[E|f] = H_p(n+1)$
  Solve (4.6) /* result: $\mathbf{p}'_{[1:T]}(n), \eta_{[1:T]}(n), a(n) */$
  Calculate $C^{SC}_p(n)$ with (4.12)
  Find knee-point, $\tilde{n}$, of sequence $(C^{SC}_p(1), \ldots, C^{SC}_p(n))$
  if no knee-point found then
    continue
  Calculate $\hat{\chi}^2_{M,\tilde{n}}$ with (4.14)
  if $\hat{\chi}^2_{M,\tilde{n}} \sim 95\% \chi^2_M$ then
    break

$n^* = \tilde{n}$

Figure 4.1: Order-selective Weighted Hankel Total Least Norm algorithm

a greedy algorithm, which makes a locally optimal choice at each iteration with the hope of finding the global optimal.

4.3 Performance Evaluation

In this section, experiments are presented to empirically prove the performance of the wosHTLN algorithm.

Since we are primarily concerned with spatial problems in two dimensions in this thesis, we analyze autoregressive models for two-dimensional time series, i.e. $d = 2$ and $\mathbf{p}_t \in \mathbb{R}^2$. To systematically study the performance of the proposed method, we resort to using synthetic test data and thus gaining the ability to define a wide range of challenging signals.

Synthetic data

The synthetic data generation employs an $n$-th order autoregressive model which we derive from a discrete $n$-th order all-pole linear time-invariant model. The $n$ poles are selected in a uniformly random manner, but we restrict their magnitude such that the generated signal is reasonably bounded, and limit the frequency to prevent atypical fast oscillations. Then we recover the autoregressive parameters, $a$, by writing the characteristic polynomial associated with the poles. We generate a set of random initial conditions and construct the $T$-long time series, $\mathbf{p}'_{[1:T]}$, by convolving $a$ with a
4. TRAJECTORY RECOVERY

$T$-long white Gaussian noise sequence with zero mean. To allow a comparison between two different signals, we normalize $p'_{[1:T]}$ to unit standard deviation.

The above procedure may be repeated for each of the $d$ channels of $p'_{[1:T]}$ to create different dynamics. Random errors that are normally distributed with mean zero and $\sigma^2_\eta$ variance are then added to the signal, $p'_{[1:T]}$, generating the perturbed signal $p_{[1:T]}$.

We use a Monte Carlo simulation to determine the modeling properties of the proposed method by testing it on 200 random models, where we repeat each test with 50 different noise signals.\(^4\)

For the wosHTLN algorithm we choose the following parameters, which are not tuned to the benefit of single experiments: The termination criteria are $\delta_a = 10^{-8}$ and $\delta_\eta = 10^{-5}$. The order selection scheme is executed by the means of the knee-point detection for the Schwarz criterion and the $\chi^2$-whiteness test, but is not using the optional termination through the analysis of the average error, i.e. $\mu_{\text{max}} = 0$.

The IHTLS algorithm serves as a reference solution, where we use the true noise variance, $\sigma^2_\eta$, as a termination criterion. IHTLS is solving the rank estimation problem by the means of the method of weights in a similar fashion as described in Sec. 4.2.2 employing a Householder reflections-based QR method on dense matrices [101]. Thus we have to choose a moderate constraint violation parameter, $\pi = 10^9$, as described in Appendix B.

Model quality measure

The frequently used normalized mean square error (NMSE),

$$\text{NMSE}(p_{[1:T]}, \hat{p}_{[1:T]}) \overset{\text{def}}{=} 1 - \frac{\|p_{[1:T]} - \hat{p}_{[1:T]}\|_2}{\|p_{[1:T]} - \frac{1}{T} \sum_{t=1}^T p_t\|_2},$$

is employed as a quantitative representation of the differences between values predicted by a model, $\hat{p}_{[1:T]}$, and a reference, $p_{[1:T]}$. The NMSE-cost varies between 1 (perfect fit) and $-\infty$ (worst fit). If the cost function is equal to zero, then $\hat{p}_{[1:T]}$ is no better than a straight line at matching $p_{[1:T]}$.

For a $d$-dimensional vector-valued argument, we define NMSE as the mean NMSE-cost over all $d$ dimensions.

\(^4\) The subsequently presented results are calculated as follows. Let $x_{ij}$ be the value of interest for the $i$-th random model and the $j$-th noise realization, then we calculate results concerning a single random model as the median value of $\bar{x}_i \overset{\text{def}}{=} \{x_{ij}\}_j$, and the overall Monte Carlo result is given my the mean over the median values $\bar{x}_i$. 

83
4. TRAJECTORY RECOVERY

4.3.1 Model Order Selection Criteria

For this test, we focus on the comparison of various model order selection criteria. We compare the performance of the Schwarz criterion (SC) (4.12), the well-known Akaike information criterion (AIC) [120], AICc (corrected AIC for short time series) [121], AICf (adapted AIC for vector-autoregressive series) [122], MLDc (minimum description length criterion for moderately long time series) [123], and MAS (mean of AIC and SC).

![Comparison of various model order selection criteria](image)

Figure 4.2: Comparison of various model order selection criteria used in the wosHTLN algorithm for the test framework described in the text and signal-to-noise ratios 5, 2, and 1 [dB] (dark, medium, and light gray bars, respectively). The reference solution (“Opt”) uses the true noise variance. Left figure: The filled bars represent the mean NMSE-cost over all simulations, and the error bars correspond in length to ±1 standard deviation of the NMSE. Right figure: We show the difference \( n - n^* \) of the wosHTLN-estimated model order \( n \) in respect to the true order \( n^* \) with the mean difference (filled bars) and ±1 standard deviation (error bars).

For this test we create synthetic data of length \( T = 100 \), as described above, and artificially remove about 50% of all \( T \) sampling points around a random location \( t \), separating the time series into two short segments. To test the stability against noise, we perform a Monte Carlo simulation for three signal-to-noise ratios, SNR = 5, 2, 1 [dB].

The interpretation of the test results, depicted in Fig. 4.2, shows that AIC is biased and selects on average a too large model order, which leads to a decreased NMSE-cost. It is surprising that the MLDc method is failing as it was designed to perform best on time series with a non-vanishing ratio \( n \frac{1}{T} \). SC outperforms the other model order selection criteria and the wosHTLN method selects on average the correct model order and thus confirms the choice of the Schwarz criterion for the model order selection task.
4. TRAJECTORY RECOVERY

4.3.2 Numerical Solution Methods

Figure 4.3: Comparison of numerical methods to solve the wosHTLN problem with the iterative method (“Newton-KKT”), as well as the method of weights with QR decomposition using both Householder reflections (“Householder-QR”) and Givens rotations (“Givens-QR”). The reference solution (“Opt-IHTLS”) uses the true noise variance and is generated by the IHTLS algorithm. The data is presented as in Fig. 4.2 (left) for signal-to-noise ratios 10, 5, 2, and 1 [dB] (dark through light gray bars, respectively). The two figures show the sensitivity of the sparse Householder algorithm to the constraint violation weight, $\pi$, of the method of weights; only the Givens-QR based method remains numerically sound for large penalizations. The results for both “Givens-QR” and “Householder-QR” are calculated with a moderated and large penalization, $\pi = 10^9$ (left) and $\pi = 10^{12}$ (right), respectively.

This test evaluates the performance of three numerical methods to solve (4.6) using synthetic data. We test the QR decomposition employing Householder transformations as well as Givens rotations, as discussed in Sec. 4.2.2 and Appendix B, to solve the unconstrained linear least norm minimization (4.8). As an alternative, we also solve the equality constrained HTLN problem of (4.6) using the iterative method discussed Appendix B, which is based on the the Newton-Raphson method applied to the problem’s Karush-Kuhn-Tucker conditions.

We apply the same testing framework (signal-to-noise ratios and reference solution) used in the Model Order Selection Criteria test, above, but remove the additional challenges of missing data, i.e. $\omega[1:T] = 1$. The test results are depicted in Fig. 4.3.

The results corroborate our discussions, above, in regards to the Givens rotations. Even though a large penalization ($\pi = 10^{12}$, c.f. right panel in Fig. 4.3) was selected, the Givens-QR decomposition remains numerically sound. This does not hold true for the QR decomposition using Householder transformations, which fails catastrophically. The QR implementation of the SUITEPARSEQR package [124], which we used here, is inaccurate in the case of matrices with
widely varying row-norms, because it is not using row permutations to stabilize the numerical solution, and it also destroys a preliminary established row ordering [109] as SUITE_SPARSE_QR is focusing on exploiting a potential sparsity structure. The iterative Newton-KKT method is, as expected, performing similarly good as the stable Givens-QR decomposition method.

The performance of the wosHTLN algorithm is expectedly deteriorating with decreasing signal-to-noise ratio. However, it is noted that the performance is not gracefully decaying, which is attributed to problems of the model order estimation in the presence of pronounced noise.

In closing this section, we revisit the example use case of trajectory recovery with a reference to low-latency applications. The typical spatial trajectory recovery task operates on video with about 30 frames per second and is either a two-dimensional problem if the motion is modeled in the video image space or even three-dimensional for world-coordinate modeling approaches. Such problems thus require the analysis of trajectories with a number of elements \((dT)\) in the order of 1000. From Fig. 4.3 we see that only the Householder-QR decomposition is fast enough to allow low-latency operations. This particular method is applicable even though its solution quality is only acceptable for medium-valued penalization weights \(\pi\). However, since our trajectory recovery problem is not susceptible to (very) small errors, this drawback does not pose a considerable issue.

4.3.3 Run-time Comparison

At the core of the investigated problem is the solution of (4.8), a linear system of equations with matrix \(M\) of size \(2dT - Rd\)-by-\(dT + R\), which becomes quite expensive for large \(T\). Focusing on the run-time, we show that algorithms exploiting the sparse structure of the matrix \(M\) in (4.8) must be selected in order to employ iterative HTLS-based methods in low-latency applications.

The subsequent run-time comparison was conducted on a single core of an Intel Core i7 (2.4 GHz) without using hyper-threading to allow a fair comparison between the presented methods, and results are shown in Fig. 4.4.

In solving (4.8) using the QR factorization of \(M\), we factorize \(M = QR\), and may form \(d = QTb\) to solve the triangular system \(Rx = d\). This method’s overall computational complexity is dominated by the first step, where the number of operations for archetype QR factorizations – when using Householder transformations – is \(2mn^2 - \frac{2}{3}n^3\) for dense \(m\)-by-\(n\) matrices. In the case of the investigated problem (4.8), the run-time will be following \(O(T^3)\), because in a Fermi estimation sense both \(R\) and \(d\) can be neglected as \(d/T, R/T \ll 1\). The results of this method are denoted as “Dense-QR” in Fig. 4.4.
Figure 4.4: Monte Carlo simulation for a run-time comparison of numerical methods to solve the wosHTLN problem as described in the text. The run-time of the reference solution (IHLTS) is of the same order of magnitude as the time of the “Dense-QR” case. The displayed value is normalized such that the shortest run-time equals 1.

The state of the art Householder-QR decomposition for sparse matrices described in [124] is used to solve the wosHTLS problem iteratively. Again, we use the freely available implementation provided in the SUITE SPARSEQR’s spqr. Due to both the complexity and settings of the algorithm as well as its dependency on structure and sparsity, the computational complexity of this algorithm cannot be quantified in general. However, for sufficiently large problems, the sparse QR factorization demonstrates a substantial speedup in respect to employing algorithms specifically designed for dense matrices. For the problem at hand, such as the $M$ matrix in (4.8), this is corroborated by our test (c.f. “Sparse-QR” in Fig. 4.4).

As expected, the Newton-KKT method is slower than the sparse-QR algorithm even though its superlinear convergence requires less iterations to solve the problem. The speed deficiency originates in the overhead of solving the quadratic subproblem and the feasibility check of the involved Hessian (c.f. Appendix B).

This test excludes the option of using Givens rotations for the QR factorization. To the best of our knowledge there exists no generally available algorithm for this task, which is comparable in run-time with the Householder QR. Our proof-of-concept implementation of the self-scaling fast Givens rotations [125] operates slower than the “Dense-QR” method.
4. TRAJECTORY RECOVERY

Figure 4.5: Monte Carlo result of the variable noise problem as explained in the text is displayed as a cumulative histogram of the NMSE-cost. The difference of the NMSE-cost distributions is an indicator of the benefit of the wosHTLS algorithm. The low scoring tests stem from signals violating the missing data assumptions mentioned in the text.

4.3.4 Missing Data, Variable Noise, and Other Challenging Scenarios

In the case of missing data and with decreasing number of observations, with the worst-case scenario being consecutive gaps, solving the rank estimation becomes increasingly difficult. However, even in this case, the signal can be determined correctly as long as the gaps are not masking any significant properties of the signal’s dynamics. We also note that the wosHTLN method requires the length of consecutive segments to be (magnitudes) larger than the true model order, e.g. a model of order 2 cannot be estimated if every other sample is missing. If those assumption are violated, the model as well as the reconstructed data can no longer be trusted.

Variable Noise

Scenario: Object tracking through occlusions with variable tracking quality.

The following evaluation is presenting empirical evidence that the weighted HTLN algorithm, described in 4.2.2, is suitable to solve the missing data rank estimation problem reliable in the presence of variable noise and dynamic changes; two occurrences frequently observed in object tracking through occlusions.
Figure 4.6: Examples of the indicator function effect to solve variable noise wosHTLN problems for two reconstruction problems of low-order $n^* = 3$ (left) and $n^* = 7$ (right) as explained in the text. For the sake of clarity, we only show data of the signal’s first channel. The gray scale bar on the figure’s bottom represents the value of wosHTLN’s $\omega$ where black and white are the extreme weights of $\omega_t = 1$ (”trustful”) to $\omega_t = 1$ (”missing data”), respectively. The gray shaded patches indicates the additive noise samples, $\eta_t$, where the patch’s height corresponds to twice the standard deviation used to generate the sequence $\eta_{[1:T]}$.

Test signals, $p_{[1:T]}$, of dimension $d = 2$ and $T = 150$ are generated for low-order $n$, and then evenly divided in three segments, $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_3$, by two gaps of length 30. To simulate variable tracking quality, the signal is corrupted with mean-free Gaussian white noise using a variable variance, $\sigma^2_\eta$. For the first and last segment we choose a $\sigma^2_\eta$ such that the signal-to-noise ratio equals 2, and the middle segment is corrupted with even more pronounced noise ($4\sigma^2_\eta$). The indicator function is reflecting the present knowledge of the tracking quality, and here we choose:

$$
\omega_t = \begin{cases} 
1, & \text{if } t \in \mathcal{T}_1 \cup \mathcal{T}_3 \\
\frac{1}{2}, & \text{if } t \in \mathcal{T}_2 \\
0, & \text{otherwise}
\end{cases}
$$

For the reference solution, we follow the IHTLS requirements and set $\omega_t = 1$ for $t \in \bigcup_{i=1}^{3} \mathcal{T}_i$ and otherwise it is set to zero [101], which means that IHTLS cannot distinguish between various noise levels.

In Fig. 4.5, we present the result of the Monte Carlo simulation for this scenario test. Fig. 4.6 displays two examples demonstrating the performance of the wosHTLN algorithm. It is
4. TRAJECTORY RECOVERY

apparent that the IHTLS result is corrupted by noise as the fixed noise termination procedure is selecting a model order larger than the true order. The wosHTLS method, on the other hand, is discounting the effect of the strong noise in the middle segment and selects the correct model order with its combined model order estimation and validation scheme.

Nonzero Mean Noise

Scenario: Object tracking through occlusions with drifting tracking result.

Figure 4.7: Monte Carlo result of the nonzero mean noise wosHTLN problem as explained in the text. The result is presented as in Fig. 4.5. The difference of the NMSE-cost is an indicator of the benefit of the wosHTLS algorithm.

Here, we generate the synthetic test data as before (c.f. variable noise case), but add the challenge of nonzero mean noise to model a drifting object tracker. Therefore, the true \( \eta_t \) is drawn from a Gaussian white noise distribution with time-dependent mean, \( \mu_1(t) = \frac{\kappa^2}{\delta_1}(t - 1)^2 \) for \( t \in T_1 \) and \( \mu_3(t) = \frac{\kappa^2}{\delta_3}(T - t)^2 \) for \( t \in T_3 \), respectively, where \( \kappa \) is a small constant selected uniformly random from the interval \([-0.15, 0.15]\), and \( \delta_i = \max_{t \in T_i} p_t \) relates the extent of the drift to the true signal. For all times \( t \not\in T_1 \cup T_3 \), the noise \( \eta_t \) will exhibit zero-mean, i.e. for those times the equality \( \mu(t) = 0 \) holds. To enable the wosHTLS algorithm to include knowledge about the data quality, which is here affected by the variable-mean noise, the indicator \( \omega \) must be modulated. Consequently,
4. TRAJECTORY RECOVERY

Figure 4.8: Examples of the indicator function effect to solve nonzero mean noise wosHTLN problems for two reconstruction problems of low-order \( n^* = 3 \) (left) and \( n^* = 7 \) (right) as explained in the text. The data is displayed as in Fig. 4.6.

the \( \omega \) is changing over time too, and here we artificially select \( \omega_t = \frac{1}{\mu_t(t)} \), \( \forall t \in T_1 \cup T_3 \), \( \omega_t = 1 \) for the middle segment \( T_2 \), and \( \omega_t = 0 \) for all other times. In order to gain a better understanding of the data generation, we refer the reader to Fig. 4.8, which displays two exemplifying cases.

In Fig. 4.8, we present two time series reconstructions where the wosHTLN method is using the additional information carried in the weighting function \( \omega \) to estimate \( p' \) from nonzero mean noise corrupted observations with 40% missing data. The success of the wosHTLN method can be explained by the gradual discounting of the corrupted observations through the decaying value of the indicator \( \omega \), and thereby limiting the bias introduced by the nonzero mean noise. If \( \omega \) is strictly limited to \( \{0, 1\} \) only to respectively signal missing and present data (which is the IHTLS approach), the nonzero mean error is modeled in the “cleaned” signal, \( p' \), by a model with order \( n > n^* \).

The Monte Carlo result, displayed in Fig. 4.7 shows that the wosHTLN algorithm is solving the problem at hand consistently better than the reference algorithm, which is ineffective in recognizing the variable-mean noise.

4.4 Concluding Remarks and Further Research Recommendations

In this chapter, we presented an efficient algorithm that recovers missing observations from a dynamical system of unknown order using the dynamic cues captured in only sparsely available, short sequences of noise corrupted observations. The presented method recovers the missing observations while concurrently estimating the system order (or McMillan degree) by the
4. TRAJECTORY RECOVERY

means of a Hankel matrix rank estimation problem associated with the observations. The proposed algorithm, belongs to the class of parameter estimation methods and as such is an extension of the total least norm algorithms for real-valued, Hankel-structured problems. The two key enablers are (a) the approximate solution to an overdetermined linear system of form $Ax \approx b$, while preserving the Hankel-structure of perturbations in $[A|b]$, and (b) finding the best model order which explains the data in the most efficient way by using a combined model order and verification scheme. The result of applying the wosHTLS technique is, amongst others, a best-rank approximation of both the noise and signal part of the observations.

To incorporate a priori knowledge about the probability of correctness of observations, we showed that it is necessary to define a function that indicates the importance of the associated observation in the rank estimation problem. In the applied case of our trajectory recovery, the indicator function may encode the precision of the information contained in each observation, and thus low-quality observations can be discounted by the algorithm such that they are not biasing the estimated variables.

We also investigated numerical methods to solve the combined rank estimation and observation recovery problem. Our analysis showed that the problem is best solved with either the method of weights using a sparse Householder-QR decomposition with medium-valued constraint violation weight or an iterative method solving sequential quadratic programming subproblems by the means of the conjugated gradient method. The latter method should be selected in cases where a high-quality solution is required, and the former approach is justified in situations where computational speed takes the preference over the solution quality.
Bibliography


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


Appendix A

Microscopic Crowd Simulation

In this section, we discuss one method that we used to generate data for the evaluation of our crowd anomaly framework presented in Section 2. As discussed in Sec. 2.3.1, there are many agent-based crowd simulations suitable for the offline simulation of human crowds. We selected the optimal reciprocal collision avoidance (ORCA) algorithm as it provides the means to generate numerical simulations of large, dense crowds. In the following we review the basic ORCA algorithm, describe our extensions to it, and provide some simulations that were used to test the crowd anomaly method presented in the main text.

A.1 Optimal Reciprocal Collision Avoidance Algorithm

In 2009, Berg et al. introduced ORCA, an online path planning method for mobile agents, which interact in a physical space, avoiding collisions in a reciprocal manner, as discussed below. The ORCA algorithm belongs to the class of microscopic motion models as it treats crowds as an aggregation of individuals. The improvements in [126] allow the simulation of large numbers of agents and ORCA became useful in crowd simulation, e.g., in computer graphics or virtual reality, and was used to describe crowd movement during the Tawaf [127], for example.

ORCA independently - and possibly simultaneously - determines the motion of many agents driven by some objective (e.g. moving towards a given target) while minimizing collisions with other moving agents and static obstacles. Specifically, each agent selects its velocity from a permissible set of velocities that is restricted by the presence of other agents, their observed velocities as well as obstacles that are treated as static agents [126].

1For the sake of brevity, we restrict the discussion to agent-agent interactions and refer the reader to the reference [126]
APPENDIX A. MICROSCOPIC CROWD SIMULATION

To further the discussion, we subsequently define the problem and introduce some variables. Let $M$ agents share a planar environment in $\mathbb{R}^2$. Each agent is modeled by a circular disk with radius $r$, which is set to the average of the semi axis lengths of the ellipse model used in [71], i.e. $r = 0.2025$. The position (the disc’s center) and velocity of each agent $i \in 1, \ldots, M$ is denoted by $p_i$ and $v_i$, respectively. $v_{\text{max},i}$ is the maximum permissible velocity. The preferred velocity, denoted by $\hat{v}_i$, is determined by the agent’s desired motion, objective, etc. For example, we may choose $\hat{v}_i = v_{\text{max},i}d_i$, where $d_i$ is the unit length direction of $i$-th agent’s target located at $x_{i,t}$: $\|x_{i,t} - p_i\|d_i = x_{i,t} - p_i$.

According to the ORCA method, the quantities $(p_i, v_i, r)$ may be observed by other agents, whereas all other variables can not be observed by any other agent. ORCA is a discrete method, determining the agents’ motion iteratively at discrete times $t \in \{k\triangle t\}_{k \geq 0}$, where $\triangle t$ is the time step. The ORCA algorithm has been shown to be numerically stable for time steps up to 0.2 [128] and we select $\triangle t = 0.1$. At each time step, each agent reevaluates its velocity according its surrounding, trying to prevent collisions with other agents and static obstacles. If there are multiple velocities satisfying the latter condition, the agent selects its new velocity as close as possible to its preferred velocity. More formally, the position update at each simulation step is done linearly through

$$p_i(t + \triangle t) = p_i(t) + v_i(t)\triangle t, \quad (A.1)$$

where the agent’s velocity, $v_i(t)$, is such that

$$v_i(t) \overset{\text{def}}{=} \arg \min_{v \in \text{ORCA}_i} \|v - \hat{v}_i\|. \quad (A.2a)$$

In the latter equation, $\text{ORCA}_i$ is the set of permitted velocities:

$$\text{ORCA}_i = \{v : \|v\| \leq v_{\text{max},i}\} \bigcap_{j \neq i} \text{ORCA}_{ij}, \quad (A.2b)$$

that are induced by all agents in the neighborhood of agent $i$, i.e. $j \in \mathcal{N}_i(r_{\text{max}})$, where $\mathcal{N}_i(r) \overset{\text{def}}{=} \{k : \|p_i - p_k\| < r\}, k = 1, \ldots, M$. Below, we will revisit how $\text{ORCA}_{ij}$ is formed.

To appreciate the ORCA algorithm, it is beneficial to briefly recapitulate the velocity

---

2 In this section, we use dimensional variables based on the International System of Units (SI). In particular, the units are as follows: meter for spatial variables, seconds for temporal variables. Derived variables, e.g., velocity and acceleration, derive from the basic units. For the sake of brevity, we do not show the variable units in this section.
APPENDIX A. MICROSCOPIC CROWD SIMULATION

obstacles (VO) concept, introduced in [129], which is the theoretical basis for ORCA. The VO method updates a mobile agent’s velocity through geometric calculations in velocity space and in particular through the definitions of obstacles in the velocity rather than in the physical space. For two agents \(i\) and \(j\), agent \(j\) induces a VO on \(i\), \(\text{VO}_{ij}\), and \(i\) induces a symmetric velocity obstacle on \(j\), \(\text{VO}_{ij}\). For two agents \(i\) and \(j\), the VO of agent \(j\) induced on agent \(i\) is formally

\[
\text{VO}_{ij}^\tau = \{v | \exists t \in [0, \tau] : tv \in D(p_j - p_i, r_j + r + t)\},
\]

where \(\tau > 0\) is some time horizon and \(D(p, r) = \{q : \|q - p\| < r\}\) is an open disc of radius \(r\) around location \(p\). Likewise, agent \(i\) induces a symmetric velocity obstacle on \(j\): \(\text{VO}_{ji}^\tau\). Those VOs may be interpreted as the set of velocities that would have agents \(i\) and \(j\) collide within time \(\tau\). Now, if agent \(i\)’s velocity is chosen outside of \(\text{VO}_{ij}^\tau\), agent \(i\) is guaranteed not to collide with obstacle \(j\) within time \(\tau\). The latter guarantee only holds if \(j\)’s velocity remains constant within the time horizon, which may be violated if \(j\) is another agent that is following its own collision-avoiding strategy. In this case, agents \(i\) and \(j\) would inhibit nonphysically oscillating motion trajectory to prevent their mutual collision because both agents would independently try to avoid the entire collision and thus over-correcting their motion [126]. Here, ORCA improves the VO ansatz by sharing the burden of collision avoidance in a reciprocal manner by agents \(i\) and \(j\) both sharing half the effort to prevent their collision. Assuming that agents \(i\) and \(j\) move with velocities \(v_i\) and \(v_j\), respectively, then

\[
\mathbf{u} \overset{\text{def}}{=} \left( \arg \min_{v \in \partial \text{VO}_{ij}^\tau} \|v - (v_i - v_j)\| - (v_{i,\text{opt}} - v_{j,\text{opt}}) \right) - \left( v_{i,\text{opt}} - v_{j,\text{opt}} \right) \quad (A.2c)
\]

is the minimum velocity deviation required in the relative velocity of both agents, \(v_i - v_j\), to avert the collision between themselves. In other words, \(\mathbf{u}\) is the velocity vector from the current relative velocity to the closest point on the boundary of their VO (\(\partial \text{VO}_{ij}^\tau\)). To avoid the collision each agent must at least change their velocity by \(1/2\mathbf{u}\), which implies that the total velocity deviation is larger than \(\mathbf{u}\). The permitted velocities for agent \(i\) induced by \(j\) are then given by the linear constraint

\[
\text{ORCA}_{ij} \overset{\text{def}}{=} \left\{ v : \left( v - (v_{i,\text{opt}} + \alpha \mathbf{u}) \right) \cdot \mathbf{n} \geq 0 \right\}, \quad \alpha = \frac{1}{2},
\]

(A.2d)

where \(v_{i,\text{opt}} = v_i\) is the agent’s current velocity and \(\mathbf{n}\) is the normal vector of \(\partial \text{VO}_{ij}^\tau\) at the VO’s closest point to the ORCA line. For multiple agents, (A.2b) defines \(i\)’s permitted velocities through an intersection of those constraints. In cases where the set \(\text{ORCA}_i\) is empty, i.e. no velocity exists to
APPENDIX A. MICROSCOPIC CROWD SIMULATION

prevent all collisions, a minimally violating solution to (A.2b) may be found as suggested in [126].

A.2 Extension of the ORCA Algorithm

While ORCA provides a neat theoretical framework for motion planning of multiple agents in a crowd, it does not explicitly model the characteristics of human motion, which leads to simulation results that do not reflect many of the observed behaviors of human crowds [130]. For example, the pedestrians velocities appear to be too large, motion changes are too abrupt, and basic self-organization phenomena are not reproduced accurately. Therefore, we suggest the following extensions to Berg’s ORCA algorithm and provide a qualitative comparison, showing the improved crowd behavior.

A.2.1 Human Physiology Constraint

ORCA was initially developed for unconstrained velocity changes $\Delta v_i = v_i(t + \Delta t) - v_i(t)$. However, human physiology imposes a physical constraint on motion changes [131], which are included in the model (A.2a) by limiting the acceleration $a_i \overset{\text{def}}{=} \frac{\Delta v_i}{\Delta t}$ through

$$v_i(t + \Delta t) = v_i(t) + \frac{a_{\text{max},i}}{a_i} \Delta v_i,$$

if $a_i > a_{\text{max},i}$, for some positive maximum acceleration $a_{\text{max},i}$. In our simulations we use a uniform maximum acceleration for all agents equal to 0.83 [132].

A.2.2 Asymmetric Collision Avoidance and Group Dynamics

The original ORCA method is known to produce motion solutions that are not observed in reality. In order to overcome this limitation, we extended the ORCA algorithm as follows. Firstly, we followed the suggestions in [127] to include priority and right of way, which allows agents to respond to potential collisions asymmetrically, depending on context. For example, in a commuter station situation, persons exiting a train are given priority over travelers entering the train, decreasing the exiting/boarding time. Priority is added to the ORCA model by modifying (A.2d) as follows
APPENDIX A. MICROSCOPIC CROWD SIMULATION

[127]. Let $p_i \geq 0$ denote the priority of agent $i$, then

$$p_{ij} \stackrel{\text{def}}{=} \begin{cases} \max (p_i - p_j, 1) & \text{if } p_i \geq p_j \\ 0 & \text{otherwise} \end{cases} \quad (A.3a)$$

is the relative priority of the $i$-th agent in respect to agent $j$ or in other words the right of way and (A.2d) introduces priorities through

$$v_i^{\text{opt}} = (1 - p_{ij})v_i + p_{ij} \hat{v}_i, \quad (A.3b)$$

$$\alpha = \begin{cases} \frac{1+p_{ij}}{2} & \text{if } p_{ij} \geq 0 \\ \frac{1+p_{ji}}{2} & \text{if } p_{ji} > 0 \end{cases}. \quad (A.3c)$$

The priority-based ORCA formulation is equivalent to (A.2) if both involved agents have the same priority. In case one agent is of higher priority, the right of way is increased proportionally, resulting in a decreased effort to avoid the collision and instead delegating it to the opposing agent. Furthermore, the selected velocity will be closer to the agent’s preferred velocity due to the linear interpolation in $v_i^{\text{opt}}$, above.

In our implementation, the priority in (A.3a) results from three different models such that $p_i = p_{b,i} + p_{d,i} + p_{g,i}$: (a) the base priority, $p_{b,i} \geq 0$, reflects the agent’s aggressiveness as suggested in [127]; (b) the distress priority, $p_{d,i} \geq 0$, models agents that are in distress, which may exhibit an increased desire to reduce their stress level and thus will react more aggressively, which we interpret as a higher priority [133]; and (c) the group priority, $p_{g,i} \geq 0$, giving agents traveling in a pack precedence over smaller groups or even individuals. The introduction of a group-based priority stems from the criticism that ORCA simulations do not model groups but merely treat agents as independently acting individuals. To more fully capture group dynamics [134], we extend the priority feature of agents to prioritize groups. This reproduces the observed behavior of agents traveling together for a period of time, dominating their neighborhood [134]:

$$p_{g,i} \stackrel{\text{def}}{=} \sum_{j \in N_i(r_g)} k_1 \frac{v_j \cdot v_i}{\|v_j\| \|v_i\|} e^{-k_2 \frac{(p_j - p_i)^2}{r_g^2}}, \quad (A.4)$$

where $k_1, k_2$ are parameters and $r_g$ is the group priority’s neighborhood radius. The tuning of $p_{g,i}$’s parameters was executed to reproduce the results of Helbing’s social force model [28]. In particular,
we sought the parameter triplet \((k_1, k_2, r_g)\) that results in the lane formation following (2.13). In the limit of an agent traveling alone, i.e. \(N_i(r_g) = \{i\}\), (A.4) evaluates to zero, which corresponds to the original ORCA algorithm. The effect of that model extension is seen in Fig. A.1. Whereas the original ORCA algorithm’s number of lanes scales linearly with corridor width, the present implementation exhibits the correct lane formation property, as discussed in Sec. 2.3.3.2.

### A.2.3 Density Avoidance

As discussed in Sec. 2.3.2, the motion of an individual in a crowd is influenced by other individuals. One such effect is the territorial effect which is due to the psychological reason that humans feel increasingly uncomfortable the closer they get to another person [135]. In general, this effect manifests in humans maintaining a private sphere, which means keeping some distance from other pedestrians. This distance is influenced by many factors such as travel intent, cultural norms, crowd density, and velocity. In terms of crowd simulations, this is modeled by repulsive effects of other pedestrians in the neighborhood of an individual. Helbing’s social force model includes this effect by defining a repulsive potential that leads to motion trajectories avoiding crowded areas [28]. In other models, this effect is incorporated through a term representing a density avoidance force, to simulate individuals avoiding highly packed areas and instead prefer less crowded areas. The ORCA model is insensitive to such density-related effects, which results in congested crowds, an overestimation of motion velocities [127] as well as unexpected crowd dynamics as shown in Fig. A.2. One possible solution to incorporate the repulsive force would be to modify the ORCA definition

Figure A.1: Number of lanes in bidirectional crowds for various corridor widths of Berg’s ORCA algorithm [126] (crosses) and the present ORCA implementation using group priorities (circles) and their fits following Eq. (2.13) (bold and dashed lines, respectively). The dotted line corresponds to Helbing’s social force model (2.13). All simulations use the setup described in Sec. A.3.
by using an artificially inflated agent size, i.e. increasing the agent’s radius [131]. Even though this solves the private sphere problem, the ORCA solution still lacks the property of individuals avoiding crowded areas. In the following, we describe an extension of the ORCA algorithm that models the repulsive term of the social force model by blending the individualistic nature of ORCA with spatial density awareness.

Density-related crowd effects require the definition of a local density from the microscopic ORCA model. In particular, we determine the local density \( \rho(x, t) \) at the location \( x \) as

\[
\rho(x, t) \overset{\text{def}}{=} \frac{1}{A} \sum_{i=1}^{M} e^{-\frac{\|p_i(t)-x\|^2}{R^2}} ,
\]

where we average over the circular area \( A = 4 \) (square meters) and \( R \) is determined through the relationship \( A = \pi R^2 \). In our simulations we choose \( A = 4 \), which is justified by the findings in [136]. With (A.5), we define the local pressure as

\[
p(x, t) \overset{\text{def}}{=} \frac{\rho(x, t)}{\rho(x, t) - \rho_{\text{max}}} .
\]

Using the disc model for the agents’ spatial extent and neglecting any deformations due to crowd pressure, the maximum admissible density, \( \rho_{\text{max}} \), is determined by the densest packing of circles in a plane, which is the arrangement of circles in a hexagonal lattice of the bee’s honeycomb with a packing density of \( \eta = \frac{1}{6} \pi \sqrt{3} \) [137]. Thus, \( \rho_{\text{max}} = \frac{\eta}{\pi \sigma} \), which evaluates to about 7 persons per square meter. That value compares well to the upper limit on crowd density (eight persons per square meter) reported for the most densely packed crowds at the Hajj [138].

Density avoidance is the tendency of the moving crowd to prefer motion orientation from high to low density, hence along the negative pressure gradient. Using (A.6), we can define such a motion direction as

\[
d_{p,i} \overset{\text{def}}{=} -\beta \frac{\rho(p_i)}{\rho_{\text{max}}} \nabla p|_{p=p_i} ,
\]

where the pressure gradient may be calculated over a neighborhood of \( p_i \) and \( \beta \) is some positive, scaling factor. The effect of this ORCA extension is only visible at higher densities as \( d_{p,i} \) is scaled linearly with the density.

Even though the ORCA framework does not provide for a direct implementation of density effects without sacrificing the optimality condition of collision avoidance, the preferred velocity definition allows us to include density effects efficiently. The preferred velocity is the velocity the
Figure A.2: Simulation results at times $t = 75, 100, 112.5, 125, 137.5, 150$ (top to bottom) for the original ORCA algorithm [126] (left column) and for the ORCA algorithm using the density avoidance (right column), as discussed in the text. Each disk represents one agent; green and gray disks are agents with general walking direction from left to right and right to left, respectively. The colormap encodes the total flux in $x$-direction. Both simulations share the same initial as well as boundary conditions and use the same parameter set. Employing a density avoidance results in a strongly organized crowd pattern whereas Berg’s algorithm exhibits a congestion at a density level way below the critical density where congestion should occur [71]. All simulations use the setup described in Sec. A.3.
agent would assume if no other agents interfere with it \[126\], for instance a velocity directed towards the agent’s goal with a magnitude equal to the agent’s preferred speed: $\hat{v}_i = \hat{u}_i \hat{d}_i$, where the direction $\hat{d}_i$ has unit length and the preferred speed, $\hat{u}_i$, may be a function of the density, as described in the next section. Incorporating density avoidance may be accomplished through redefining the preferred velocity: $\tilde{v}_i \leftarrow \tilde{v}_i + d_{p,i}$, where density avoidance has equal importance as motion target tracking. Furthermore, this implementation may violate the preferred speed constraint (i.e. $\|\tilde{v}_i\| = \hat{u}_i$). In our implementation, we redefine the preferred velocity as follows:

$$\tilde{v}_i \leftarrow \begin{cases} 
\hat{u}_i \frac{d_{p,i}}{d_{p,i}} & \text{if } \|d_{p,i}\| \geq \hat{u}_i \\
\tilde{v}_i + d_{p,i} & \text{if } \|\tilde{v}_i + d_{p,i}\| \leq \hat{u}_i \\
t^\star \tilde{v}_i + d_{p,i} & \text{otherwise}
\end{cases} \quad (A.7)$$

where $t^\star$ is the solution to $(t^\star \tilde{v}_i + d_{p,i})^2 = \hat{u}_i^2$. The definition (A.7) weighs the density avoidance as more important than motion target tracking while maintaining the preferred speed constraint. The effect of the discussed density avoidance is seen in the improved simulation results of a bidirectional crowd flow with the setup as discussed in Sec. A.3 in a 20 meters wide corridor as shown in Fig. A.2.

Finally, we note that the derivation of (A.7) does not violate the ORCA condition that only the external agent quantities ($p_i, v_i, r$) may be observed.

### A.2.4 Density-speed Relationship

It is well known that the speed of human crowds exhibits a relationship to its density. This fact is typically expressed in the so-called fundamental diagram relating the crowd density to the crowd flow, which is defined as density times speed. The literature on density-speed relationships is vast (e.g. [63, 71, 72, 139, 140, 141, 142]) and there is a significant spread in the reported data [143]. Until now there exists no universal law that relates pedestrian velocity to crowd density as well as the factors that can affect this relationship (age, gender, culture, travel intent, type of infrastructure, etc.). Fruin’s level-of-service concept states that a random, non-directional crowd’s speed is reduced when the density is above one person per square meter [72]. Fruin limits the significance of his result as there are several situations in which his level-of-service scheme does not apply [72]. The so-called Kladek formula, a nonlinear density-speed law proposed in [144], tries to include the travel intent and [71] provides calibrated values for pedestrian in rush hour traffic or walking for leisure. Self-organization in crowds may increase the feasible crowd flow [139]. An extreme case of this
phenomenon are soldiers marching synchronized in formation. In this type of crowd it is possible to maintain a walking speed in high density that is larger than the prediction by the Kladek formula. Typical pedestrian crowds, however, do not maintain the required precise cadence and homogeneous density that is necessary to attain high flow rates for prolonged period of times.

In our crowd simulation, we provide the means to adjust the density-speed relationship through a flexible implementation of the desired speed, \( \Vert \hat{v}_i \Vert \), as a function of density and possibly other parameters. In the following, we use the Kladek formula to determine the magnitude of the preferred velocity,

\[
\| \hat{v}_i \| = u_\infty \left[ 1 - e^{-\frac{\gamma}{\rho_{\text{max}}} \left( \frac{1}{\rho(p_i)} - \frac{1}{\rho_{\text{max}}} \right)} \right], \tag{A.8}
\]

where we select the free-stream velocity for “rush hour” pedestrians [71], i.e. \( u_\infty = 1.69 \) and \( \gamma = 0.15 \), and \( \rho(p_i) \) is the local density at the agent’s position \( p_i \) as defined in (A.5). The fitting parameter \( \gamma \) was selected by calibrating the average, absolute speed of the agents in our simulation to the fundamental diagram for “rush hour” in [71] as seen in Fig. A.3 (right). Berg’s ORCA algorithm exhibits the fundamental density-speed relationship shown in Fig. A.3 (left) and the deviation from the Kladek law stems from the fact that \( \| \hat{v}_i \| \) is independent of the density and in fact set to a constant value.
Table A.1: Parameters of the ORCA crowd motion model used in the simulation of crowds using the ORCA crowd model described in the text.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radius</td>
<td>0.2025</td>
<td>[71]</td>
</tr>
<tr>
<td>Intended free-stream velocity $v_{\text{max},i}$</td>
<td>$\mathcal{N}(1.56, 0.17)$</td>
<td>[71]</td>
</tr>
<tr>
<td>Maximum acceleration $a_{\text{max},i}$</td>
<td>0.83</td>
<td>[132]</td>
</tr>
<tr>
<td>Maximum density $\rho_{\text{max}}$</td>
<td>7</td>
<td>[137, 138]</td>
</tr>
<tr>
<td>Density-speed relationship $u_{\infty}$</td>
<td>1.69, γ = 0.15</td>
<td></td>
</tr>
<tr>
<td>Priority $p_{b,i}$</td>
<td>$\mathcal{N}(0.5, 0.25)$</td>
<td></td>
</tr>
<tr>
<td>Group priority $k_1, k_2$</td>
<td>0.13, $r_g = 2$</td>
<td></td>
</tr>
<tr>
<td>Density avoidance $A, \beta$</td>
<td>4, 0.4</td>
<td>[136]</td>
</tr>
</tbody>
</table>

**ORCA:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time step $\Delta t$</td>
<td>0.1</td>
<td>[128]</td>
</tr>
<tr>
<td>Time horizon for agents $\tau$</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Time horizon for static obstacles $\tau$</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

### A.3 Simulations

In this section, we describe the setup of the crowd simulations with motion planning using the above discussed modified ORCA algorithm with the parameter set listed in Table A.1.

We simulate a crowd in a two-dimensional physical space $\Omega_p \subseteq \mathbb{R}^2$, which may feature areas that are unaccessible to agents, e.g., spaces bounded by walls (static obstacles), and at the physical space’s boundaries, we define line segments, where agents are allowed to enter and exit. The crowd consists of agents with the characteristics listed in Table A.1 that are traveling from their entry position to a designated exit and all agents sharing the same entry-exit combination may be seen as one sub-population, as defined in Sec. 2.3.2. When a new agent is introduced to the simulation, we assign it to a sub-population and thus implicitly a target exit.

To steer the agents towards their targets, we define a static path vector field $d(x)$ for all accessible locations $x \in \Omega_p$. In order to accommodate for multidirectional crowds, we define one $d$-field for each entry-exit combination. Those path fields may be found by any pathfinding strategy that takes a starting point as well as a target and then finds a series of points that together comprise a path to the target. Intermediate waypoints may be introduced to define more complex path geometries [21]. In our simulations, we employ the $A^*$ algorithm to accomplish the path finding task [145].

No crowd motion is perfectly deterministic and so we impose a small random noise on the preferred velocity. This also aids the simulation by preventing jamming in perfectly symmetric

---

*Here, $\mathcal{N}(\mu, \sigma)$ denotes a normal distribution with mean $\mu$ and standard deviation $\sigma$.}
solutions. The noise is additive in the velocity expressed in polar coordinates $(\|\hat{\mathbf{v}}_i\|, \angle \hat{\mathbf{v}}_i)$:

$$(\|\hat{\mathbf{v}}_i\|, \angle \hat{\mathbf{v}}_i) \leftarrow ((1 + E_i \zeta_v)\|\hat{\mathbf{v}}_i\|, \angle \hat{\mathbf{v}}_i + E_i \zeta_\alpha),$$

where $\zeta_v$ and $\zeta_\alpha$ are random samples from a normal distribution. The noise component is a function of the motion efficiency, $0 \leq E \leq 1$, which we define for the $i$-th agent as

$$E_i \overset{def}{=} \max \left( 0, 1 - \frac{\|\mathbf{v}_i\|}{\|\hat{\mathbf{v}}_i\|} \right) \left( 1 - \frac{\rho_i(\mathbf{p}_i)}{\rho_{\text{max}}} \right).$$

This models the observation of vanishing noise at both saturated density and motion with the preferred speed. When modeling human crowds, we modify the noise term with a low-pass filter, such as a first order lag term, as humans do not exhibit Brownian noise in walking.

All simulations start at time $t = 0$ and then advance in discrete time steps $\Delta t$. At each time step, we randomly spawn new agents at the entries, remove those agents that are leaving through the defined exits, calculate the preferred velocities, and update the position of all agents with (A.1).

**Measurements**

In order to generate data for the evaluation of our crowd anomaly framework, we need to define the measurements (2.3.2) required by the anomaly algorithm (c.f. Sec. 2.3.2). In particular, those are the states $\rho_i$ and $u_i$ for each sub-population $i$. Following the discussion in Sec. 2.3.2, we use a preassigned grid of points, denoted by $\Omega$, that discretizes the entire surveillance area and define all states as well as local quantities on those points.

According to our crowd model, we define a sub-population as all agents sharing similar parameter sets as used in (2.5). This is motivated by the fact that when entities belonging to the same group are acting in a coordinated behavior influenced by collective decisions rather than individual goals, those entities exhibit similar behavior and are acting as one crowd. In particular, this means that two agents belonging to the same sub-population share the same travel intent, target direction, etc. For example, in a bidirectional crowd we define two sub-populations and assign the agents to their sub-population according to the travel direction. More formally, we say that agent $j$ belongs to the $i$-th sub-population if $j \in J_i$. This association is unique and has zero uncertainty in case we would have omniscient knowledge, which is the case in the above discussed simulations, where we assign agents at spawn time to their sub-population. However, in reality the agent’s sub-population association may be unknown and must be estimated.
APPENDIX A. MICROSCOPIC CROWD SIMULATION

Figure A.4: Examples of microscopic simulations. Various types of crowds (top row, from left to right): unstructured, low density, high density bidirectional crowd, jammed crowd. Different layouts (top row, from left to right): intersection, passage, room with single exit. The simulations are presented as in Fig. A.2.

In our implementation, we associate an agent $j$ to sub-population $i$ by using the agents’ location and velocity information. In particular, we map the agent’s normalized velocity onto all unit-length target vectors at its location $p_i$, which results in the distributed association weights

$$w_{ij} \overset{def}{=} \max \left( 0, \frac{d_i(p_j) \cdot v_j}{\|v_j\|} \right)$$

that are normalized over all $N$ sub-populations s.t. $\sum_{i=1}^{N} w_{ij} = 1$. We then define the sub-population association by $J_i \overset{def}{=} \{ j : w_{ij} > 0 \}$. To reduce the effect of noise, the association calculation may be filtered by computing it over a short time horizon.

Following the discussion in Sec. 2.3.2, we use a preassigned grid of points, denoted by $\Omega$, that discretizes the entire surveillance area and define the local densities and velocities at $x \in \Omega$ by

$$\rho(x, t) \overset{def}{=} \frac{1}{A} \sum_{j \in J_i} w_{ij}(t) e^{-\frac{||p_j(t) - x||^2}{R^2}}, \quad (A.9a)$$

$$u(x, t) \overset{def}{=} \frac{\sum_{j \in J_i} w_{ij}(t) v_j(t) e^{-\frac{||p_j(t) - x||^2}{R^2}}}{\sum_{j \in J_i} w_{ij}(t) e^{-\frac{||p_j(t) - x||^2}{R^2}}}, \quad (A.9b)$$

respectively, where we employ the same smoothing as used in (A.5).
Appendix B

Equality Constrained Least Squares

This appendix reviews the theory and solution of a variant of the ordinary least squares problem. The linear least squares problem with additional equality constraints on the decision variable $x \in \mathbb{R}^n$ is written as

$$x^* \overset{\text{def}}{=} \arg \min_x \|Ax - b\|_2 \quad \text{s.t.} \quad Bx = d,$$

where $A \in \mathbb{R}^{(m \times n)}$, $b \in \mathbb{R}^m$, $B \in \mathbb{R}^{(l \times n)}$, $d \in \mathbb{R}^l$, and the dimensions are restricted by $m \geq n$, and $n \geq l$. Those problems types are of general concern as they frequently arise in applications of, for example, curve fitting, nonlinear optimization, and statistical or physical modeling.

A solution to (B.1) exists if the equality constraints are consistent, i.e. $\text{rank}(B) = l$. This solution is unique if $A$’s and $B$’s nullspaces intersect trivially. The latter condition is equivalent to the augmented matrix $[B^T \ A^T]$ having rank $n$. In the case of non-unique solutions, a unique minimum-norm solution can be specified by adding the postulation $\min \|x\|_2$ to the problem (B.1).

The methods for solving the constrained least squares are either of direct or iterative nature. Direct methods include, for example, the direct elimination method, the nullspace method, and the method of weights. The conjugate gradient least squares algorithm provides an iterative solution of (B.1).

In the following, we revisit three solution methods: an exact one using the singular value decomposition (Sec. B.1), an approximate one employing the method of weights (Sec. B.2), and the iterative conjugate gradient least squares algorithm (Sec. B.3).
APPENDIX B. EQUALITY CONSTRAINED LEAST SQUARES

B.1 Exact Solution

A reference solution of (B.1), \( x_{\text{LSE}} \), can be found with the generalized singular value decomposition (GSVD) of \( A \) and \( B \) [146, 147]. Van Loan showed in [146] that \( A \) and \( B \) can be simultaneously diagonalized by orthogonal square matrices \( U = [u_1, \ldots, u_m] \), \( V = [v_1, \ldots, v_l] \) and non-singular square matrix \( X = [x_1, \ldots, x_n] \), such that

\[
U^T A X = D_A, \quad V^T B X = D_B,
\]

where \( D_A \) and \( D_B \) are diagonal matrices with entries \( \{\alpha_i\}_{i=1}^n, \alpha_i > 0 \forall i > l \) and strictly positive \( \{\beta_i\}_{i=1}^l \), respectively, and the ordering conditions, \( \alpha_i \leq \alpha_j, j > i \) and \( \beta_i \geq \beta_j, j > i \), hold.

Now, since the norm of a vector is invariant to orthogonal transformations, we can rewrite (B.1) as

\[
\min_x \|U^T A x - U^T b\|_2 \quad \text{s.t.} \quad V^T B x = V^T d,
\]

and employ \( A \) and \( B \)’s GSVD \((U^T A = D_A X^T, V^T B = D_B X^T)\) to find

\[
\min_x \|D_A X^T x - U^T b\|_2 \quad \text{s.t.} \quad D_B X^T x = V^T d.
\]

Now, it is readily apparent that the solution is \( x_{\text{LSE}} \triangleq X^{-T} y_{\text{LSE}} \), where

\[
y_{\text{LSE}} \triangleq \left( \frac{v_1^T d}{\beta_1}, \ldots, \frac{v_l^T d}{\beta_l}, \frac{u_{l+1}^T b}{\alpha_{l+1}}, \ldots, \frac{u_n^T b}{\alpha_n} \right).
\]

B.2 Approximate Solution

An approximate solution to the problem at hand (B.1) can be found by employing the method of weights which transforms the optimization into an unconstrained problem by including the equality constraints in the cost function [110, 111]:

\[
x_\pi \triangleq \min_x \left\| M x - \begin{pmatrix} \pi d \\ b \end{pmatrix} \right\|_2, \quad M \triangleq \begin{pmatrix} \pi B \\ A \end{pmatrix},
\]

where \( \pi > 0 \) is some large constant penalizing constraint violations. The identity \( \lim_{\pi \to \infty} x_\pi \to x_{\text{LSE}} \) was proven in [111]. In other words, only for \( \pi \gg 1 \) we can guarantee that \( x_\pi \) is a feasible approximation of the optimal least squares solution. However, a very large penalization constant, \( \pi \),
yields stiff, ill-conditioned matrices and thus inaccurate results when solving (B.2) numerically. For example, a solution via the normal equations of (B.2),

\[ (\pi^2 B^T B + A^T A)x = \pi^2 B^T d + A^T b. \]

will be erroneous when the first term on the left-hand side is dominating, which is most certainly the case for very large \( \pi \). However, by applying carefully chosen methods this ill-conditioning is overcome.

One such algorithm is by the means of the QR decomposition

\[ \begin{pmatrix} \pi \mathbf{B} \\ \mathbf{A} \end{pmatrix} = QR = \begin{pmatrix} Q_{\pi,1} & Q_{\pi,2} \end{pmatrix} \begin{pmatrix} R_{\pi} \\ 0 \end{pmatrix}, \tag{B.3} \]

and the solution of the upper triangular system

\[ R_{\pi} x_{\pi} = Q_{\pi,1} \begin{pmatrix} \pi d \\ b \end{pmatrix}. \tag{B.4} \]

Accurate solvers for equations of type (B.4) are readily available and sparse problems can be handled computationally efficient.

There exist several methods for calculating the QR decomposition (B.3), such as the Gram-Schmidt algorithm, Householder transformations [148] or Givens rotations [149]. Note that the Gram-Schmidt algorithm is not applicable here when the column vectors of \( M \) are (nearly) linearly dependent.

The Householder-QR factorization is the quasi-standard for the decomposition (B.3). However, for matrices with widely varying row-norms, such as \( M \) in the problem at hand, the Householder method must follow the specifications given in [109]. This is most importantly, that \( M \)'s rows are sorted in decreasing row-norm before the factorization is applied [75]. The effect of this prepended sorting is to minimize numerical errors due to rounding effect in the intermediate stages of the Householder algorithm. In addition, this allows us to employ standard linear algebra packages such as LAPACK’s DGEQP3 routine [150].

Lastly, we mention the use of Givens rotations to perform the QR decomposition, which, while being analogous to using Householder reflections, is not susceptible to row ordering and varying row-norms [110]. This algorithm is advantageous in cases where the Householder factorization is
APPENDIX B. EQUALITY CONSTRAINED LEAST SQUARES

numerically unstable. However, this benefit comes with about twice the computational costs. This
disadvantage is overcome with the fast Givens rotations introduced by Gentleman [151], and Anda et
al. decreased the computational disadvantage even further [125]. Even though extensive research
into the performance of Givens rotation-based algorithms was conducted [152, 153, 154, 155], the
Householder-QR remains faster. Therefore, we do not consider Givens rotation methods for the task
at hand.

B.3 Iterative Solution

For the iterative solution, we now consider the modified least squares problem,

\[ x_{NL} \overset{\text{def}}{=} \min_x f(x) \quad \text{s.t.} \quad r(x) = 0, \quad (B.5) \]

with both nonlinear objective function, \( f : \mathbb{R}^n \to \mathbb{R} \), and equality constraints, \( r \in \mathbb{R}^l, l \leq n \).

Eq. (B.1) is readily recovered by setting \( f = Ax - b \) and \( r = Bx - d \).

In solving (B.5) iteratively, a sequence \( \{x_1, x_2, \ldots\} \) is generated by the formula \( x_{k+1} = x_k + \alpha \Delta x \), where the vector \( \Delta x \), the search direction, is modified by the scalar step length \( \alpha \).

This method is only applicable if an initial point, \( x_1 \), can be provided; for example, by solving
the constraint equations, i.e. \( \{x_1 : r(x_1) = 0\} \) and selecting an \( x_1 \) close to the (unknown) optimal
solution. In the limit, the solution sequence converges for given initial condition towards an optimal
solution of (B.5).

One method for solving (B.5) is derived by applying the Newton-Raphson method to the
problem’s Karush-Kuhn-Tucker (KKT) conditions, which are the first order necessary conditions for
a solution in nonlinear programming to be optimal [156], namely

\[ F_e \overset{\text{def}}{=} \begin{pmatrix} g(x) - J^T \lambda \\ r(x) \end{pmatrix} = 0, \]

where \( g(x) = \nabla_x f(x) \) is \( f \)'s gradient, and \( J \in \mathbb{R}^{(l \times n)} \) is the Jacobian of the constraints, which has
the \( i \)-th row \( \nabla_x r_i(x)^T \), the gradient of the \( i \)-th constraint function, \( r_i(x) \).

The fundamental method is stated in terms of the Lagrangian function to combine the
objective function and constraints [157], \( \mathcal{L}(x, \lambda) = f(x) - \lambda^T r(x) \), where \( \lambda \) is the \( l \)-length vector
containing the Lagrange multipliers. Assuming that both \( f \) and \( r \) are twice continuously differentiable,
we approximate the Lagrangian with a second order Taylor series expansion around the point \((x, \lambda)\).
APPENDIX B. EQUALITY CONSTRAINED LEAST SQUARES

Then, we apply the stationarity KKT condition, which results in

\[ 0 = \nabla_x L(x + \Delta x, \lambda + \Delta \lambda) = H(x, \lambda)\Delta x + g(x) - J^T (\lambda + \Delta \lambda), \]  

(B.6a)

where \( H(x, \lambda) = \nabla^2_x f(x) - \sum_{i=1}^{l} \lambda^T \nabla^2_x r_i(x) \) is the Hessian of the Lagrangian with respect to \( x \).

The second KKT condition requires primal feasibility, i.e. \( r(x) = 0 \) or in linearized form:

\[ 0 = r(x) + J^T \Delta x. \]  

(B.6b)

This means that the application of Newton’s method to the necessary conditions requires the solution of the following symmetric KKT system:

\[
\begin{bmatrix}
  H & J^T \\
  J & 0
\end{bmatrix}
\begin{bmatrix}
  \Delta x \\
  -\Delta \lambda
\end{bmatrix} = -F_c,
\]

(B.7)

which is equivalent to solving

\[
\begin{align*}
\min_{\Delta x} \frac{1}{2} \Delta_x^T H(x) \Delta x + g(x), & \quad \text{s.t. } r(x) = -J^T \Delta x, \quad \text{(B.8a)} \\
\min_{\Delta \lambda} \| J^T \Delta \lambda + H(x) \Delta x + g(x) + J^T \lambda \|_2, & \quad \text{(B.8b)}
\end{align*}
\]

Here, we use the conjugate gradient (CG) approach to solve the quadratic program subproblem (B.8a), which can be executed by computing a basis for the nullspace of the constraint matrix, \( J \), to eliminate the constrains, and then applying the CG optimization to the reduced problem. In the case that \( H(x) \) is positive definite in the nullspace of \( J \), the subproblem solution will be unique. Due to the fact that the explicit calculation of the nullspace of a large sparse matrix is prohibitively expensive, we employ the preconditioned CG with residual update suggested in [158]. The second subproblem (B.8b) is simply solved for \( \Delta \lambda \) in a least squares sense.

The achievement of the above derivation is an iterative algorithm to solve (B.5), which converges superlinearly\(^1\) in the variable \( x \) [159]. Starting at a feasible point \( x_1 \), and thus \( \lambda_1 = 0 \) and for the following \( k \) iterations until convergence, the KKT subproblem (B.7) must be solved for the

---

\(^1\)Removing the constraints’ curvature contribution from the Hessian, \( H \), would result in a linear convergence of \( x \) towards its optimum.
APPENDIX B. EQUALITY CONSTRAINED LEAST SQUARES

tuple \((\Delta x, \Delta \lambda)\), and an improved solution estimate is constructed using the step

\[
\begin{align*}
x_{k+1} &= x_k + \alpha \Delta x, \\
\lambda_{k+1} &= \lambda_k + \alpha \Delta \lambda.
\end{align*}
\]

The step length can be fixed to \(\alpha = 1\) or may be dependent on the subproblem. For brevity, we refer the interested reader to the detailed discussion of this issue in [159] or any technical reports, e.g., [160, 161, 162, 163].

By definition, the solution point, \(x_{\text{NL}}\), must satisfy the KKT conditions for a local minimum. However, it may not be possible to satisfy the KKT conditions exactly because of the limited precision in the computed values. Instead, the algorithm stops at a feasible solution, \(x_k \approx x_{\text{NL}}\), when \(\|\alpha \Delta x\|_2 < \delta_x\) and \(\|\alpha \Delta \lambda\|_2 < \delta_\lambda\), for given positive tolerances \((\delta_x, \delta_\lambda)\). Other convergence criteria can be added if necessary and according to need.
List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANchOVy</td>
<td>“Annotation Of Objects In Videos”. A software toolbox for image and video labeling [103].</td>
</tr>
<tr>
<td>CCTV</td>
<td>Closed-circuit Television.</td>
</tr>
<tr>
<td>CG</td>
<td>Conjugate Gradient.</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface.</td>
</tr>
<tr>
<td>HTLN</td>
<td>Hankel Total Least Norm. Iterative method for the structure-preserving solution of an overdetermined linear system [164].</td>
</tr>
<tr>
<td>IHTLS</td>
<td>Iterative Hankel Total Least Squares. A method to estimate the rank of incomplete, noisy Hankel matrices [101].</td>
</tr>
<tr>
<td>KKT</td>
<td>Karush-Kuhn-Tucker. First order necessary conditions for a solution in nonlinear programming to be optimal.</td>
</tr>
<tr>
<td>LIC</td>
<td>Line Integral Convolution. Line integral convolution is a technique to visualize fluid motion.</td>
</tr>
<tr>
<td>NSE</td>
<td>Navier-Stokes Equations. The Navier-Stokes equations govern the motion of fluids.</td>
</tr>
<tr>
<td>OpenFOAM</td>
<td>Field Operation And Manipulation. An open source <em>computational fluid dynamics</em> software package by OpenCFD Ltd to solve complex fluid flows [69].</td>
</tr>
<tr>
<td>OF</td>
<td>Optical Flow. Dense field of instantaneous displacements (or velocities) of pixels (or small patches) in consecutive video frames [12, 13].</td>
</tr>
<tr>
<td>ORCA</td>
<td>Optimal Reciprocal Collision Avoidance. An algorithm solving the reciprocal (n)-body collision avoidance problem.</td>
</tr>
</tbody>
</table>
**PDE** Partial Differential Equation.

**ROC** Receiver Operating Characteristic. The receiver operating characteristic curve, a graphical illustration of a binary classifier system’s performance as its discrimination threshold is varied.

**SSI** Sensitive Security Information. Sensitive security information is a category of sensitive but unclassified information under the United States Department of Homeland Security’s information sharing and management directives.

**ViPER** Video Performance Evaluation Resource. A software for ground truth generation and visualization of video analysis results [48].

**VO** Velocity Obstacle. In motion planning, a velocity obstacle is the set of all velocities that would have two agents collide at some moment in time, assuming that the other robot maintains its current velocity [129].

**wosHTLN** Weighted Order-selective Hankel Total Least Norm.
Index

Annotation labels, 59
Annotation Of Objects In Videos, 58
Anomaly detection, 10
Boundary conditions, 28
Bounding box, 60
Circulant tracker, 62
Cleveland Hopkins International Airport, 68
Continuum assumption, 18, 19
Convection term, 22
Crowd anomaly, 13, 44
Crowd anomaly detection, 5, 45
Crowd density, 114
Crowd disasters, 11
Crowd mass dynamics, 21
Crowd model, 4, 14, 26
Crowd model parameter estimation, 35
Crowd momentum dynamics, 22
Crowd motion, 13, 19
Crowd motion dynamics, 21
Crowd motion model parameters, 35
Crowd motion state, 16, 20, 119
Crowd pressure, 20
Crowd self-organization phenomena, 13, 31
Crowd sensing, 3
Crowd simulation, 4
Crowd sub-populations, 18
Crowd surveillance, 11
Density avoidance, 113
Density-speed relationship, 24, 116
Disc model, 114
Discretization, spatial, 26, 28
Discretization, temporal, 27
Dynamic crowd viscosity, 22
Extraneous forces term, 23
Fundamental density-flow relationship, 117
Group dynamics, 111
Heuristic anomaly rules, 46
Kladek law, 116
Knudsen number, 30
Label framework, 58
Lane formation, 113
Large-scale crowd anomaly analysis, 14
Local density, 120
Local velocity, 120
Macro-scale crowd model, 17
Mass flux, 27
Maximum admissible density, 29, 114
Meso-scale crowd model, 17
Micro-scale crowd model, 17
Microscopic crowd model, 108
Motion dynamics estimation, 63
Navier-Stokes-inspired crowd model, 18
Observed domain, 19, 29
Observed velocity, 20
OpenFOAM, 25
Optimal Reciprocal Collision Avoidance, 108
Peak-to-sidelobe ratio, 63
Preferred velocity, 24, 109, 116
Pressure term, 23
Priority, 111
Resource allocation, 13
Right of way, 111
Social force model, 32, 112
Spatial annotation, 60
State smoothing, 40
Stochastic motion term, 24
Stochastic parameter variation models, 15
Sub-population density field, 19
Sub-population velocity vector field, 19
Total density field, 20
Tracklets, 63
Velocity obstacles, 110
Video labeling, 7, 56
Viscosity term, 22