Universiteit Utrecht

The Impact of Density Measurement on the Fundamental Diagram

Marijn van der Zwan

Master Thesis (ICA-3401928)

Supervisor: dr. R. J. Geraerts

Technical Artificial Intelligence
University of Utrecht, the Netherlands

January 7, 2016

Abstract

With the increasing demand for crowd simulations, validation and verification of the simulation becomes more important. As simulations often concern the safety of real persons it is critical that the simulated behavior is validated to be comparable to behavior of real people. Validation is often done by comparison to the pedestrian fundamental diagrams. These diagrams capture the relation between speed, density and flow in a crowd. However, many conflicting versions of these diagrams exist. Furthermore, the methods used to measure density and flow differ between papers.

In this work we take a critical look at existing density measurement methods. We compare the classic method, a Gaussian-based, and a Voronoi-based method with one another in multiple scenarios. Results show that each of these methods has different strengths and weaknesses, depending on the environment. By using video data from a real crowd to create fundamental diagrams, we show that choice for a measurement metric has a large impact on the resulting diagram. Results indicate that a measurement metric should be chosen carefully, as it directly influences reliability of the validation.

This project was conducted as part of a collaboration between Utrecht University and Movares Nederland B.V. and has been supported by the COMMIT/ project (http://www.commit-nl.nl/).

Contents

1 Introduction

As the world population grows, demand for crowd simulations increases. Events draw more and more visitors, leading to an increased focus on the safety of visiting crowds. Tragedies such as the Love Parade disaster in 2010, and the long history of incidents during the Hajj pilgrimage in Mecca underline the risk involved when a large amount of people navigate in limited space. Still [1] shows a large amount of crowd disasters can be attributed to mistakes during the design phase of an event. For practical and ethical reasons, it is often not possible to test evacuation and panic scenarios in real life.

Crowd simulations are a way to help predict the behavior of people in large groups. Crowd simulations can help understand the complexity of a situation, without having to spend a large amount of money recreating the exact scenario. Furthermore, simulations allow for quick experimentation with dangerous scenarios, without endangering the lives of people.

However, simulations are based on assumptions about the behavior of people under certain conditions. It is necessary to validate that the outcome of a simulation accurately predicts behavior in the real situation. This can be done by direct comparison to data from a similar situation. When this data is not available, characteristics of the simulated crowd can be compared to characteristics recognized in real crowds. One such characteristic is the observed relation in a crowd between speed, density, and flow. The higher the amount of people in a crowd, the lower the average speed of these people will be. Similar dependencies are known for *speed-flow* and *flow-density*. These relations are captured in a set of three diagrams known as *Fundamental Diagrams*, first described by Greenshields in [2].

1.1 Project Goals

In practice, fundamental diagrams are often used to validate simulation results. They provide a verifiable and intuitive way of validation. Many different versions of the diagrams exist however, and some of the research disagrees with each other. The form of the diagrams is dependent on multiple variables, such as the type of environment, nature of the crowd, and measurement technique used for density and flow. Density measurement is well defined for one-dimensional situations, but in two-dimensional situations multiple methods are used in practice. In this thesis we explore the problems that occur when validating simulations using fundamental diagrams, and analyze the impact of different measurement metrics.

In this thesis we will look at the different density measurement methods in detail, and perform a comparative analysis of the strengths and weaknesses of each method. Finally, we will show that the choice for a density metric has an impact on the resulting fundamental diagram.

1.2 Thesis Outline

In Section 2, we give an overview of validation and density measurement methods. We first look at relevant validation techniques for crowd simulations and analyze the problems with these approaches in Section 2.1. Fundamental diagrams rely on accurate measurements of density and flow. In Section 2.2, we look at problems that occur when measuring density in a two-dimensional environment, what requirements a suitable method should meet, and summarize relevant measurement metrics from literature. Finally, we review Voronoi diagrams in Section 2.3, and summarize literature on retrieving crowd data from videos in Section 2.4.

In Section 3, we analyze the problems with validation, based on related work. First, we describe the reasons for validating on a crowd, rather than an individual level, in Section 3.1. In Section 3.2, we then discuss metrics that allow for a simulation to be validated on a crowd level.

In Section 4, we analyze one of these validation metrics: density. We describe the density measurement methods, summarized in Section 2.2, in more detail. The advantages and disadvantages of the Classic, Voronoi and Gaussian methods are discussed.

In Section 5, we describe our comparative analysis of the density measurement methods discussed in the previous section. Each method is compared on the requirements set up in Section 2.2.1. We try to find the strengths and weaknesses of the methods in different environments.

In Section 6 we create a fundamental diagram from video data. We convert video footage of a crowd to trajectories, and create a *speed-density* fundamental diagram, with each of the density metrics analyzed in Sections 4 and 5. We then describe how these fundamental diagrams would be used to validate a crowd simulation of the real situation.

In Section 7, we summarize and conclude our findings. In Section 7.3, we discuss our ideas for future work, as well as some of the problems we encountered. Finally, we summarize a crowd simulation case study we performed for the Grand Départ of the Tour de France as part of this thesis in Appendix A.

2 Related Work

In this section, we summarize our literature review on validation and density measurement in Sections 2.1 and 2.2, respectively. In Section 2.3, we summarize Voronoi diagrams, which form the basis of one of the density measurement methods further described in Section 4, and the Explicit Corridor Map method, used for performing the crowd simulations in Appendix A. We review methods of retrieving data from a video of a real crowd in Section 2.4, since validating crowd simulations is often done with videos of real crowds.

2.1 Validation

One of the main applications of crowd simulations is to improve crowd safety during large-scale events, such as concerts or sports matches. A simulation can be used to test how an event entrance handles the expected amount of people, or whether a planned evacuation strategy can be executed safely. Since the lives of many people can depend on the outcome, it is important to ensure that a simulated scenario is both quantitatively and qualitatively similar to the real-world situation.

In the crowd simulation field, however, validation is complicated. Crowd behavior is dependent on the environment, the crowd size, and the crowd demographic. Because of these complications, it is often only feasible to test a simulation technique on a subset of the possible scenarios. An algorithm is developed and tested on a number of scenarios, generally in simplified real-world environments. These environments are then used to showcase the performance of a method in a situation that would normally not occur in real life. Examples include two large groups trying to pass through a single door, or a group of people trying to move to the opposite side of a circle. In other fields, such as research focused on fire or ship evacuations [3, 4], more focus is put on validation. Emergency evacuation plans, for example, have to fulfill strict criteria, set by the responsible authorities, to ensure that a plan is valid in the real world.

Since simulation results are often only available as a closed product, without underlying models, it is tempting to visually compare the quality of the results to those of other models. Alternatively, you would have to implement your own version of an algorithm, the environment, make the right assumptions and guess the optimal parameter settings. It is often easier to visually compare results. When humans judge a crowd simulation they tend to look for patterns such as congestion, egress rate out of a building, or remarkable behavior. This can lead to a subjective and incomplete comparison.

Humans tend to focus on individual behavior rather than crowd movement, making it hard to identify unnatural behavior. It is easy to notice characters bumping into walls or obstacles in small scenarios. In larger-scale environments the focus is often more on the crowd flow than on individual characters. A real-life crowd will have some amount of unpredictable behavior in it, with people running into each other when not paying attention, or randomly stopping when they happen to drop their keys. Not everyone will take the most efficient path to their goal. Some people may prefer to avoid busy routes by taking a detour.

When a crowd is too organized and structured it can look unnatural and robotic. By validating a simulation with data of real crowds it can be tuned to show the right amount of chaos without having to forcefully create it.

We can classify most existing validation methods in the following categories:

- **Fundamental diagram:** Compares the simulation to the fundamental diagrams of pedestrian flow.
- **Comparing:** Matches the output of a simulation to real data of the same scenario.
- **Prediction:** Tries to understand behavior found in real data to see if it can predict behavior present in simulations.
- **Standardized testing:** Evaluates the quality of a simulation on a number of standardized test cases.

In the Sections 2.1.1 through 2.1.4 we will describe research done in these categories and discuss benefits and drawbacks of the methods. In Section 2.1.5, we look at the validation methods that crowd simulation frameworks on the market used to validate their results. Finally, we conclude and summarize our findings in Section 2.1.6.

2.1.1 Fundamental Diagram

Fundamental diagrams are a concept originating from traffic flow theory [2]. It captures the characteristics of pedestrian movement in one-dimensional traffic. Fundamental diagrams are a set of three graphs representing the relations *flow-density*, *speed-density*, and *velocity-flow*. In these diagrams, flow is measured as the number of cars passing a certain point per second. Density is measured as the number of cars per meter of road, and velocity as distance in meters traveled per second. Many experiments, for example by Greenshields [2], have been done that show a relation between these metrics. By assuming the fundamental diagram is correct, one can find out all three values from the measurements of a single metric.

The same concept can be observed in pedestrian dynamics. Various studies [5, 6, 7, 8] recognized the relation between pedestrian flow, density and velocity. The higher the amount of pedestrians in a certain area, the lower their average velocity, and the lower the flow rate through that area. An example of a fundamental diagram is shown in Figure 1.

Fundamental diagrams exist for different scenarios, such as one-directional hallways, bi-directional hallways, T-junctions, and crossings. However, these diagrams do not always agree with each other, as observed by Zhang [9]. This difference cannot be fully explained by cultural differences alone. Zhang [9] conducted experiments that compare fundamental diagrams to real-world data of people moving through varying environments.

In practice, fundamental diagrams can be used to rate the capacity of pedestrian walkways and escape routes. The relationship between density and safety of movement has been captured as *Levels of Service* by Fruin [10], see Table 1. The Level of Service of an escape route is rated from A to F, depending on the maximally supported density.

While flow is relatively easy to measure in one-dimensional situations, it is less straightforward in two-dimensional situations. For this reason, crowds are mostly validated with the *density-speed* diagram. Fundamental diagrams have been used to validate simulations in [11], and were used as the basis of a simulation in [12] and [13].

	LOS	Personal space	Density
🟪	A	over 3.3 m^2	$<$ 0.30/m^2
🟦	B	2.3 - 3.3 m^2	0.30 - 0.44/m^2
🟩	C	1.4 - 2.3 m^2	0.44 - 0.71/m^2
🟨	D	0.93 - 1.4 m^2	0.71 - 1.07/m^2
🟧	E	0.46 - 0.93 m^2	1.07 - 2.17/m^2
🟥	F	under 0.46 m^2	$>$ 2.17/m^2

Table 1: Fruin Levels of Service

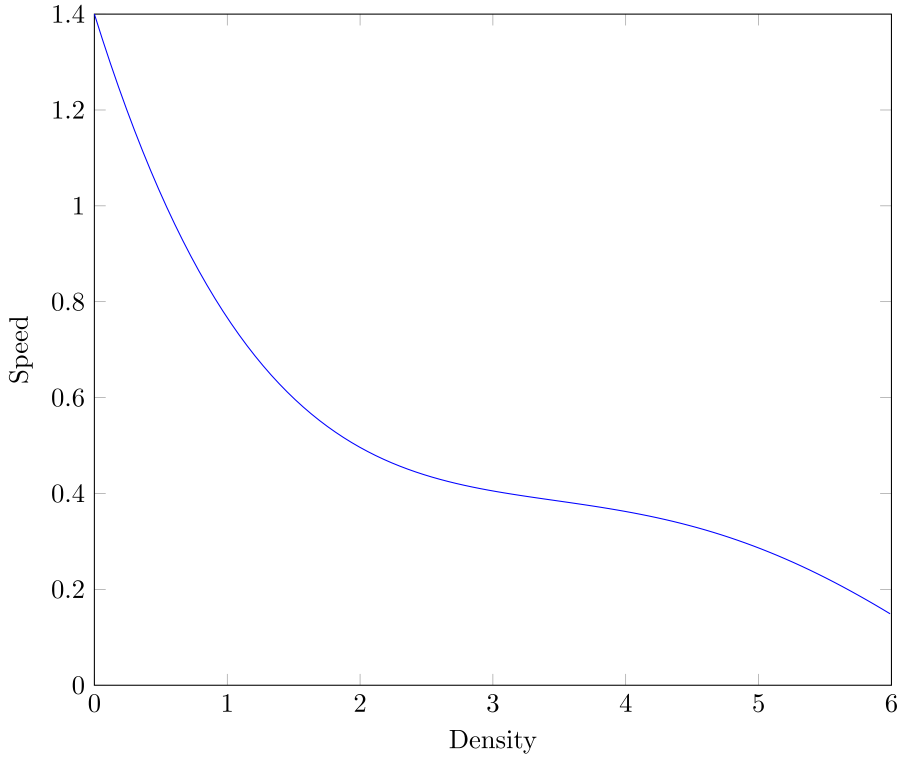


Figure 1: An example of a fundamental diagram of the speed-density relation, taken from [14].

2.1.2 Validation by Comparison

One method of validation is to directly compare simulations to data of the real situation. The data can consist of raw visitor numbers, density information, or video footage of a crowd. Most of the methods we look at in this section use data retrieved from an input video, as that is the most efficient way of capturing a real world situation. Pedestrian position, speed, and density information can all be obtained from a video, either by manual processing or by using computer vision techniques.

The simplest way of validation we look at is done by setting up experiments and then using the results to calibrate the simulation parameters. By setting up the experiments yourself, you can ensure you retrieve exactly the data that you need. Lemercier et al. [12], for example, use experimental data to tune the preferred speed of pedestrians in their model of pedestrian following behavior.

Many papers focus on tuning their simulations to video footage of the real situation. To prevent pedestrians from being obscured by obstacles or other people, the footage is often captured by a security camera placed on a high position. The video is converted into a list of positions, from which trajectories, speed, and density can be derived. Using this information, the state of the crowd simulation in the next time step can be predicted. Guy et al. [15] and Banerjee and Kraemer [16] designed methods to estimate the next state of a crowd. They then compare this to the actual state of the crowd simulation, and score it based on the difference.

Various methods focus on also optimizing the overall performance of a simulation, using the comparison score. Wolinski et al. [11] and Jin and Bhanu [17] use genetic algorithms to tune their algorithm to an input video. Wolinski et al. recognize it can be hard to compare simulations with each other, due to the different assumptions and simplifications made in the models. It would be hard to compare Moussaid's vision-based approach [18] to Helbing's force-based approach [19], for example. Wolinski et al. use their method to tune three simulation methods with varying complexity to input data, by optimizing the values of the parameters to match the data.

Validation by comparison is a very direct approach. Given a crowd simulation and an input video, it can validate the behavior by directly comparing positions. Since only the simulated positions are needed, no code integration is required. Most of the complexity lies in converting a video into pedestrian data. With advancements in computer vision this is expected to become easier and give better results over time. Using optimization methods a scenario can be directly tuned, giving a crowd simulation that can closely match an input video.

However, this validation method does not scale well. It only rewards behavior identical to the input. A specific scenario can only be validated if a video of that same environment is present, and the behavior of the crowd matches the behavior in the video. For every combination of environment, crowd density, and crowd mood a separate video needs to be available. This means it can serve as a simple means of validating a single scenario, but is not suitable for generic, large-scale validations. Even when this data is available, you run the risk of overtraining your method to specific scenarios, and causing it to perform poorly when confronted with an unknown scenario.

2.1.3 Validation by Prediction

Internal motivations can be hard to observe in input data. In data, for example videos, only the part of a pedestrians' path within the camera's range is captured. It can be hard to explain behavior in videos without knowing the pedestrians' goal, intent, or mood. Because of this, some methods focus on capturing external motivations to predict pedestrian movement. The methods discussed in this section make the assumption that external stimulations, information about a pedestrians' surroundings, can fully explain steering behavior. Given this assumption we can try to predict crowd movement from video footage.

Lerner et al. [20] propose a technique for detecting unnatural behavior, based on this assumption. Video footage of pedestrians is converted into a database of state-action pairs. The position of pedestrians in an input video is tracked for every frame, resulting in a set of trajectories. The state of each individual is recorded and saved in a database with the taken action. A pedestrian state is defined as its speed, presence of other agents around him, and preceding trajectory. A distance function is defined to compare states, and a similarity function is defined to determine the similarity of two actions.

Given a crowd simulation the actions of pedestrians in each time step, taken to be every 1.5 seconds, can then be evaluated. Taking the current state of a pedestrian, the action taken can be predicted by finding an example in the database with similar state and taking the resulting trajectory. This trajectory is compared with the action taken in the simulation to give an evaluation score for that time step. The score of each agent can be evaluated over the course of the simulation to find potential unnatural behavior.

Further work by Lerner [20] converts this idea into a crowd evaluation framework, where a crowd simulation is compared to data extracted from a set of input videos. Each pedestrian in the simulation is given a score per time step, ranging from 0 to 100%, depending on how accurately it matches data. By averaging this score over all pedestrians a simulation can be rated on how well it represents the behavior of the input video.

This idea has been extended on by Charalambous and Chrysanthou [21]. They present a more complicated data structure for pedestrian state, including visual data and static obstacles. Using more efficient data structures they also claim a performance increase over Lerner's method. In later work by Charalambous et al. [22] they try to simulate a crowd by building a database of agent states and actions. The database is trained with a set of agent trajectories taken from a video. A number of metrics are defined that form the agent state. For each agent in the simulation the state is compared to its k-nearest neighbors with Pareto ranking. The action belonging to the state with the highest rank, meaning the most natural behavior in the agents' situation, is then taken by the agent.

This method could be slightly altered to perform detection of abnormal behavior. If the average Pareto rank of a pedestrians' neighbors is higher than a given threshold the agent is likely showing unnatural behavior. By using the database to judge actions taken in the simulation, it can be scored in a similar way to [20].

The concept of using external stimuli to predict movement could be used to validate a crowd simulation. The evaluation scores can be used to spot unnatural behavior and provide an intuitive understanding of the evaluation by having the score reflect similarity. The only thing required from the input video is a set of agent positions. Positional tracking of obstacles in a video is possible to perform with computer vision, for example with background removal [23]. From these positions a pedestrian trajectory can then be constructed, as well as its state. Compared to some other methods, the generated state-action database is more generic, since it is not dependent on the scenario or the environment. Once a good database is created, it can be reused to validate any simulation with a similar crowd profile.

A good, complete data set is needed, however. Otherwise, any behavior not present in the input will be classified as unnatural. Separate videos would be needed to evaluate outlying behavior, such as climbing stairs or crossing a busy road. Unnatural pedestrian behavior in the input, for example a person suddenly turning around after he finds out he lost his keys, can also be incorrectly classified as natural in the simulation. It requires further post-processing to remove unwanted examples.

One of the problems with this method is that results are subjectively judged to be unnatural or not. Low score simply indicates potentially unwanted behavior. A human would still have to judge whether the found behavior is indeed undesirable. A big assumption made by this concept is that pedestrian behavior can be sufficiently explained by looking at external motivations. However, sometimes internal goals can cause a pedestrian to show otherwise unexplainable behavior. If this behavior is not present in the input data, it will be marked as unnatural. This method of validation is therefore well suited for validating larger crowds, where individual unnatural behavior can be assumed to not have a large impact.

2.1.4 Validation by Standardized Testing

Many of the discussed evaluation methods only evaluate a specific scenario and not the overall performance of an algorithm. It can be required to validate a model independent of scenarios. For example, simulations of fire evacuations often try to meet standardized requirements [4, 24]. A framework with benchmark tests can be used for this. The performance of a simulation gets tested by running a number of standardized scenarios in which it should succeed.

One example of a fire evacuation benchmark is the international standard proposed by the fire research division of the United States [24]. This standard tries to verify that the simulation is a correct representation of the real situation. The simulation should perform accurately in any environment and show certain desired behaviors, such as queuing for a lift, reduced speed on stairs, or fleeing to the nearest exit. An extension to this standard [4] has been offered that introduces extra qualitative measurements to reward more complex algorithms, and separates mandatory properties from ones that are nice to have.

These benchmarks are focused on evacuation scenarios, where the primary goal is for everyone to exit the building. Other benchmarks exist that focus on steering behavior [25, 26, 17]. The main assumption in these methods is that humans try to minimize the amount of energy used when moving towards their goal. This means the performance of a crowd simulation algorithm can be evaluated by looking at the efficiency of paths, in terms of time and length. Both SteerBench [26] and SteerBug [17] evaluate against a number of preset scenarios. The Scenario Space framework [27] evaluates against a randomly generated set of scenarios. Agents are put in an environment with randomly generated obstacles and goals, and are judged on number of collisions and time to reach the goal.

Using these metrics, behavior can be caught that cannot be judged by a human eye. For example, agents following a slightly oscillating path can be punished by penalizing excessive steering. It also provides a good stress test for a simulation model and can indicate what the simulation struggles with, compared to other models.

This method of validation is more focused on validating an algorithm by comparing its overall performance to other algorithms, rather than comparing it to real data. The advantage of this approach is that validation can be done independent of scenarios. Since the assessment of the model is meant to be a general one however, the scenarios need to be chosen carefully. It needs to be ensured that they can reflect the strengths and weaknesses of any model. This can lead to some unfair comparisons and scores not reflecting the model's intended purpose. A vision-based simulation model should always be expected to score better than a simpler force-based model. Some models make simplifying assumptions, for example discretizing the scenario environment [28]. On these benchmark tests they can still score higher than models that do not make these assumptions, which leads to a score that does not always reflect the model's power.

Another problem is that the resulting evaluation scores lack intuition. They are related to the number of agents, agent density, and the size of the environment. Some of the proposed metrics can be normalized to reflect this, but others, such as number of (near-)collisions cannot. In a scenario with more agents, the number of agent interactions increases exponentially, which is hard to reflect in the given score. The scores give no indication of performance, unless compared to a score in a similar scenario. It would be ideal to have a scoring system that does not have this limitation. The scenario space method has achieved this, but instead lacks the reliability and repeatability of the other frameworks, due to its randomness.

There are also some practical downsides to this method. A simulation model needs to be implemented in the evaluation framework, or the benchmark scenarios need to be implemented in the simulation software. In both situations reimplementing existing work is required, which can lead to different results.

2.1.5 Validation in Existing Frameworks

In this section we look at some of the widely used crowd simulation frameworks available on the market, and how they validated their simulation models.

Viswalk [29] use a social force model as a basis for their simulations. Viswalk claims scientifically validated results that provide a realistic simulation both in- and outside of buildings. As far as we can see, this validation is mostly based on Kretz et al. [30] and Henningsson et al. [31]. In [30], the authors show that a social force simulation can be tuned to match data from real experiments. In [31], Viswalk is evaluated against experimental results. It is shown that after optimizing the user settings both evacuation time and flow are similar to the experimental data. The default user settings give wide variation however, deviating by up to 95% on evacuation time, and up to 54% on predicted flow. The final conclusion is that it is possible to obtain results similar to a real situation, but the user has to have a good understanding of the model and the predicted crowd behavior for this.

Legion [32] has validated their simulations using a large collection of video footage and pedestrian movement across many scenarios, according to their website. They calibrate their algorithms to this data and then separately validate it against real crowd behavior.

AnyLogic simulation software [33] does not claim to be validated. Instead of this it contains a calibration tool. The user can specify a behavioral pattern it desires the simulation to follow. The software then optimizes the parameters to closely match the given pattern.

MassMotion [34] is pedestrian simulation software developed by Oasys. Its models are verified both by the *International Maritime Organisation*, and the *National Institute of Standards and Technology*, and validated with a range of real-world evacuation events. The verification and validation process is described in [35]. The authors demonstrate that the real situation is sufficiently represented by comparing evacuation times in MassMotion to observed evacuation times. The resulting predictions range from 7.3% faster to 15% slower.

The crowd simulation frameworks on the market validate their simulations in varying ways, using different ways of representing real crowd data. Because of this, frameworks can give different results for the same scenario, depending on the representation of the environment and the crowd data used. Ideally, frameworks should give similar output when validated with the same data.

2.1.6 Conclusion

Ideally, a simulation should be validated by data from a comparable real-life scenario. Of the validation methods discussed, the most applicable ones for this purpose rely on validation by comparison or prediction, the methods described in Sections 2.1.2 and 2.1.3, respectively. Standardized testing is mostly relevant when comparing the performance of simulation models between each other, rather than validating with data.

The main limitation of the methods performing validation by comparison is that they try to directly match a given video. This results in overtraining of a model to specific scenarios and means an input video is needed for each separate scenario. This method also tunes models to match individual behavior rather than crowd behavior. Some of the discussed methods [17, 11] try to find the optimal parameter values by using optimization techniques, for example genetic algorithms. However, the representation of the agent state is very basic. Both papers simply compare the total position difference between the video data and a simulation as state, and use this as a direct score for the simulation.

The methods that perform validation by predicting a pedestrians' movement yield more reusable results. The prediction is based on a more complex representation of agent state than the comparative methods. The main limitation of these approaches is that they are mostly focused on recognizing and marking potential problems, leaving interpretation open to humans. None of these methods give a way to fix the unnatural behavior after recognizing it.

It can be tempting to focus on the presence of noticeable visual artifacts when validating a simulation. How convincing a simulation is perceived mostly relies on details, since the human eye is trained to notice small details breaking a pattern. When visually comparing a simulation to the real situation humans will mostly notice abnormal behavior, such as sudden turns or excessive acceleration, rather than differences in flow or density.

Regarding crowd safety however, interesting metrics to compare to reality are density, crowd flow, and throughput. Crowd simulations cannot account well for situations where actions of an individual endanger crowd safety. Humans are unpredictable, have large variance between them and are thus hard to model accurately. It therefore makes more sense to focus on group metrics as means of validation. Most of the safety risk in large crowds lies in crowd density and flow through bottlenecks. A bottleneck with too little capacity can result in a situation where a crowd is at risk, due to increased density.

Fundamental diagrams are a way to validate simulations with group metrics. The relation between speed, density, and flow is extensively studied in literature, and provides a way to easily compare a simulation to real data. However, simulations are only directly compared to fundamental diagrams. One of the many available fundamental diagrams is selected, and a simulation is considered valid if the *speed-density* relation matches the chosen diagram. We believe this is an inaccurate way of validation, since there are so many factors influencing the exact form of the diagram. As discussed in Section 2.1.1, environment, crowd demographic, and measurement metrics can all have a large impact on the resulting diagram. In the remainder of this thesis we focus on the effect of density measurement metrics on the resulting fundamental diagram.

2.2 Measuring Density

In most real life scenarios a crowd will be spread out. Crowd density will vary over time and by location. When a train arrives on a train station, crowd density will peak for a while, until passengers have exited the platform. Similarly, local crowd density will be higher in a bottleneck situation such as a narrow exit than in a large open space. When a large amount of people try to cross a narrow space, density will increase, and the crowd flow will slow down [10]. According to various studies this can lead to a dangerous situation if the density increases above 6 people per square meter for static crowds, or 3 to 4 per square meter for moving crowds [1, 36, 37]. To correctly assess the risk in a crowd, we need to be able to accurately measure density.

Density can be measured in a straightforward and intuitive way. We will first take a look at problems with this simple way of measuring in Section 2.2.1, and requirements a metric should meet. We will then look at some more advanced approaches that try to solve these problems. In Section 2.2.2, we look at approaches that discretize the environment into a grid of cells. In Section 2.2.3, we discuss approaches that look at density from a pedestrian perspective.

2.2.1 Observations

Traditionally, density is measured by dividing the total amount of people inside a region by its area, resulting in a number of people per square meter. This straightforward way of measuring density leads to some problems however.

Averaging the amount of people over the occupied area will often lead to very rough estimates. Helbing et al. [38] studied ingress data of the Jamarat Bridge and found local density values can be twice as high as averaged density, even for a relatively small area of 27 by 22 meters. Since global density is not a reliable indication of the local situation, we need a metric that can recognize local maximals.

The classic method of measuring density is not robust to small changes in space and time when used to measure local density. To increase detail and precision, the area considered when measuring density is often kept small compared to the size of a person. Since the range of densities physically possible is relatively small as well, this means a single person can have a large impact on local density. When measuring density, a person is often counted either completely, or not at all. For these reasons, a person walking on the edge of a measurement area can lead to large fluctuations in measured density, as observed in [9]. For similar reasons, placement and orientation of the measurement area can have a large effect on the result. If the result is independent of these factors, reliability is increased. The outcome should also be easy to interpret and be comparable to results from literature.

To summarize, the requirements we look for in a density metric are the following:

- The metric should be able to recognize local peaks in density.
- The result should not show large scatter over time.

- The result should be easy to interpret.
- The result should have a relation to the global density.

2.2.2 Discretized Density Measurement

As discussed in Section 2.2.1, we would like to recognize maximal density on a local, rather than global scale. One of the ways to increase the sensitivity is to discretize the continuous environment into smaller regions, and measure density separately for each resulting region.

An intuitive way of dividing the environment is to create a grid of equally spaced, square cells. The density in the cell can then be calculated by simply counting the number of agents in each cell and dividing it by the area. To increase detail and precision the area considered when measuring density is often kept small, compared to the size of a person. However, the resulting density of this method is highly dependent on chosen parameter values, such as cell size. To reduce this effect pedestrians can be counted proportionally to the ratio of their body inside each cell.

Helbing et al. [38] propose a grid-based metric based on weighted distance from a measurement point to each pedestrian. A grid of points is placed in the environment, after which the local density can be calculated for each of these points. A Gaussian function is used to smoothly reduce the influence of pedestrians on the measurement point. The farther away a person is from the measurement point, the less influence he will have on that point's density.

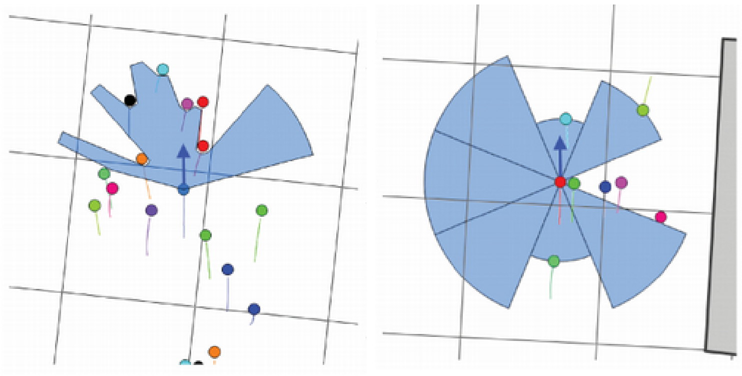
Steffen et al. propose another approach based on Voronoi diagrams [39]. The method computes a Voronoi diagram with pedestrian positions as sites. The Voronoi cell of each person can be seen as its personal space. The environment is then divided into a grid of cells, and density is computed for each cell. Density is computed based on the size of the Voronoi cell, for each cell intersecting the measurement area.

Zhang [9] compares this method to classic density, and concludes Steffen's approach has less scatter and is less sensitive to small changes. The method also provides an intuitive result, giving a density per square meter that is independent of the size of the measurement area.

2.2.3 Perceived Density Measurement

An alternative way of measuring density is by looking at density perceived from a pedestrians' perspective. This approach is used as part of agent state in several of the validation methods discussed in Section 2.1. In this section, we look at methods that measure the perceived density around pedestrians.

When measuring density by rasterizing the environment into a grid, a large part of the cells may not contain any people. A way to solve this overhead would be to compute the density each person perceives in its local neighborhood. This means the amount of work done is not based on the size of the environment, but rather on the amount of pedestrians. It also gives the option of taking a person's



(a) Visibility-based density

(b) Radial-based density

Figure 2: State representations that can be adapted to measure density, by Charalambous et al. [22].

perception of the surrounding density into account. Someone walking right in front of you will have a larger effect on the perceived density than someone behind you, outside of your field of view.

A simple way to calculate perceived density would be to measure the percentage of occupied space around each person, and convert the result to a number of people per m^2. This can be done by taking an area of one square meter around a person and counting the amount of people in this space. This has several downsides however. Similar problems to the problems regarding classic density, as discussed in Section 2.2.2, apply. The result will suffer from large scatter over time and location. Furthermore, the resulting density will have a minimum value of 1, as the perceiving person will be contained within the measurement region.

Charalambous and Chrysanthou propose a Temporal Perception Pattern to represent the state of a person [21]. The state is encoded in a visibility polygon of the area around each person. They create this state by sampling the vision of a person on regular angular intervals, using a ray that finds the closest object in a direction. This results in a visibility polygon as seen in Figures 2a and 2b. The state shown in Figure 2a only takes samples along the person's field of view, while the state shown in Figure 2b finds the closest obstacle in all directions. This representation of state could be adapted to represent density. This could be done either by taking the area of the visibility polygon as a fraction of the total area, or by taking the amount of non-blocked rays compared to the total number of rays. These results lack intuition however. It is not immediately clear how a result such as *30% visible* should be interpreted.

Jacques et al. [23] use Voronoi diagrams to detect and classify groups of people. A diagram is computed, using pedestrian positions and obstacles as Voronoi sites. By checking the distance of a person to each of his Voronoi neighbors over time, it can be estimated whether an emergent group is voluntary or involuntary. If two or more people keep shorter distance than the size of their preferred personal space for an extended amount of time, it is likely to be a voluntary group. Vision can be taken into account by only taking the area of the personal space within a person's field of view into account.

When measuring density it can be useful to detect the presence of groups. Groups can have a large effect on local density. Groups indicate a large amount of people voluntarily packing together more so than necessary, which could lead to exaggerated density measurements. Although this method in itself is not enough to measure density, it could easily extend an existing method with group recognition, for example the Voronoi approach described in Section 2.2.2.

2.3 Voronoi Diagrams

Given a planar area and a set of defining sites, a *Voronoi Diagram* is a division of the area into separate regions. Each of these regions contains exactly one of the given sites. Any point inside a region is closer to this containing site than to any of the other sites. An example of a Voronoi diagram can be seen in Figure 3. A Voronoi diagram is defined by its *sites*. In the standard Voronoi diagram [40], sites are points lying on a plane. The Voronoi diagram can be extended to also include other site shapes, such as lines, discs, and convex polygons [41].

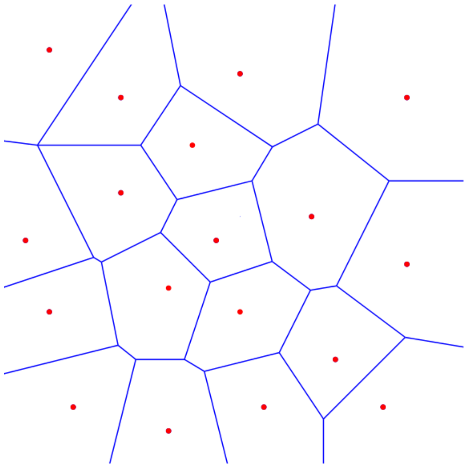


Figure 3: An example of a Voronoi diagram. The red points depict *sites*. The available area is divided into Voronoi cells by blue *Voronoi edges*.

2.3.1 Applications

Voronoi diagrams have a number of useful properties. Voronoi diagrams are relatively cheap to compute. Given a set of n sites, an initial diagram can be computed in $O(n \log n)$ time, and updated in $O(n)$ time per obstacle when an obstacle is added. The resulting structure uses $O(n)$ vertices, as each Voronoi site is a convex obstacle consisting of a linear number of vertices.

Any point lying on a Voronoi edge is equally far away from at least two obstacles. As Voronoi vertices represent a shared point of two or more Voronoi edges, each Voronoi vertex is equally far away from at least three sites. The Voronoi edges represent the routes through the environment that maintain the biggest distance to obstacles. The medial axis of a *Generalized Voronoi Diagram* can be used for planning a path with minimal clearance to obstacles, as done in [42], [43], and [44]. By applying A* path planning to the resulting medial axis graph, a shortest path in the graph can be planned through an environment containing obstacles. The ECM Framework [45], used for simulation in Appendix A, uses this concept for crowd simulations.

2.4 Retrieving Data from Video

Many methods focus on validating simulations by using data from a real situation. The most common data is video footage of a crowd, as that is the most efficient way of capturing a real-world situation. qwepruioedestrian position, speed and density information can all be obtained from a video, either by manual processing or by using computer vision techniques.

In this section we first look at what we need from an input video in Section 2.4.1, how to convert video coordinates to world coordinates in Section 2.4.2, and the way existing research acquired their input data in Section 2.4.3.

2.4.1 Data

When comparing a video to a simulation, the only thing required is the position of pedestrians for each time step. Given a set of positions, a pedestrians' trajectory and speed can be approximated by comparing subsequent positions, as long as the time step between frames is small enough. Some amount of detail will be lost due to the trajectory being discrete. However, the standard frame rate for videos is significantly higher than a sample rate that results in visually smooth trajectories, so this is usually not a problem.

As discussed in [11], a crowd simulator can be seen as a state machine converting a crowd, consisting of a set of agent positions and velocities, into a next state for the crowd. When comparing crowds, we are only interested in the external state of the crowd as a whole. Internal state, such as personal goals, are not relevant. Because of this, capturing only position gives a sufficient representation of the crowd in the video.

2.4.2 Screen to World Coordinates

One of the challenges lies in the conversion from points on a 2D screen to 3D world positions. Often it is assumed that the environment is filmed by a camera on a fixed position, leading to a fixed viewing angle. The pedestrians are assumed to walk on a planar area, so positions can be represented by an x- and a y-coordinate. This assumption is inherent to the conversion from 2D to 3D, since a single screen pixel can represent an unlimited amount of positions in the 3D space. By assuming the pedestrians walk on a plane, this can be reduced to a guaranteed position. By computing the intersection of a ray through the given pixel and the ground plane, an exact position can be calculated, as suggested by [46] and [47].

2.4.3 Tracking

Automatic tracking of objects is an active and evolving area of research within computer vision. A wide range of techniques have been designed for this purpose. The general technique consists of a preprocessing step where the background is removed from the video. Dynamic parts of the videos are separated from static parts in this step. Remaining parts are then classified as either an object that needs to be tracked, or noise that can be discarded. This is done using a variety of techniques, using prior knowledge about the objects' structure. For humans the shape of the body or head can be used to filter pedestrians from noise.

After an object is recognized, it needs to be tracked for the duration of the video. As an object moves, it will change in appearance, due to perspective, lighting or possible obscurance by other objects. As long as the changes in appearance and position are small, likelihood can be used to determine whether an object is likely to be the same as one on a previous frame. [48] and [49] give a broader overview of the varying techniques used for recognition and tracking in the computer vision area.

Jacques et al. [23] automatically track people in a video and then use the data to classify groups in a crowd. Other papers in the crowd simulation field take shortcuts to convert a video to data. Many of the papers focusing on validation use videos from controlled experiments with marked pedestrians to skip the recognition phase and make tracking easier [12, 50, 51]. Videos from uncontrolled crowd simulations are often tracked manually, by annotating the pedestrians' position per frame [17, 52, 21, 20, 53]. The other papers discussed in Section 2.1 do not use video data, or do not specify the way they converted their video.

3 Validation

As discussed in Section 2.1, several types of validation can be recognized in literature. Many methods focus on validation by comparing to input data, or by predicting movement based on an agent's state. The comparative method is the most straightforward one, trying to directly match the simulation to input data. However, the representation of the agent state is simple, and often compares only on position.

Predictive methods use a more complicated implementation of agent state: taking surrounding pedestrians, speed, or vision into account. This is an improvement over the comparative methods' state, but we believe it is better to evaluate a crowd on a larger scale than individual movement. Individual behavior is sensitive to noise and is hard to replicate in a simulation. More important for crowd safety is the crowd-wide behavior. Actions of a single pedestrian can have a large impact on the crowd behavior, but are hard to predict and simulate correctly. It makes more sense to do a separate simulation with an adapted scenario when testing cases where an individual affects a crowd.

In this section we take a closer look at fundamental diagrams as a means of validation on a crowd level, in Section 3.1. In Section 3.2, we discuss group metrics that can be used to compare the simulation to real data.

3.1 Crowd Level Validation

Much of the existing research validates crowd simulations by directly comparing positions and speeds of individuals to real data. However, this means a single person showing outlying behavior can have a large influence on the result. Comparing a simulation on personal characteristics will ensure that it is visibly similar to real data, but regarding crowd safety it is important that simulations are similar in the number of people and the overall group behavior. Safety risks often lie in increased density and reduced flow, as discussed in by Still [1].

Fundamental diagrams, discussed in Section 2.1.1, capture this relation between density, flow, and speed. The higher the density is in a crowd, the lower the average speed and flow. Fundamental diagrams are commonly used as a means of simulation validation. However, many different variations of the diagram exist, as analyzed by Zhang [9]. Zhang reviewed the fundamental diagrams for different flow types, such as uni- and bidirectional flow, and concluded that they differ from one another, for densities of over $1/m^2$. This is in contrast with observations done by Fruin [10], Pushkarev et al. [54], and Lam et al. [55]. Different fundamental diagrams are also available for many environments, such as open areas, bottlenecks, and staircases [14].

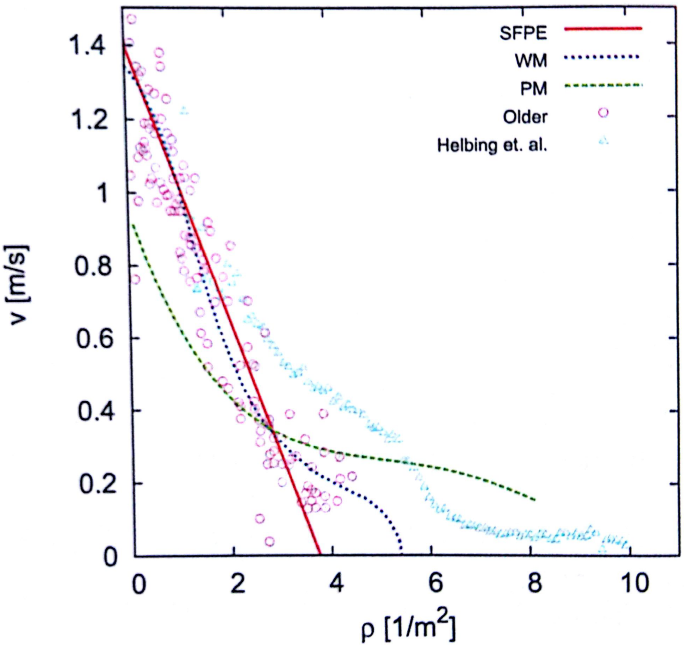


Figure 4: An example of a fundamental diagram of the *density-speed* relation, taken from [9].

Much of the research disagrees with each other. Zhang [9] tested evacuation time for a simple scenario using the fundamental diagrams described by [56], [10], and [14], and found evacuation times were incomparably different. As seen in Figure 4, fundamental diagrams agree reasonably well for lower densities, but vary greatly for higher densities. According to Weidmann's model, a crowd is unable to move at a density of $5.5/m^2$, but Predtechenskii and Milinksii's model suggests this is not the case until much later. Both conflicting cases are supported by experimental data, from Older [6] and Helbing [38], respectively.

Part of this difference can be explained by the nature of the crowd and the environment. However, the measurement method used for measuring density or flow can have a large impact on the results as well, as observed in [57]. Flow measurement for fundamental diagrams is well-defined for scenarios like a one-directional hallway, similar to traffic flow. For bidirectional flow, or flow in open environments it is harder to define the metric. Speed can easily be calculated in both one- and two-dimensional environments, by taking the magnitude of the velocity vector.

As further discussed in Sections 4.1 through 4.3, computing density is not entirely straightforward. Multiple methods exist, each with their own strengths and weaknesses. In existing literature it is not always stated which density method is used.

Simulations are mostly validated using the *speed-density* diagram. Seyfried et al. [58], Curtis et al. [13], and Lemercier et al. [12] use a one-dimensional variation of density for this. The density at a location is determined by following distance between two pedestrians at this point. Wolinski et al. [11] use fundamental diagrams to optimize simulations, but leave their definition of density unspecified.

Predtechenskii and Milinskii's fundamental diagram [14] uses a density measurement method closely related to the classic method described in Section 4.1. Helbing et al. [38] uses the Gaussian method, described in Section 4.2, to compute density in their comparison of video data and the fundamental diagram. Fruin, in contrast, uses the *Pedestrian Area Module*, the available area per square feet, to represent density. This can be computed with the Voronoi density method [39], described in more detail in Section 4.3.

The large variety of scenarios and measurement methods makes it hard to judge quality of results. Many fundamental diagrams exist, each of which yields different results for the relation between speed and density. Adding the different measurement methods as well, it is possible to tweak results so that one method is favored. In Section 5, we will experiment with the different density metrics discussed in Section 4, to get a better understanding of their parameterization, and their strengths and weaknesses.

3.2 Crowd Metrics

As discussed in Section 3.1, we believe it is important that simulations are validated based on a crowd, rather than an individual level. Validation is mostly done by capturing metrics from behavior of a real crowd, and comparing these metrics to the behavior shown in the simulation. These metrics can be divided into individual and group metrics.

Individual metrics focus on validating the behavior of a single pedestrian. This includes metrics such as average movement speed, total distance traveled, and steer rate. On a lower level, the direct position per simulation step is compared.

Group metrics capture the macroscopic properties of a crowd. As discussed by Still [1], risk in large crowds is directly related to density and flow. For this reason, simulations are often validated using one of these metrics, together with fundamental diagrams. On an environment level, density can be calculated by counting the number of people in an environment. Flow can be computed by dividing the number of people that pass through a corridor by the time taken, to obtain a throughput rate in number of people per second. However, local values for these metrics often show large variance over time and space, compared to the global values, as observed by [38]. For this reason, when validating a crowd it is necessary to compare to local values.

As discussed in Sections 2.2, local density measurement methods are well-defined for a one-dimensional environment, but for a two-dimensional environment this is not the case. For density, we recognize three suitable grid-based methods: Classic, Gaussian-based, and Voronoi-based. In Sections 4 and 5, we look at the advantages and disadvantages of measuring density using each of these methods. These methods compute a value in number of people per square meter, which can then be compared to the densities, flows, and speeds measured in the real data. Flow has an intuitive definition in two-dimensional scenarios where agents move into similar directions, such as corridors or evacuation scenarios. It is not clear what a suitable flow measurement method would be for two-dimensional environments with unstructured flows. Because of this, for validation we focus on the *speed-density* version of the fundamental diagram.

4 Measuring Density

Given the requirements specified in Section 2.2.1, the Voronoi density method [39], and the Gaussian density method [38] are the most suitable methods. The other methods discussed in Section 2.2.2 struggle too much with high scatter of data, and high reliance on choosing the right parameter values to be applicable. The local methods discussed in Section 2.2.3 focus more on computing personally perceived density, instead of crowd density. This is useful when trying to understand, rather than analyze crowd behavior. However, the results are harder to interpret and not comparable to safety metrics in literature, measured in number of people per area.

In this section we will further discuss the three density measurement methods that most suit our needs: Classic, Gaussian and Voronoi density. In Section 4.1 we will discuss the advantages and disadvantages of the simple, classic way of measuring density. In Sections 4.2 and 4.3 we will do the same for the Gaussian and Voronoi methods, respectively.

4.1 Classic Density

The most straightforward way of calculating density divides the environment into a grid, and computes local density by counting the number of people in each cell. We will discuss the approach used for this method in Section 4.1.1 and its strengths and weaknesses in Section 4.1.2.

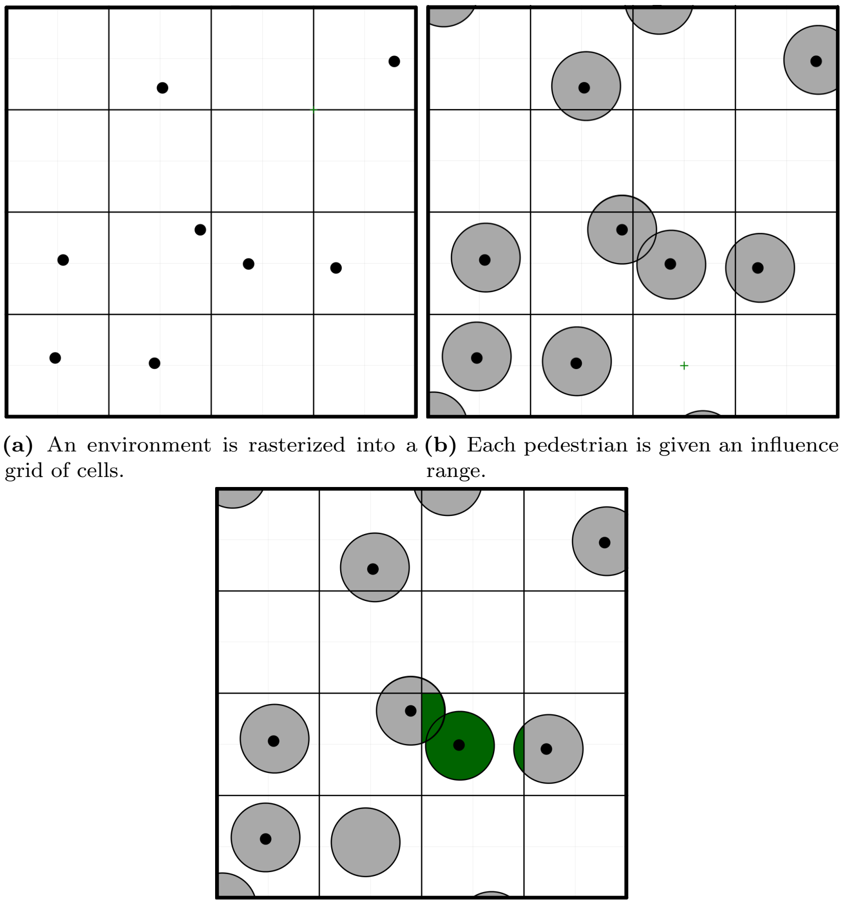
4.1.1 Approach

With this method the number of pedestrians per cell are counted. To decrease temporal and spatial variance pedestrians can be treated as disks, instead of points. This adds an extra parameter, namely the radius of the disk. If pedestrians are treated as disks they can be proportionally counted towards the density for each cell they intersect. An overview of the approach is shown in Figure 5.

The approach is shown in Equations 2 and 1. For each pedestrian p in the set of pedestrians P, it is computed what portion lies inside the measurement area A_m, by integrating over the area $|c_p|$ of the circle c_p around the location of p. The total is summed up, and divided by the area of a single pedestrian disk to obtain the number of people per square meter. In Equation 2, $\vec{x}$ denotes a location inside p.

$$f(\vec{x}) = \begin{cases} \frac{1}{A_m}, & \text{if } \vec{x} \in A_m \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$D = \sum_{p \in P} \frac{\int_{c_p} f(\vec{x}) d\vec{x}}{|c_p|} \tag{2}$$



(a) An environment is rasterized into a grid of cells.

(b) Each pedestrian is given an influence range.

(c) Density is calculated by intersecting the influence ranges with the area of a cell. The green areas determine the density in the chosen cell.

Figure 5: The classic density measurement method.

4.1.2 Strengths and Weaknesses

The classic density method mostly performs well when measuring global density. It is hard to measure density locally without introducing high variance in the measurements. One of the characteristics of crowd density is that its value is limited by the amount of people that physically fit on a square meter. Since this value is relatively low to start with, it means a single person can have a large impact on local density. When measuring density, a person is often counted in a binary way; either completely, or not at all. For these reasons, a person walking on the edge of a measurement area can lead to large fluctuations in measured density, as observed in [9]. For similar reasons, placement and orientation of the measurement area can have a large effect on the result.

It can be hard to determine a value for the radius parameter R. Interpreting it as the physically occupied space would be a logical approach, but can lead to high variance for smaller grid sizes. Using higher values of R can lead to loss of detail, as the influence of a pedestrian is spread out over a large area.

By introducing a dynamic parameter for influence range, interpretation of the density results becomes harder. These issues make R a hard to tune parameter. It is hard to intuitively choose a value for R, and its effectiveness is related to the chosen grid size.

This method does not take obstacles into account. One solution for this would be to subtract obstacles from a cell's area. This could lead to artifacts in the density measurements, where a single pedestrian inflates density to a high level. For this to happen the pedestrian would have to be close to the wall, which is often a situation that only happens on higher densities. For this reason subtracting obstacles from the cell area is a good way of taking obstacles into account.

4.2 Gaussian Density

In [38], the authors recognize global density is not an accurate reflection of risk in a crowd, since maximal local density can be significantly higher than average density. They propose a metric that calculates local density by computing the influence of a pedestrian with a Gaussian distribution. In Section 4.2.1, we will discuss the general approach used for this metric, and in Section 4.2.2, the strengths and weaknesses of this method will be discussed.

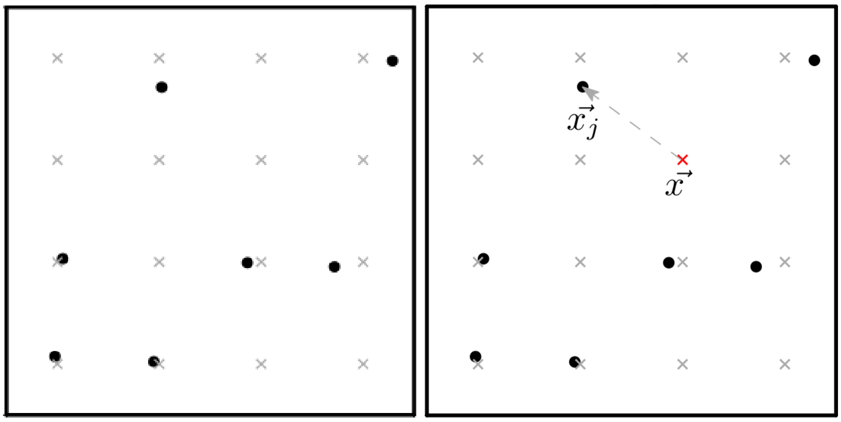
4.2.1 Approach

Given a pedestrian location $\vec{x_j}$, a measurement point $\vec{x}$, and a parameter R controlling the measurement radius, the influence is computed as

$$f(\vec{x_j}, \vec{x}, t) = \frac{1}{\pi R^2} \exp\left[\frac{-\|\vec{x_j}(t) - \vec{x}\|^2}{R^2}\right] \tag{3}$$

Density D for a measurement point $\vec{x}$, at time t, is calculated by summing the resulting weight of each pedestrian in the simulation, as in Equation 4. A visual overview of the method is shown in Figure 6.

$$D(\vec{x}, t) = \sum_j f(\vec{x_j}(t), \vec{x}, t) \tag{4}$$



(a) A grid of measurement points is created in the environment. The measurement points are marked by gray crosses.

(b) For a measurement point $\vec{x}$, the influence of each pedestrian at location $\vec{x_j}$ is calculated using Equation 3. The density is then calculated as in Equation 4.

Figure 6: The Gaussian density measurement method.

In this equation R controls the influence range of a pedestrian. Equation 3 is based on a Gaussian probability density function. It can be shown that 63% of the density contribution originates from pedestrians within R. Increasing R directly increases the influence of nearby pedestrians. When the distance from a measurement point to a pedestrian is larger than $3R$ the influence becomes practically zero. Because of the nature of Gaussian distributions, the sum of local density for each measurement point will be comparable to the global density. The Gaussian distribution is a probability density function over the environment. Therefore, the density for a single pedestrian sums up to 1 over the entire environment. When $R \to \infty$ the local density is equal for each point of the environment. When $R \to 0$, density approaches infinity for measurement points close to pedestrians, and approaches 0 for any other point.

This method is suitable for a continuous representation of local density in an environment. However, when visualizing density it is desired to have a limited number of measurement points. This can be done by dividing the environment into a grid of points.

4.2.2 Strengths and Weaknesses

As long as the speed of pedestrians is limited, and a relatively small time step is chosen, the resulting density will have low variance over time and space. As the Gaussian distribution smoothly diminishes the influence of pedestrians, the amount of data scatter will be limited. All pedestrians are considered for every time step. In contrast, for classic density the influence of pedestrians is binary, and depends on an arbitrary range parameter, leading to large variance in both time and space.

Since the method uses measurement points, rather than measurement areas, the result is independent of the chosen grid size. Some detail will be lost by interpolation for query points in between grid cells, but the overall result will be similar. Unlike the classic method, every measurement point experiences some amount of density.

The meaning of the variable R lacks intuition. The underlying assumption for the density model is that the influence a pedestrian has on its environment follows a Gaussian distribution. Given this assumption, the location of a pedestrian corresponds with μ, and R corresponds with σ. However, it is hard to give an intuitive explanation for this assumption. Influence experienced by pedestrians should be maximal when close to another person, and approaching zero as the distance grows, but it is hard to capture how the influence diminishes.

The resulting density is highly dependent on choosing the right value for R. When the value chosen is too high, a large amount of pedestrians can be taken into account, leading to a result that loses detail. When the chosen value is too low, only a single pedestrian may be taken into account, leading to high variance. Since the local density is related to global density this can lead to inflated values around pedestrians for smaller values of R. Similarly, density measurements near the border of the environment are lower, since the out of bounds area does not contain any pedestrians.

Using this measurement method, it is hard to capture the influence of obstacles. Ideally the presence of obstacles close to a measurement point should lead to a higher density, due to the decreased freedom of movement. Although it is complicated to include obstacles in the density measurement, it is possible. For example, Plaue et al. [59] have described one method of taking obstacles into account.

4.3 Voronoi Density

The Voronoi density method, described in [39], assigns a personal space to each pedestrian in the environment. The concept of personal space is used to represent the space available to the person. It is constructed by creating a Voronoi diagram [40], with each person acting as a Voronoi site. The size of a Voronoi cell has an intuitive relation to density. A large Voronoi cell means a person has space to move freely on at least one side. Similarly, a smaller Voronoi cell means the person is restricted in all directions. This results in a measure of the available free space in square meters. By taking the reciprocal of this, this can be converted into a number of people per square meter, for the given cell.

4.3.1 Approach

Like the other density methods discussed, the environment is divided into a square grid of equally-sized cells. Steffen et al. [39] discusses two density methods, based on a similar approach. The first, shown in Figure 7a, calculates density in each cell by Equation 5, in which N is the set of pedestrians within the measurement area. A_i denotes the area of the Voronoi cell of pedestrian i. This method considers pedestrians whose center of mass lies within the measurement area. Intuitively, this equation can be explained as calculating the average personal space available to agents in a grid cell.

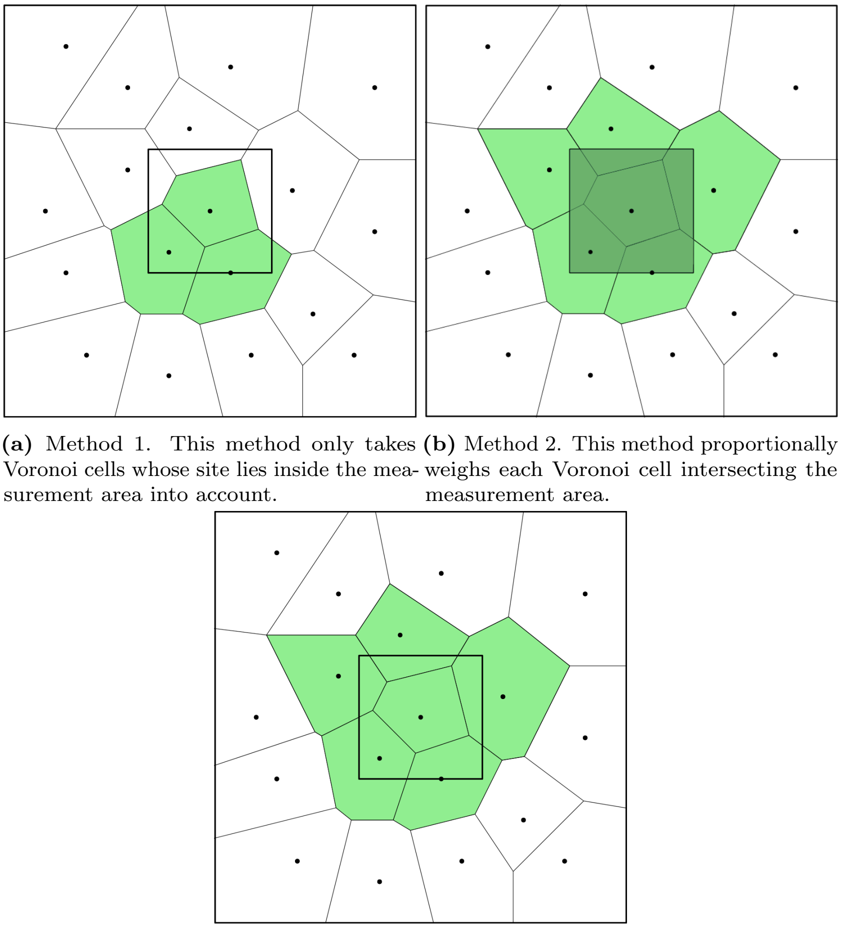
$$D_V = \frac{N}{\sum_{i=1}^{N} A_i} \tag{5}$$

The second method uses Equations 6 and 7 to calculate density for a chosen grid cell. The method computes a weighted average density over the measurement area A_m. The value of a Voronoi cell is defined by its area. The weight of each Voronoi cell is defined by the size of the intersection of the cell and the measurement area.

$$D_{V'} = \frac{\int_{A_m} f(\vec{x}) d\vec{x}}{|A_m|} \tag{6}$$

$$f(\vec{x}) = \begin{cases} \frac{1}{|A_i|}, & \text{if } \vec{x} \in A_i \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The approaches used in the paper are shown in Figure 7a, for D_V, and Figure 7c, for $D_{V'}$. As discussed in [39], D_V suffers from a high amount of scatter, due to the method only considering persons whose center of mass lies within the measurement area. Furthermore, the resulting density is undefined when no Voronoi sites lie within the area. We address this by considering any Voronoi site whose cell intersects the measurement area, rather than only the sites with center of mass within A_m. This guarantees that at least one site is considered. This makes the densities computed by D_V more consistent in lower density areas, and gives results closer to $D_{V'}$. This approach is shown in Figure 7b.



(a) Method 1. This method only takes Voronoi cells whose site lies inside the measurement area into account.

(b) Method 2. This method proportionally weighs each Voronoi cell intersecting the measurement area.

(c) Method 3. This method weighs each Voronoi cell intersecting the measurement area equally.

Figure 7: Voronoi density measurement methods.

4.3.2 Strengths and Weaknesses

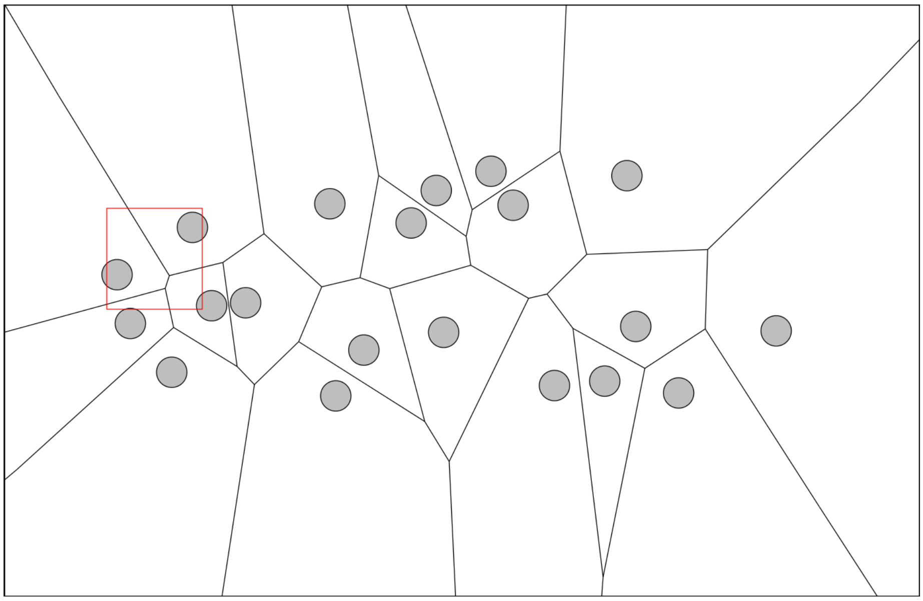
The Voronoi density method has reduced scatter compared to other methods, as shown in [9]. It gives an intuitive result, in number of people per area. This makes it easily comparable to other density results from literature. The area is still discretized, but measured values are less reliant on grid size than other methods, as it takes the ratio of Voronoi cells inside into account. Unlike the classic method, this method is also capable of dealing with small grid sizes. The classic method start giving inflated values and high local peaks when the measurement area is reduced below the size of a person. This method performs well even in sparse areas, because it takes personal space into account.

Just like with the other methods discussed in Section 2.2, it is hard to determine parameter values. The size chosen for the measurement cells has less influence on the resulting density than other methods, but still has some unwanted properties.

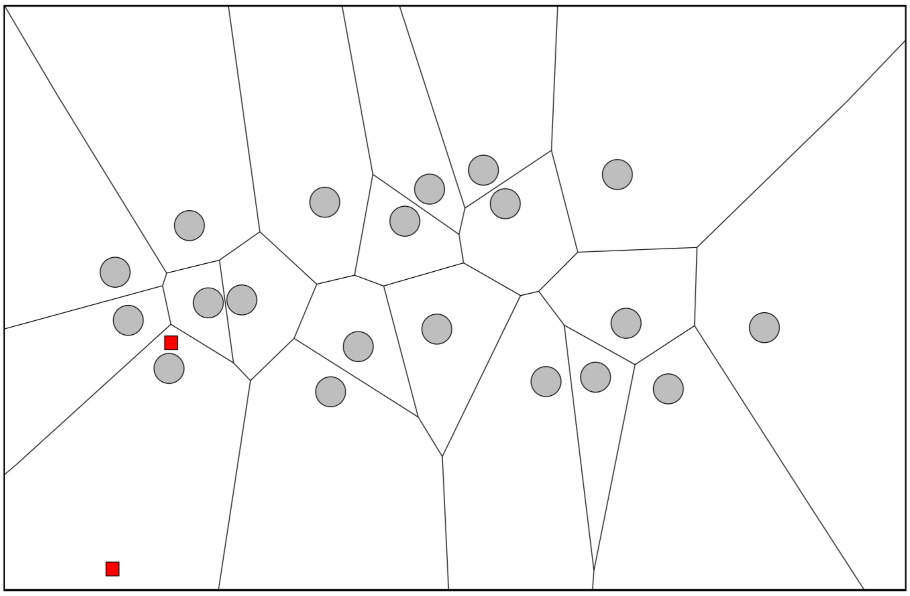
Ideally, the cell size is as small as possible, approaching zero. While the method does allow for this, eventually each cell will only intersect a single Voronoi cell. This makes the density directly proportional to the size of the Voronoi cell it is in, see Figure 8b. The strength of the method lies in the way it calculates density using multiple Voronoi cells. For this reason it is important to choose the correct cell size, large enough to cover multiple Voronoi cells in most cases, but small enough to give a local representation of density.

The method performs well for the simple hallway environments presented in the paper. However, problems occur when it is used in a more complex environment. When distance between obstacles are larger, agents at the border of a group will have large Voronoi cells. As seen in Figure 8a, this can lead to one cell having a large influence on measure density. This could be solved by limiting the influence a single cell could have. Steffen et al. [39] sets the maximum size of a Voronoi cell to a value of 2 square meters.

The original method does not deal with the presence of obstacles. Obstacles can have a large influence on the result and need to be taken into account when measuring density. Zhang [9] performs his experiments in a hallway and can thus remove any area containing obstacles from the Voronoi diagram. Alternatively, obstacles could be added as sites to the Voronoi diagram to represent agents avoiding proximity to them. However, this leads to some problems, as obstacles also get their own personal space in this case.



(a) An example of the influence of large Voronoi cells. The measurement area intersects four Voronoi cells, but if the closest obstacle on the left side of this image is far away the measured density will approach 0.



(b) An example of the complications with grid size. The resulting density in both red squares is equal, as only one Voronoi cell is taken into account.

Figure 8: Some of the issues with the Voronoi density measurement method.

5 Density Experiments

To evaluate the density measurement techniques, we have compared the performance of the three metrics described in Section 4. In this section, we discuss the experiments done to evaluate the metrics. In Section 5.1 we first discuss the setup of the experiments, followed by a description of the environments used in Section 5.2. In Section 5.3 we discuss our implementation of the density measurement techniques in the ECM Framework [60], and in Section 5.4 we evaluate the results of the experiments.

5.1 Setup

As discussed in Section 4, each density measurement method has different strengths and weaknesses. In our experiments we want to compare the qualitative performance of the methods in different situations and environments, following the desired properties as specified in Section 2.2.1. Below, we will briefly summarize the properties.

The resulting density should show a low amount of spatial and temporal variance. Measurement results from locations close to each other should not differ greatly, and density should not significantly increase from one time step to the next. If this were the case, a measurement result would not be reliable.

There should be a correlation between global density and the measured local density. The local measurement should be an approximation of the density contribution of that area to the total density. The sum of all local densities in an area should sum up to the global density.

The metric should be able to recognize local peaks in density. As observed in [38], local density for a dense crowd can be up to twice as high as average global density. Because of this, it is important to be able to recognize local peaks in density in order to correctly assess the danger in a crowd situation.

The result should be easy to interpret, ideally in terms of the number of people per square meter. This makes the density comparable to results in pedestrian dynamics. Much research has been done on critical crowd densities [1], and being able to link results to this research is an important property.

5.2 Environments

We have tested the metrics in multiple environments, testing the performance of the metrics under varying conditions.

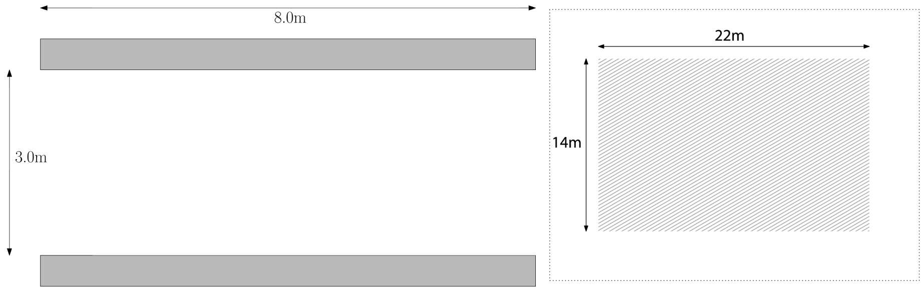
Hallway
The hallway environment as used in our experiments is shown in Figure 9a. This environment is similar to the environment used by Zhang, in [9]. Zhang analyzed a homogeneous flow through a straight corridor of 8 by 3 meters, extracted from a controlled experiment. In the experiment, 349 people walked from one side of the hallway to the other. This leads to uniform flow through the corridor, which we can replicate in a crowd simulation. The relatively large number of people

will lead to a relatively high density in an environment that is similar to real life environments in which high density can occur. A small buffer zone is present before the entrance, to ensure pedestrians enter the hallway at full speed.

We performed two sets of experiments in this environment. The first set concerns a unidirectional flow through the hallway. Four agents per second will spawn on the left side of the corridor and move towards the right, resulting in a constant flow through the hallway during the simulation. The second scenario concerns the same number of agents split up into two flows in opposite direction. The interaction between agents of opposite flows will creates a more irregular density. We test the density metrics in both scenarios, to evaluate the variance that occurs when measuring density with any of the measurement methods.

High density environment
This environment tests the quality of the methods in a high density situation, one that would occur during a crowd disaster. For this purpose, we focus on the crowd disaster during the Hajj pilgrimage of 2006. Unfortunately, no video footage of the crowd is freely available, but the density situation of the crowd is described in [38]. The disaster occurred in an open area. To avoid boundary effects, Helbing took a central area of 22 by 14 meters into account for his density measurements. Average density in this region was 5 persons/m^2. Local densities of up to 10 persons/m^2 were recorded using the Gaussian method of measuring density. With this information we can reconstruct the scenario, and compare the outcome with the measurements of the other density methods. Doing this tells us how each of the density metrics are able to deal with extremely high densities. This environment is shown in Figure 9b.

This scenario tests the effectiveness of the metrics when density is extremely high. A crowd with an average density of 5 people/m^2 is created. The focus for this scenario is on how well local density is captured, rather than spatial and temporal variance. Therefore, this scenario consists of a single snapshot from a high-density crowd of 5 people/m^2. Agents are randomly created in a 30 by 22 meter area, again to ensure agents are not influenced by boundaries. Local density is compared for the central area of 22 by 14 meters.



(a) The hallway environment. **(b)** The open environment.

Figure 9: The environments used in the density experiments.

5.3 Implementation

The environments and scenarios were implemented in the ECM Framework [60]. Simulations for the *Hallway* environments were done with the parameter settings described in Appendix B, Tables 6 and 7.

For the unidirectional hallway scenario the side preference of each pedestrian was randomized, to obtain an evenly distributed flow through the corridor. For the bidirectional scenario, agents were biased towards the right of the corridor, in order to obtain a bidirectional flow with interaction around the center of the hallway. RVO [61] was used for local collision avoidance. The time horizon for consideration of other pedestrians was set to the default value of 5 seconds.

5.3.1 Density Measurement Methods

The density methods were implemented in the ECM framework, in the following ways:

Classic Density
For our implementation of *Classic* density, we used the variation that proportionally counts agents intersecting a grid cell. For each cell, the density is computed by calculating the area of the intersection of the cell with all agents. Rather than representing the agents as a disk, the agents are represented by an octagon with area equal to the agents' disk representation. This makes calculating the intersection easier, as the result will be a convex polygon of which the area can more easily be calculated. We choose an octagon, rather than a more complicated shape, as a simplification as a disk for computational efficiency. The resulting density is divided by the area of a single agent to obtain the number of people per grid cell. The influence variable R controls the size of the octagon.

Gaussian Density
The density measurement based on Gaussian functions proposed by [38] is based on a grid of points. For comparative purposes we use the same grid setup used for the classic method, with the measurement points now lying in the center of each cell. For each point the density contribution is calculated as discussed in Section 4.2.

Voronoi Density
For our implementation of the Voronoi density method, we first computed a Voronoi diagram of the pedestrians. The diagram is computed using the ECM framework, with the pedestrians represented by point obstacles.

For our implementation of the first method described in Section 4.3, D_V, we compute the average area of the Voronoi sites intersecting the measurement area. Using boolean operations we compute the intersection of a grid cell with each of the Voronoi sites. Density in this set is then computed using Equation 5, by dividing the number of considered Voronoi cells by their total area.

The second method integrates the local density over the measurement area. We implemented this by computing the intersecting Voronoi cells in the same way as the previous method. For each of the intersecting cells, the area within the measurement area is computed. This value is then multiplied with the size of the Voronoi cell to obtain the density within the measurement area. By dividing with the area of the cell, the density in number of people per square meter is computed.

When computing a Voronoi diagram, sites at the border of the diagram will create cells extending to infinity. We addressed this by adding a bounding area that controls which part of the environment is taken into account when computing density. After computing the area of a Voronoi cell, the result is intersected with this bounding area to restrict the size of the Voronoi cells.

5.4 Results

5.4.1 Choosing Parameter Values

For each of the methods, we first test the influence of varying parameter values in order to find the settings that give the best results, for each of the methods. In this section we discuss the possible parameter settings for each method, show results, and motivate the chosen settings. We restrict ourselves to grid sizes of 0.1, 0.5 and 1 meter. Initial experimentation with grid sizes shows that grid sizes larger than 1m lack detail. The three chosen grid sizes give varying results for the resulting density, while being small enough to locally measure density.

Classic
The *Classic* method has two parameters that influence measured density: the influence range R, and the cell size of the grid. We tested influence ranges of 0.24, 0.5, and 1 meter for each of the three cell sizes. An influence range of 0.24 meter corresponds to the physical space occupied by agents. With the other two settings we test the results of a medium and large influence range.

A smaller influence range leads to high variance in the observed density. Increasing the influence range of each pedestrian diminishes the variance, but detail in the density measurement is lost, since the influence of a pedestrian is smoothed over a larger area, see Figure 10a. Increasing grid size helps reduce the variance observed when using smaller R values. Similarly, increasing R helps reduce variance due to smaller grid sizes. Because of this connection, we have decided to test two sets of parameters for the classic method: a low R-value, combined with larger grid cells, and a high R-value combined with smaller grid cells.

Gaussian
The *Gaussian* method gives a continuous representation of density, based on distance to a measurement point. In our experiments, see Figure 10b, we found that high R-values lack detail in measurement, while grid cells show large variance with lower R-values. The R-value suggested by Helbing et al. [38] produces a smooth density curve, while still preserving local peaks. One of the properties of the Gaussian density method is that grid size does not affect the measured values. For these reasons, we restrict ourselves to only one parameter setting for the Gaussian method: small grid size, with an R-value of 1 meter.

Voronoi
The *Voronoi* methods base density for a location on the Voronoi cells around this location. The only variable affecting the density is in the Voronoi sites

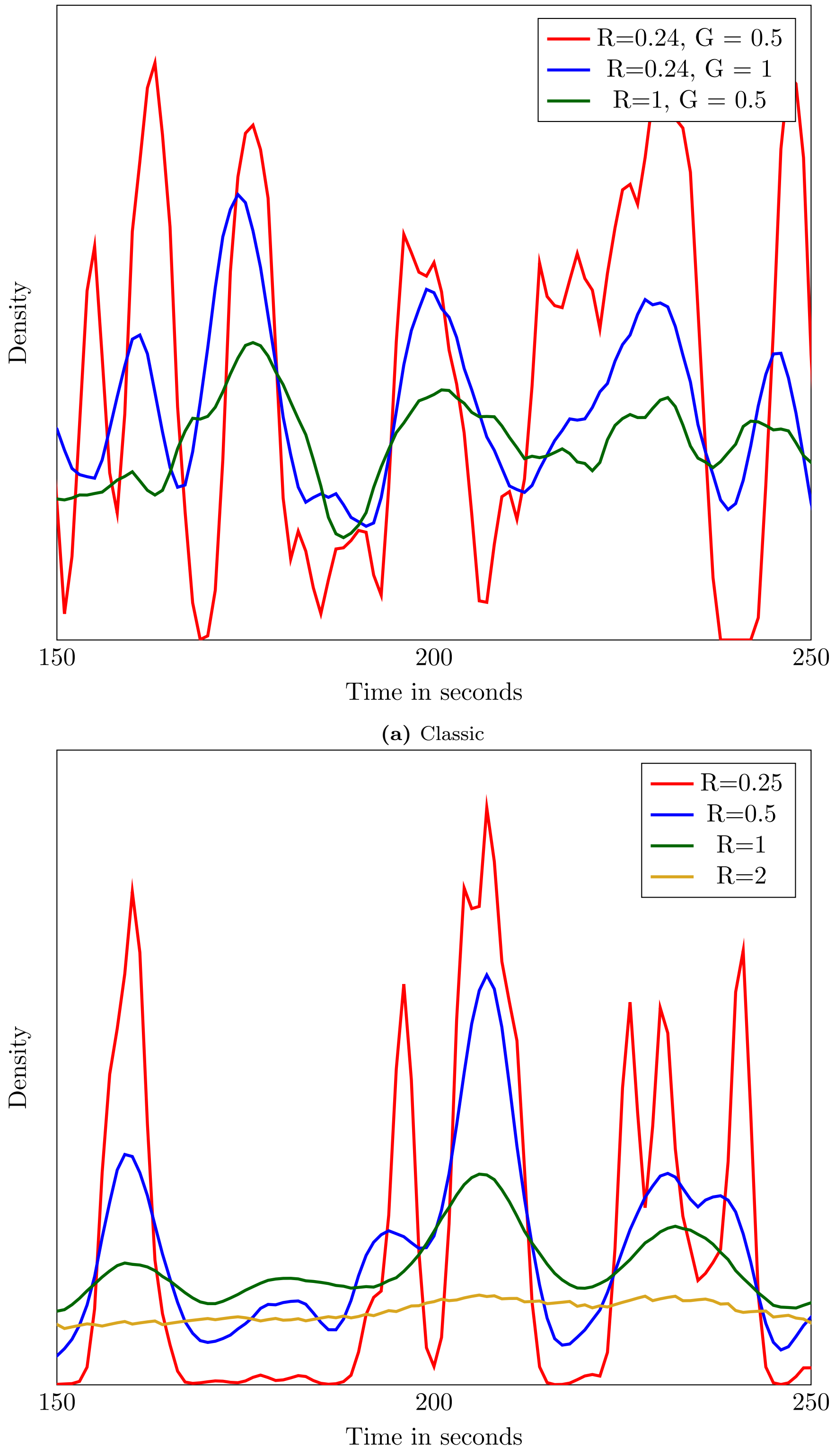
considered. Changing the grid cell size can have a large impact on what Voronoi cells are considered. The Voronoi method can be applied in two ways. The first uses a small grid size, so that for any point only a single Voronoi cell determines the density. The density for any point will be equal to the reciprocal of the area of the Voronoi cell in which the point lies. In practice this results in the density having relatively low variance, until a different Voronoi site is considered. At this point density can significantly change from one time step to the next. The second way uses a larger grid size, so that most of the time multiple Voronoi cells are considered. This leads to more sudden fluctuations in density, but the size of the fluctuations will be reduced, when compared to fluctuations with smaller grid sizes. An illustration of this can be seen in Figure 11a. As can be seen in Figure 11a, a larger grid size leads to more sudden fluctuations, but each fluctuation has reduced height, compared to smaller grid sizes. As can be seen in Figure 11b, grid size has less influence on the density computed by the *Voronoi 2* method. For this reason, we only focus on a single parameter for this method: a grid size of 0.1m. This setting should give most detail, due to its high resolution. For the *Voronoi 1* method, both the grid sizes of 0.1m and 0.5m are interesting, and give varying results. However, the grid size of 0.1m gives similar results to the ones computed by *Voronoi 2*. Since the cell size is so small, only a single Voronoi cell is considered when computing density. The size of this Voronoi cell is equal for both methods, therefore the resulting density is identical. For this reason, we only evaluate *Voronoi 2* with a higher grid size of 0.5m.

Method	Influence	Grid size
Classic 1	$R = 0.24$	$G = 1$
Classic 2	$R = 1$	$G = 0.5$
Gaussian	$R = 1$	$G = 0.1$
Voronoi 1	-	$G = 0.5$
Voronoi 2	-	$G = 0.1$

Table 2: Parameter settings used for the density experiments.

Conclusion

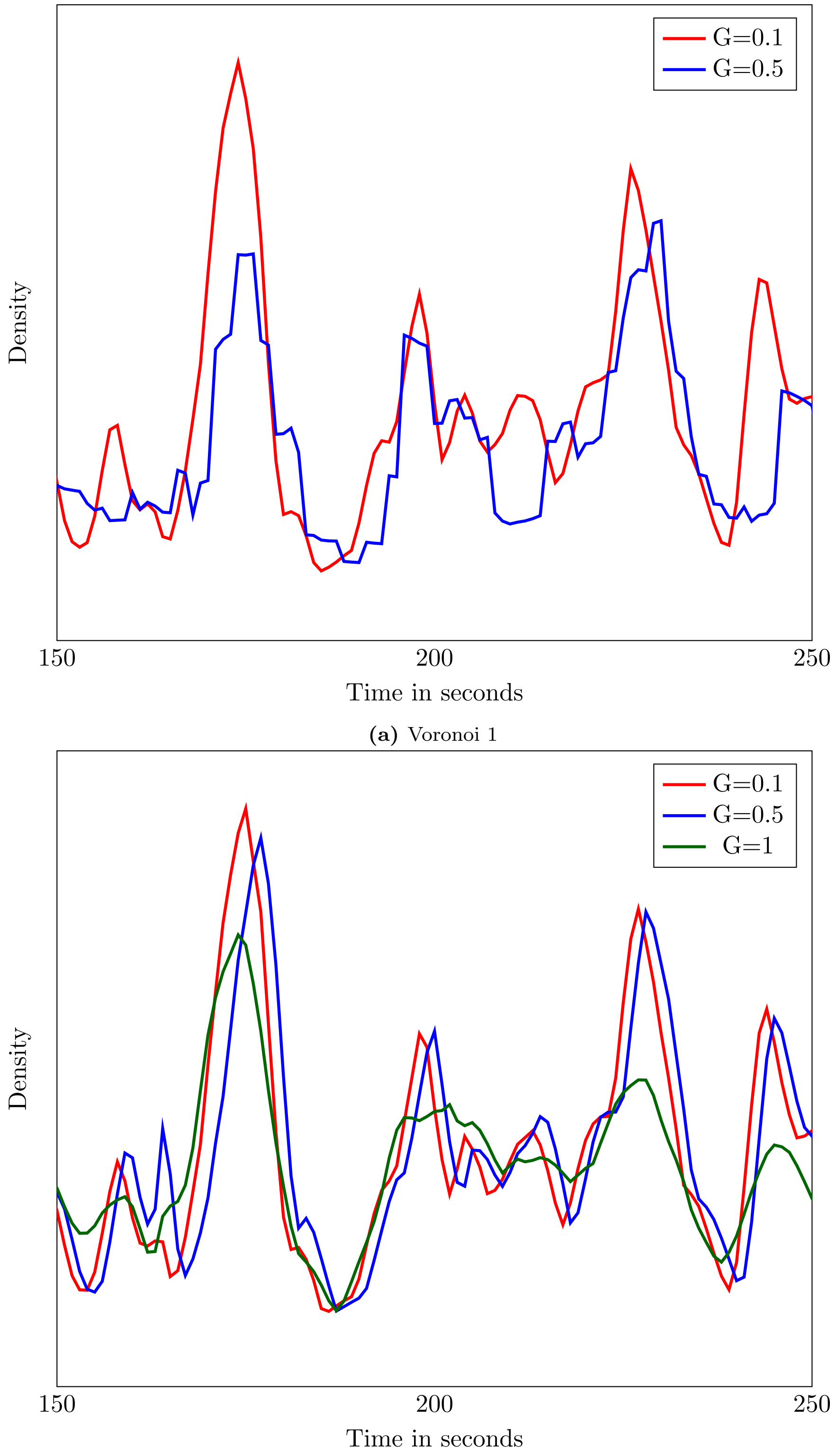
In conclusion, we test five parameter settings for the four density measurement methods, summarized in Table 2. For the *Classic* method, we use two settings. One uses a small influence range, with larger grid size. This setting is closest to the traditional way of measuring density, discussed in 2.2.1. This method is expected to have a large amount of variance. The second setting uses a larger influence range for pedestrians, with small grid size. This is expected to reduce variance, but also impacts the range of densities captured. For the *Gaussian* method, we use the setting suggested in the original paper, an influence range of 1 meter. As can be seen in 10b, this setting can capture a range of densities, while not showing too much variance. For the Voronoi methods, we focus on one setting for each of the methods. For *Voronoi 1*, we use a setting with a larger grid size of 0.5m, for *Voronoi 2* we use a smaller grid size of 0.1m. *Voronoi 2* is expected to show higher peaks than *Voronoi 1*, but less frequently.



(a) Classic

(b) Gaussian

Figure 10: The effect of varying parameters for the Classic and Gaussian density methods.



(a) Voronoi 1

(b) Voronoi 2

Figure 11: The effect of varying parameters for the Voronoi density methods.

5.4.2 Representation of Global Density

One of the criteria we observed in Section 2.2 was that the sum of local densities in an area should be similar to the global density. This is important, since local density should be an approximation of the local contribution of a certain subarea to average density. To verify this, we computed density for each environment and summed up the results. We compared this to the number of people present in the entire environment, for each time step. The result for the bidirectional hallway scenario is shown in Figure 12. The results for the other environments are included in Appendix B.

The total local density agrees well with the global density, for each of the chosen parameter settings. It can be observed that the summed up local density is often slightly lower than the global density for each of the methods, but this can be explained by the boundary effect. It is important that the local measurement methods recognize the same density spikes that occur for globally measured density. For the *Voronoi* method, the spikes are slightly reduced in size, due to the reduced scatter inherent to the method. They are still present however, so it can be said that each method is an accurate representation of global density.

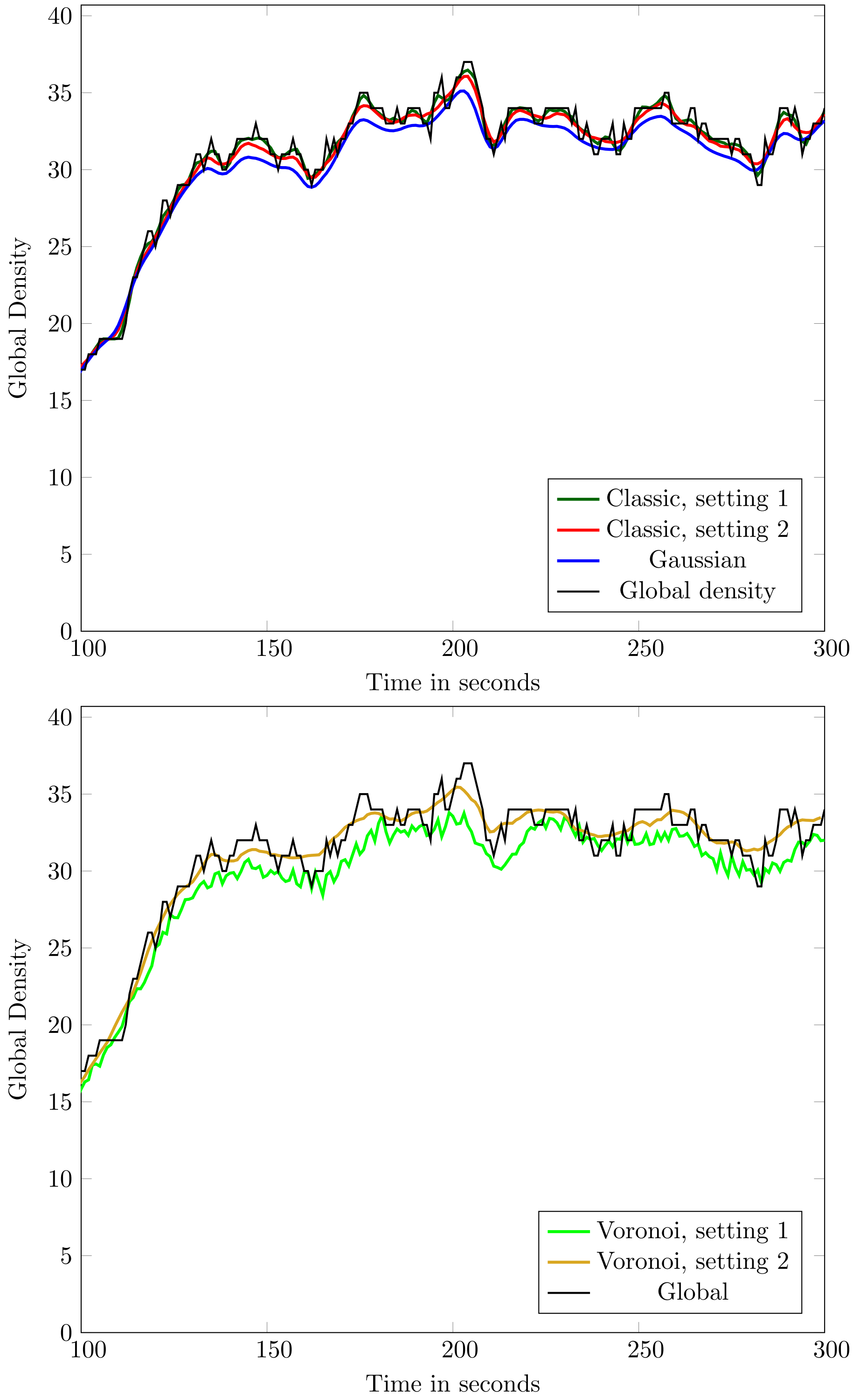


Figure 12: Global density for each of the measurement methods in the hallway scenario.

5.4.3 Variance Results

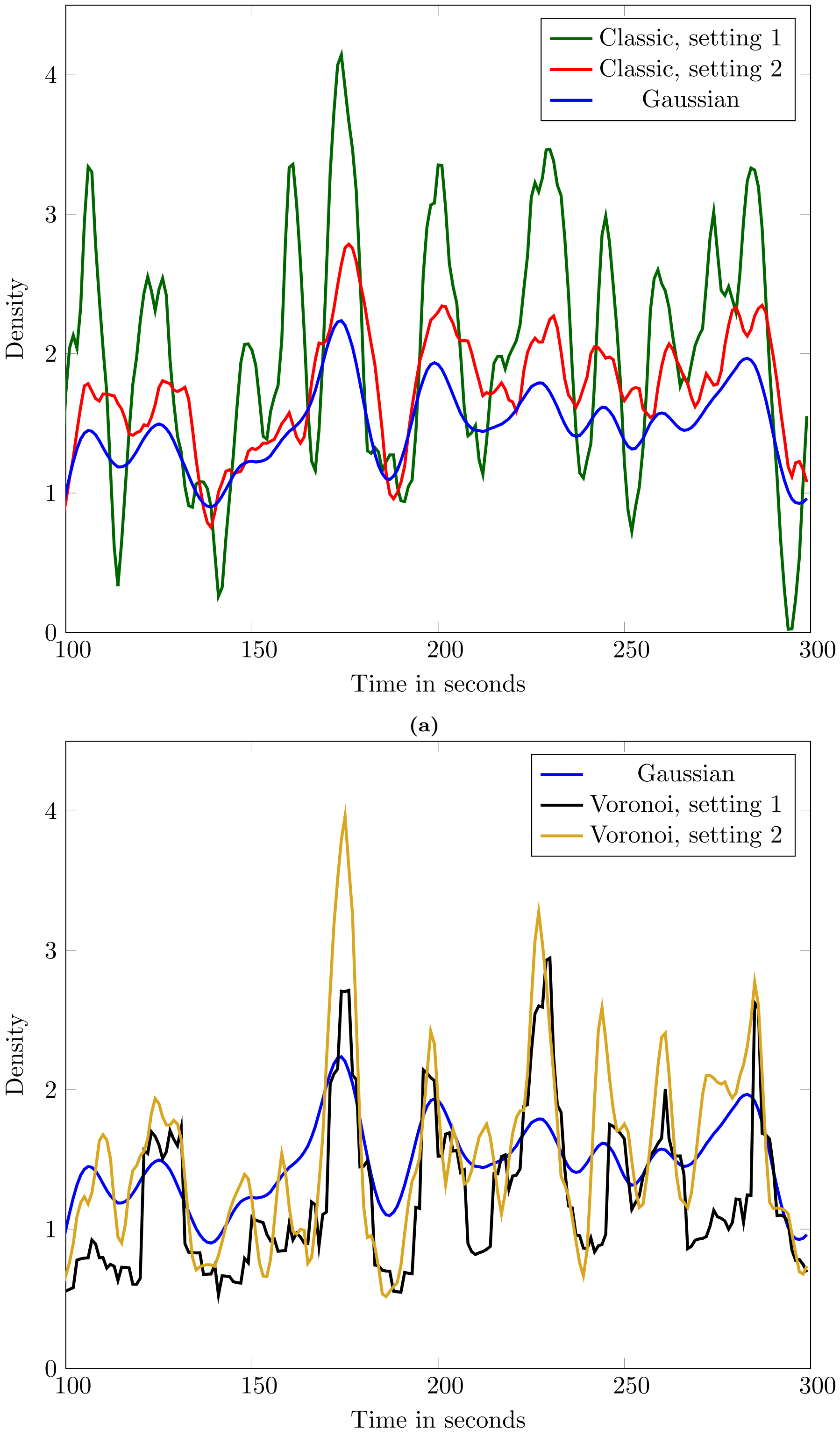
We test the variance of the density methods in three ways:

- **Over time**
  We track a single location in the environment, and compare the temporal variance in density measures for each measurement method. We choose the grid cell in the center of the hallway as reference point.

- **Over space**
  For a single time step, namely the one with highest global density, we calculate the spatial variance of each method. A cross section of the environment is taken, and variance is calculated for the resulting grid cells.

- **From an agent perspective**
  For some applications, the local density that at the agents' location is required. For example, when validating a simulation with the fundamental diagram, you would compute the local density perceived by an agent, and compare it to its speed and the fundamental diagram. For this reason, it is important to determine the performance of the different measurement methods, when using them in this way. To determine this, we track the variance a single agent experiences according to each measurement method.

The variance of the methods is quantified by computing the standard deviation from the average density. The full results are shown in Tables 8 and 9. A summary of the results is shown in Tables 3 and 4. In these tables, the standard deviation is divided by the mean density, to obtain the coefficient of variation.



(a)

(b)

Figure 13: Variance over time in the hallway environment.

Method	Temporal	Spatial	Agent
Classic 1	43.6%	59.5%	44.5%
Classic 2	23.7%	27.8%	13.1%
Gaussian	19.9%	20.7%	9.8%
Voronoi 1	44.0%	32.1%	31.4%
Voronoi 2	43.6%	40.4%	28.4%

Table 3: Coefficient of variation for the Hallway scenario

Method	Temporal	Spatial	Agent
Classic 1	67.9%	58.1%	23.0%
Classic 2	28.4%	30.7%	27.6%
Gaussian	25.9%	26.8%	22.3%
Voronoi 1	38.8%	33.7%	30.0%
Voronoi 2	29.9%	40.5%	37.5%

Table 4: Coefficient of variation for the Bidirectional Hallway scenario

Variance Over Time

In this experiment we tested variance over time for a specific point in space. As reference point we took the cell in the center of the corridor. The results can be seen in Figure 13a for the unidirectional environment, and in Figure 32 for the bidirectional environment. Based on these graphs, and the quantified variance shown in Tables 3 and 4 we can make some general observations about temporal variance.

Classic 1 shows a large amount of variance for both environments. Measurements with this method are either very high or very low. It is hard to get a reliable measurement for a location when the values spike like this.

Gaussian and *Classic 2* perform very similar. *Gaussian* is smoother but loses some detail to *Classic 2*. This is inherent to the methods' design. Both methods are based around an influence range, set to 1 meter. The difference lies in the *Gaussian* method taking every pedestrian into account, rather than only those within 1 meter.

It can be observed in the graph that *Voronoi 1*'s grid size makes density spike more often, but less high than *Voronoi 2*. Both methods provide similar values to the *Gaussian* method, with a higher range. This increases the detail of measurements, but can also lead to artifacts where density suddenly spikes high.

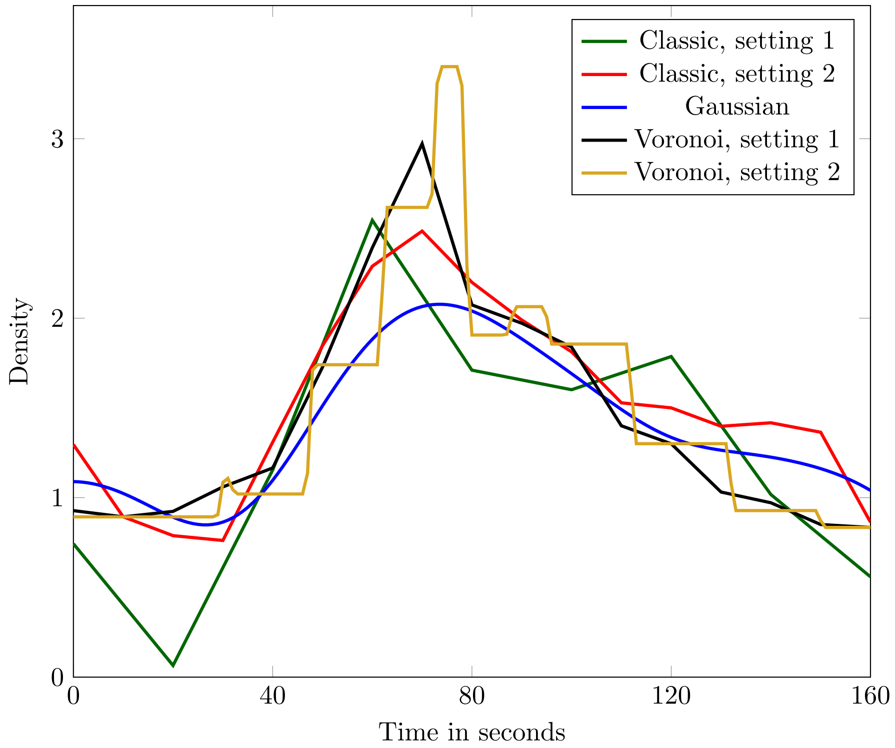


Figure 14: Variance over space in the Bidirectional Hallway scenario

Variance Over Space

In this experiment we tested spatial variance in the hallway environments. To test this, we measured the density at the center of the corridor along the entire width. The results can be seen in Figure 14 and Figure 33.

It can be observed that all five methods show similar ranges and recognize the same density peaks. *Voronoi 1* is the largest outlier, recognizing a valley around time 100. This is a result of the problem described in Section 4.3, where a large Voronoi site can influence a density measurement. Other than this, the measurement methods follow similar curves. Again, *Classic 1* has the most unstable measurements by far. The large grid size means density values will show significant variance. The curve of *Voronoi 2* shows many sudden jumps in measured density. Due to the small grid size, only a single Voronoi cell is taken into account for each cell. This results in equal density measures for neighboring cells considering the same Voronoi cell, and highly different ones when this is not the case. The *Gaussian* curve shows the least amount of variance, but lacks detail. It does not recognize the peaks and valleys that other methods find, but instead results in a smooth approximation. *Classic 2* finds similar values to *Gaussian*, at the cost of some amount of smoothness.

It is interesting to see that all methods show less spatial variance in the bidirectional hallway scenario. This can be explained by the higher densities, leading to a more constant flow through the environment. *Voronoi 2* especially performs better in higher density situations, due to the considered Voronoi sites being smaller on average. This leads to the problems discussed in Section 4.3 having less impact.

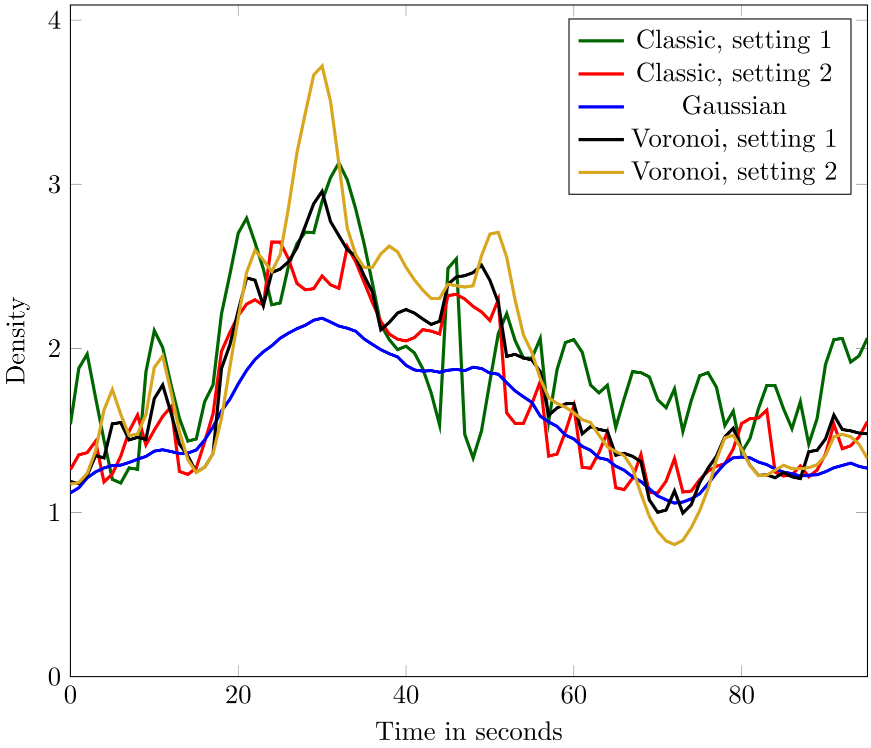


Figure 15: Agent variance in the bidirectional scenario.

Agent Variance

We tested agent variance by tracking the density measured at the location of a single agent, over the course of the simulation. We display the results of the agent with highest perceived density across all five methods in Figures 15 and 34. We can see from the curves in Figure 15 that all methods recognize similar density ranges. Since density is measured from an agent perspective, the minimally measured density for this test lies around 1.

The *Voronoi* method shows a low amount of variance compared to earlier. This is because the method is designed around measuring density from a pedestrians' perspective. Especially for the method with smaller grid size, this leads to good results. Since *Voronoi 2* often only considers a single Voronoi cell, the measured density evolves smoothly as the agent moves. *Gaussian* and *Classic 2* perform very similar, with *Gaussian* having more smoothness, but lacking detail. The *Gaussian* method only recognizes densities between 1 and 2 for the bidirectional hallway environment. Both *Classic* methods show spiking whenever the cell that contains the agent changes.

5.4.4 High Density Environment

In the high density environment we test the ability to capture local density, for each measurement method. We created a static crowd with an average global density of $5/m^2$, in an area of 30 by 22 meters. Each agent is assigned a random free position within the area, leading to a total of 3300 agents over 660 m^2. We measure density in the central area, leaving a 4 by 4 buffer area on the sides to prevent boundary effects. A snapshot of the crowd is shown in Figure 16.

Heatmaps of the density computed with each of the measurement methods are shown in Figure 17. Histograms of the density distribution are shown in Figures 18 and 19. It can be observed that the metrics recognize the same average density for the environment, around 5 people/m^2. This is in accordance with the global density the area is designed to have. Just as observed in the previous sections, *Classic 2* and *Gaussian* yield similar values. *Voronoi 2* shows a large amount of detail compared to the other models, due to the low grid size. *Gaussian* uses the same resolution, but capures less detail. *Classic 1, Voronoi 1*, and *Voronoi 2* are able to capture a wide range of densities in this dense environment. The *Gaussian* method only recognizes densities ranging from 3 to $6.5/m^2$, however. *Classic 1* and *Voronoi 1* both show discontinuities in the distribution. This is an effect of the variance of both methods, earlier observed in the hallway environments.

In general, we make the same observations in this high density environment as we did in the previous ones. *Gaussian* and *Classic 2* perform similar, and return smooth density. However, this smoothness comes at the cost of detail. The *Classic* method, and both *Voronoi* methods capture more detail, leading to more precise measurements of local density. Both *Classic* methods, and the *Gaussian* method show discontinuities in the density distribution. For the *Voronoi* methods, this is less the case.

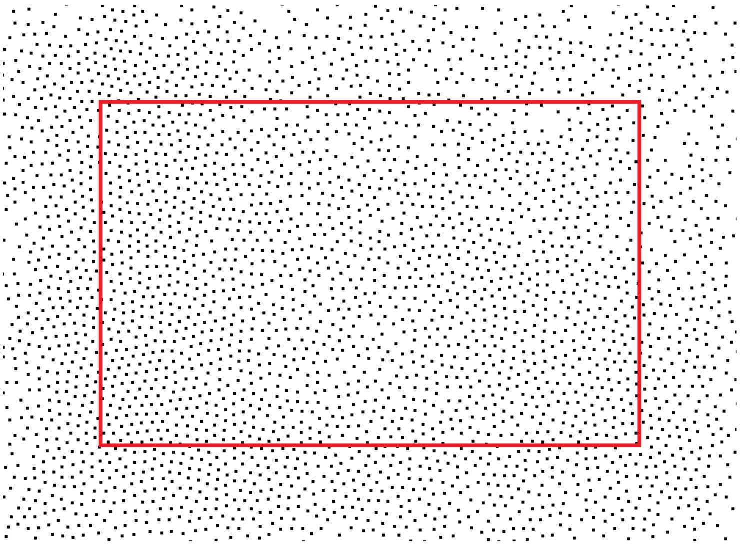
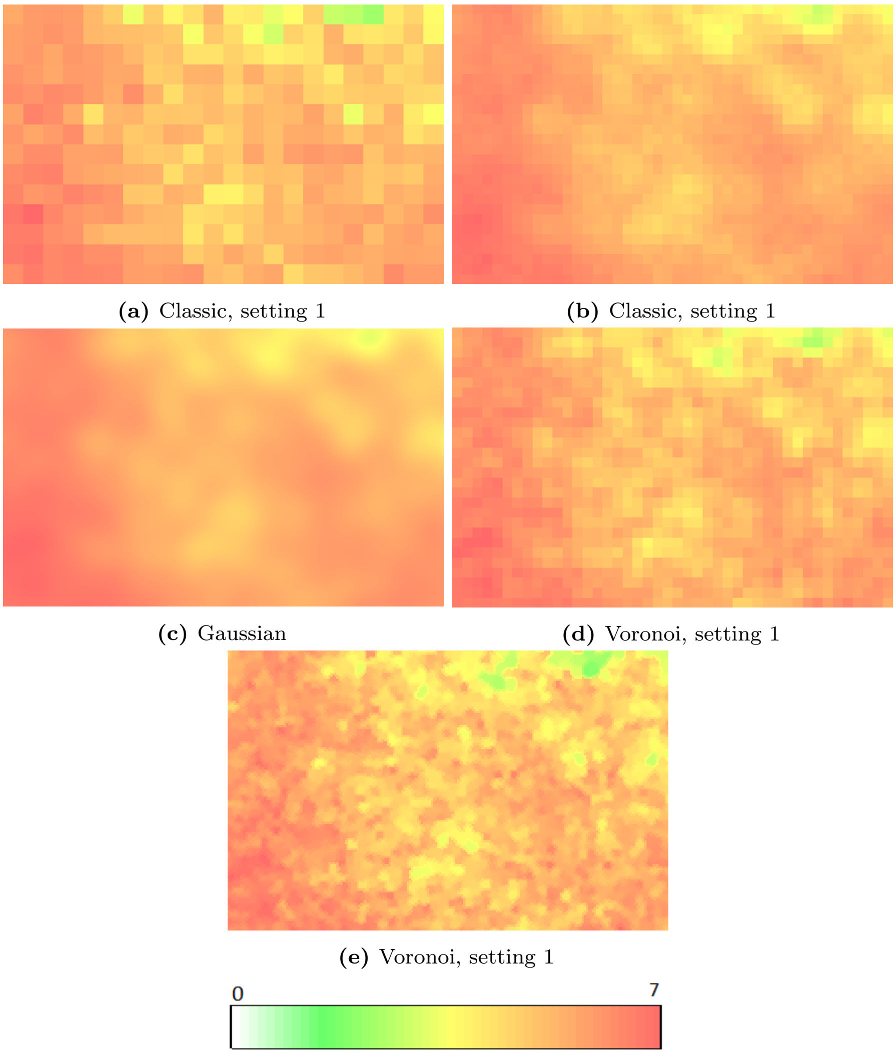


Figure 16: A snapshot of the dense crowd.


(a) Classic, setting 1

(b) Classic, setting 1

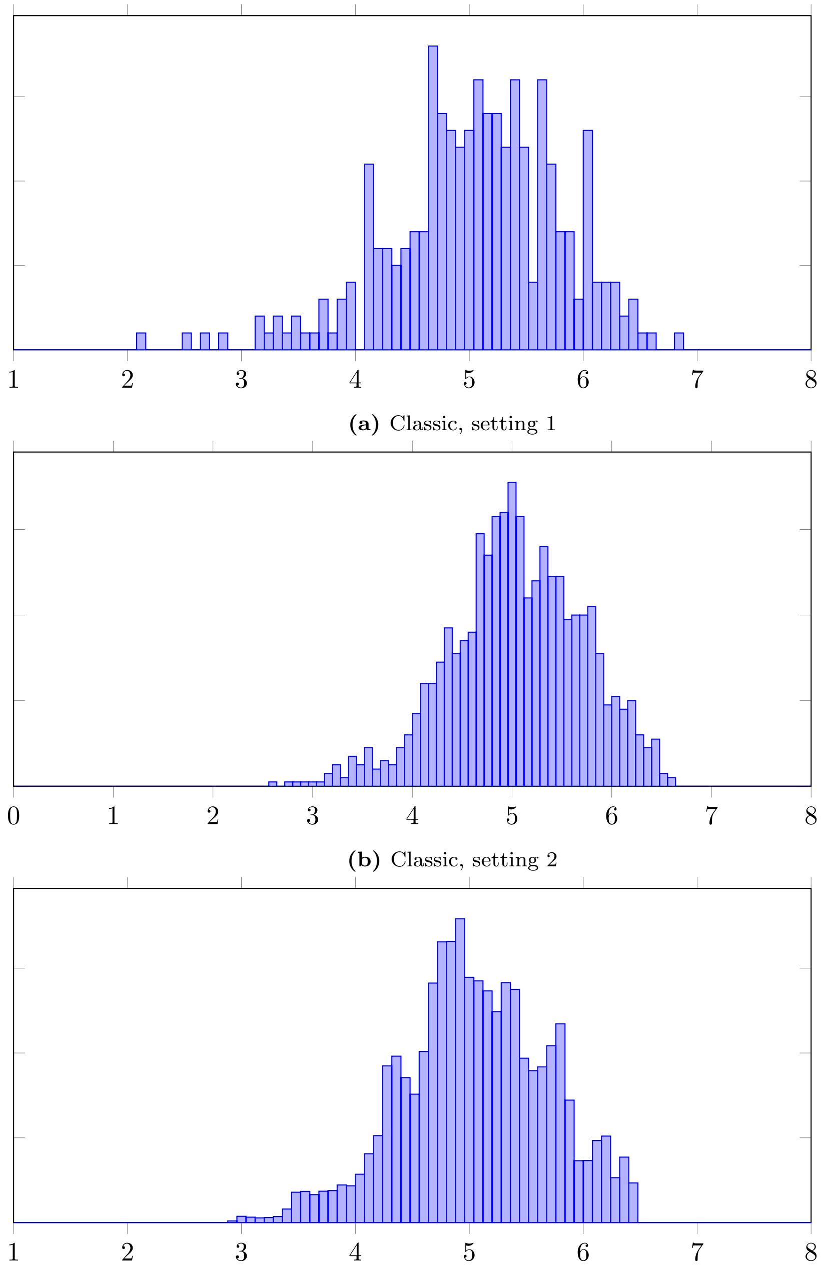
(c) Gaussian

(d) Voronoi, setting 1

(e) Voronoi, setting 1

(f) The density scale used.

Figure 17: The resulting density map for the different measurement methods. The density ranges from 3 to 8

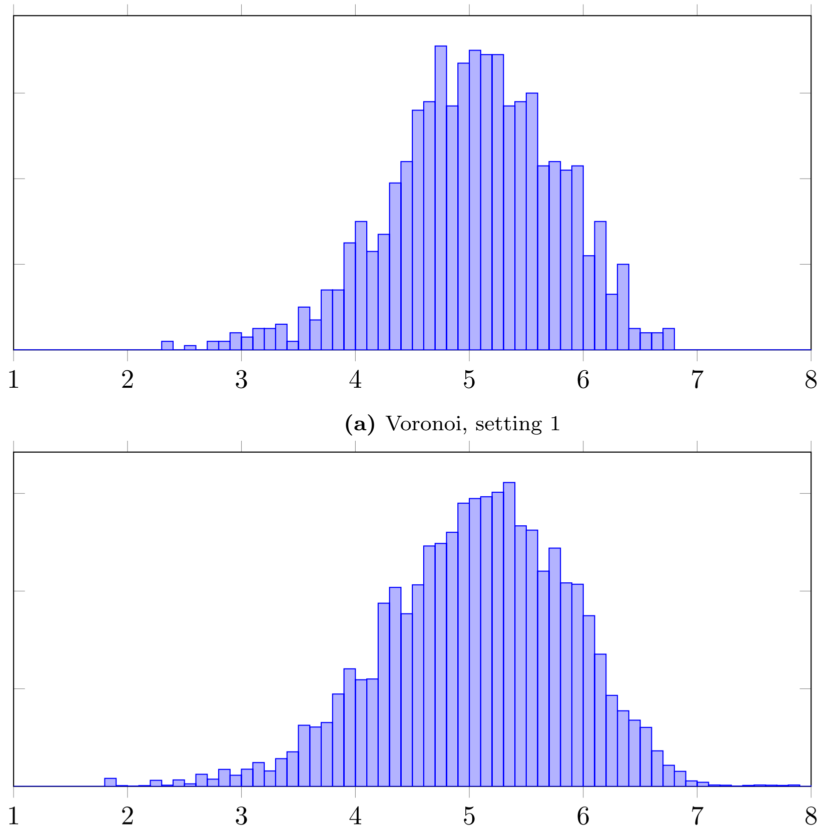


(a) Classic, setting 1

(b) Classic, setting 2

(c) Gaussian

Figure 18: Histograms of the density measures in the dense environment.



(a) Voronoi, setting 1

(b) Voronoi, setting 1

Figure 19: Histograms of the density measures in the dense environment.

5.5 Analysis

Each of the methods are able to represent global density well. When averaging density over a larger area each method gives results similar to the amount of people that can be counted inside that area. For each of the scenarios we experimented in, the *Classic* method with larger grid size and the Gaussian method give similar results. This can be explained by the similar design inherent to the methods. While the *Gaussian* method reduces the influence smoothly however, the *Classic* method only considers agents actually intersecting the measurement cell. This leads to lower variance for the *Gaussian* method, at the cost of some detail. This is especially noticeable when measuring density in the high density environment.

For the medium-density environments the *Classic 1* setting shows a large amount of variance. Due to the small influence range of 0.24m it only performs well in very dense environments, where a larger amount of people are contained within a single measurement cell. For dense environments, *Classic 1* is able to capture a wide range of densities. Gaussian and *Classic 2* are smoother but miss capturing some of the lower and higher densities.

Voronoi 2 performs well when measuring density from a pedestrian perspective. Due to the low grid size, only a single Voronoi cell is considered when computing density. This leads to the density being a direct reflection of the personal space of that pedestrian. Since only a single Voronoi cell is taken into account however, density suddenly spikes when the considered Voronoi cell changes. This leads to poor performance regarding spatial and temporal variance.

Classic 1, and both *Voronoi* methods show less variance in the bidirectional environment than in the unidirectional one. These methods perform better in higher-density environments than in ones with varying density. This observation can be confirmed with the high density test. The *Gaussian* method and *Classic 2* show less variance in environments with lower density, but capture a more narrow range of densities.

In conclusion, the *Classic* method has two viable settings, each with its own strengths. A setting with larger grid size and small influence range performs well for high-density environments, but is unable to capture lower densities without a huge amount of variance. A setting with smaller grid size, but larger influence range for pedestrians performs well for all environments, but has a more narrow density range than other methods. Similarly, the *Gaussian* method shows a low amount of variance for any environment, but cannot recognize low and high densities as well. The *Voronoi* method is able to capture density well from a pedestrian perspective. However, this comes at the cost of high spatial and temporal variance for lower densities. In higher densities, the *Voronoi* method is able to capture detailed local densities, unlike the *Gaussian* and *Classic 2* methods. The choice for grid size is a trade-off between spatial and temporal variance. For a larger grid size the method shows less variance over time, at the cost of variance over space.

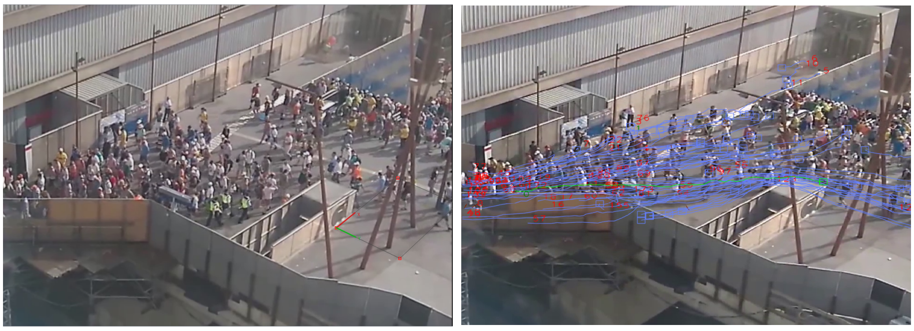
6 Fundamental Diagram From Data

As discussed in Section 2.1.1, many different fundamental diagrams are used for validation. In this section we try to show, by means of an example, that metric choice has a large influence on the created fundamental diagram. We create a fundamental diagram using each of the density measurement metrics discussed in Section 4, and compare the results with one another.

6.1 Input Video to Data

We use video footage shot at the exit of Utrecht's central train station to create a fundamental diagram. The video is footage of a medium density bottleneck scenario, with bidirectional flow. The length of the clip used is 20 seconds. The pedestrian trajectories are manually extracted from this video using a tool written by colleague Simon Rosman. The camera is calibrated using QtCalib, with a ground plane as shown in Figure 20a. The 2D position of pedestrians in the video are tracked by marking the position of their feet, resulting in a set of trajectories as shown in Figure 20b. The recorded 2D position is then converted into 3D by reversing the camera projection, using the technique described in [46].

The position of each pedestrian in the video was tracked every 0.4 seconds, resulting in a total of 4770 samples over 172 trajectories.



(a) The calibration used.

(b) The tracked crowd.

Figure 20: Pedestrian tracking

6.2 Fundamental Diagram From Input Data

Our approach is based on the experiments done by Zhang [9], and by Steffen and Seyfried [39]. From video footage of controlled experiments within a hallway, Zhang creates fundamental diagrams for different environments. Zhang choose a specific area for measurements, to decrease the effect of entering and exiting the hallway. From experimentation with the measurement area, it is concluded that the size and position of the measurement area affects the measured density, mostly due to the boundary effect. Close to obstacles, calculated density values are influenced by the presence of geometric boundaries.

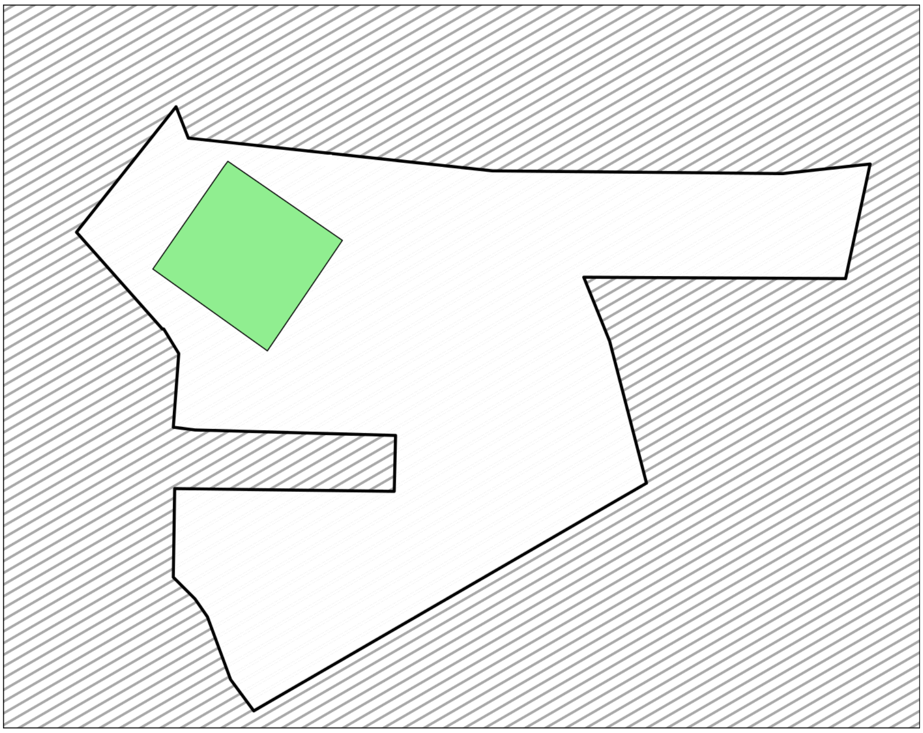
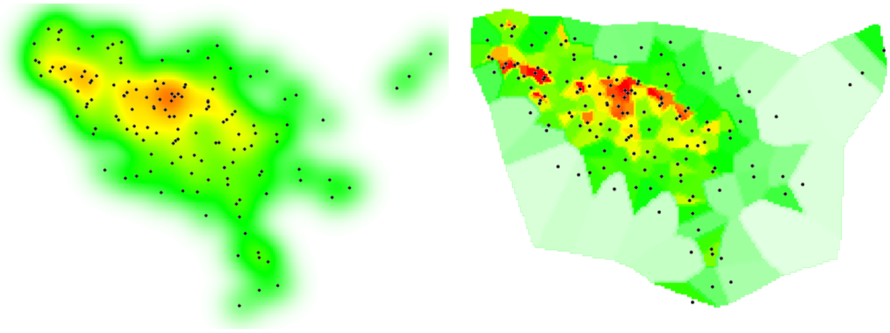


Figure 21: The created environment. The measurement area is shown in green.



(a) Gaussian

(b) Voronoi, setting 2

Figure 22: Snapshots of the density measurements in the tracked crowd. The other methods are shown in Figure 38.

We recreated the environment from the video. The result is shown in Figure 21. In order to minimize the boundary effect, we chose our measurement area to be in the center of the corridor. For the crowd extracted from the video, discussed in Section 6.1, we compute density for each time step of 0.4 seconds. The resulting density maps are shown in Figures 22 and 38. For each time step, density and speed are calculated for the pedestrians within the measurement area. Speed per time step is calculated by dividing the distance traveled by the time difference, to obtain a speed in meters per second. Density is computed for each individual pedestrian by finding the density for the grid cell corresponding to the agents' position.

6.2.1 Results

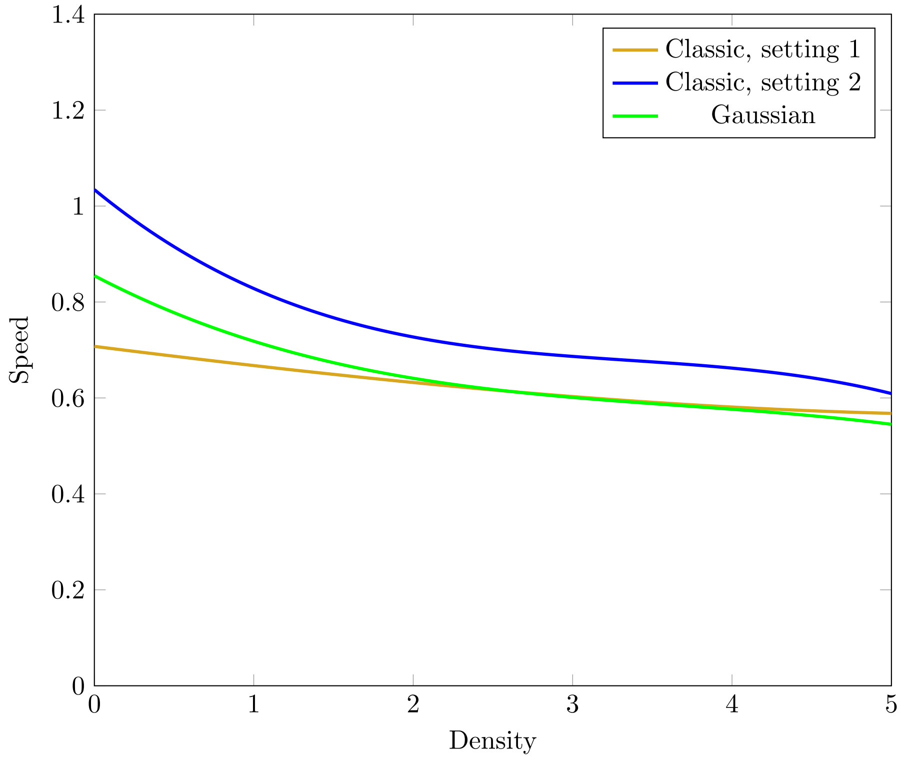


Figure 23: An example of a fundamental diagram of the relation between density and speed, taken from [14].

6.2.2 Analysis

The fundamental diagrams created from the video data are shown in Figures 23 and 24. These graphs are created by fitting a third degree polynomial curve through the scatter graphs of (speed, density) pairs. These scatter graphs are shown in Figures 35, 36, and 37.

In the scatter graphs of the fundamental diagrams, we can recognize the properties discussed in Section 5.5. *Classic 1* shows a large amount of scatter in measured density. *Classic 2* and *Gaussian* capture a limited range of densities compared to the other methods. *Classic 2* has very limited scatter for the entire density range. The *Gaussian* method shows a high amount of outliers, especially for higher densities.

The *Voronoi 2* method is able to capture the shape of fundamental diagrams the best, of the five methods. Especially for lower densities, fundamental diagrams

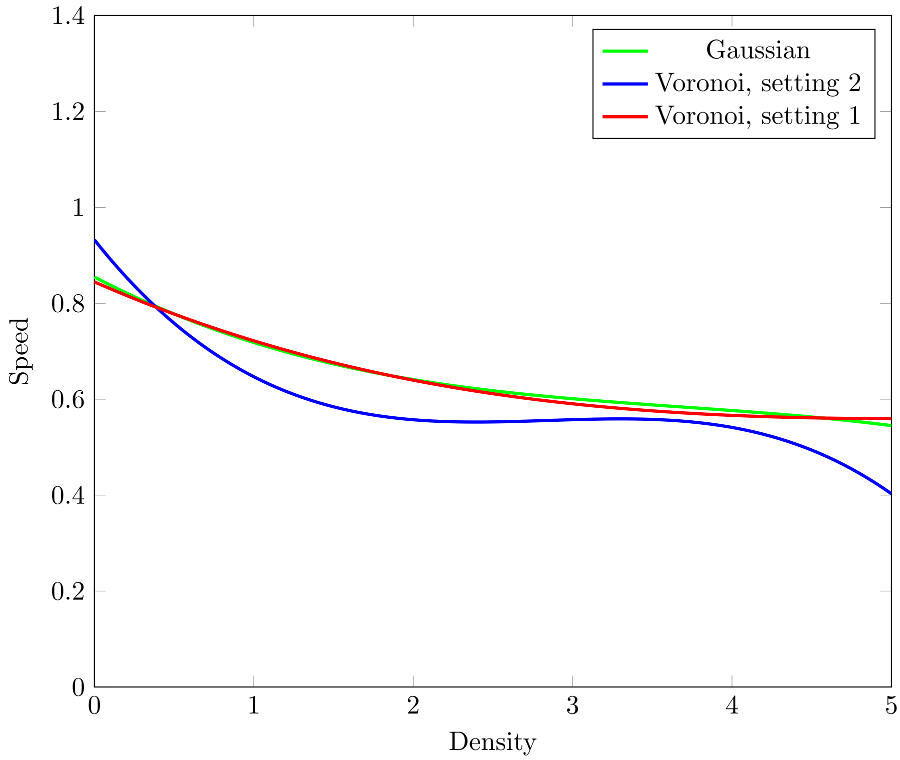


Figure 24: The fundamental diagrams obtained from the scatter graphs in Appendix C.

are well reflected with this method. *Classic 2* also results in a curve that follows the shape of the fundamental diagram. *Classic 2* also has a significantly better fit to the data than *Voronoi 2*, due to the low amount of outlying values. *Classic 1*, *Voronoi 1*, and *Gaussian* have too many outliers to result in a curve representing a fundamental diagram. In both *Classic 2* and *Voronoi 2* the characteristic shape of the fundamental diagram can be observed.

The crowd situation used for this experiment mostly contained densities of medium value. This is reflected in the resulting fundamental diagrams. Fundamental diagrams from literature commonly start at a speed of around 1.4m/s, and range until 0 m/s. In our environment, we only observed densities of up to 3.5 people/m^2. For this reason, the created fundamental diagrams lack high-density information. Measured speed only ranged from 0.4 to 1m/s. It can be observed in Figures 23 and 24 that the fitted functions agree reasonably well for medium densities, but disagree for high and low measurements.

7 Conclusion

In this section, we first summarize our findings in Section 7.1. We then discuss some of the limitations of our work in Section 7.2, and give suggestions for future work in Sections 7.3.1 and 7.3.2.

7.1 Summary

In this thesis, we have analyzed the effect of different density measurement metrics on the fundamental diagram. We looked at the performance of three different measurement techniques, Classic, Gaussian, and Voronoi, and compared it against properties a density metric should have. We performed experiments on medium- and high-density environments, and can conclude strengths and weaknesses of each method from this. The Gaussian method shows the least amount of variance across all scenarios, but this comes at the cost of detail. The Voronoi method shows more variance over space and time, but is able to measure local densities better. For the Classic method, a higher grid size and influence range decrease variance, but also decrease the resolution of the measurements with it.

The Voronoi method, and the Classic method with larger grid size show less variance in dense environments. The Gaussian method, and the Classic method with smaller grid size perform better in environments with lower densities.

These results were confirmed by the experiments in Section 6, where we created a fundamental diagram from real crowd data. The best results were obtained with the *Voronoi 2* and *Classic 2* methods. The other settings lacked resolution, or showed too much scatter. The video used for this experiment mostly contained medium densities however. More experimentation, with more extensive footage, needs to be done to verify the effects for higher and lower densities.

Fundamental diagrams are often used as a means of validating crowd simulations. However, diagrams exist for many different situations, and are dependent on a large number of variables. One of these variables is the chosen metric for density measurement. In much of the literature, the choice for a measurement metric is not motivated. As shown with the experiments however, choice for a density metric can lead to different fundamental diagram. With the growing world population, and the inherently increasing crowd risk, it is important that simulation validation is done in a transparent, repeatable, and verifiable way. Currently, this is too often not the case, leading to preventable crowd disasters, and an increased safety risk for visitors of large-scale events.

7.2 Limitations

Initially, the aim of this project was to validate crowd simulations using video data shot during the Grand Départ in Utrecht. This would allow for validation with a large amount of real crowd data. Converting the available footage to usable data turned out to be harder than expected. In the end, we manually tracked pedestrian locations in a shorter video, to use this data as a proof of concept. However, the amount of data was not sufficiently varied to allow for validation. For this reason, we shifted our research to the effect of density measurements on the fundamental diagram.

In Section 5, we tested density measurement methods on two environments, a hallway, and an open high-density environment. It would be relevant to do further testing on different kinds of environments, such as T-junctions and open environments. Our high-density experiment used a snapshot of a high-density situation. It would be nice to test performance over time of measurement methods in this situation.

In Section 6, we have measured speed as the speed of a single pedestrian at a given time. With a larger amount of data, we could use average speed in a certain area. This would reduce the variance, and likely give a more accurate fundamental diagram.

7.3 Future Work

In this section we give suggestions for future work on the validation of crowd simulations.

7.3.1 Measuring Flow

Flow measurements suffer from similar problems to density. For one-dimensional environments, such as corridors, measuring flow is straightforward. Flow is defined as the number of people passing a point in space, over a certain time period. As flow is dependent on the past situation, it can be complicated to define flow for a specific point in time. Furthermore, measuring flow gets more complicated when a second dimension is added. Directions and crossing flows have to be taken into account when dealing with two dimensions. We have performed a comparative analysis on the density measurement methods at the basis of fundamental diagrams. We focused on the *density-speed* version of the diagram, but a similar analysis could be done for *density-flow*. Much research has been done on the effect of increased density on the risk in a crowd [1]. However, for flow this effect is less straightforward. The risk depends on the pressure to keep moving, and the nature of the crowd. This can be hard to measure, especially in environments other than corridors.

7.3.2 Validation

As a possible way of validating crowd simulations, we propose to combine two of the methods described in 2.1. The parameter optimization method of Wolinski [11] can be extended with the agent state database of Charalambous and Chrysanthou [21]. Rather than focusing on individual metrics, we suggest to focus on using group-based metrics, also used in fundamental diagrams.

Using the optimization approach of Wolinski, the complex problem of validating a simulation to input data can be reduced to one with a limited amount of parameters. Parameter optimization algorithms allows for a more complete exploration of the solution space than manually experimenting with parameter settings. This allows for an objective and repeatable way of matching a simulation to specific input data.

In [22] Charalambous et al. perform outlier detection on a crowd, using a database of crowd states. A given input video is parsed into a set of trajectories. These trajectories are then segmented into segments of equal length. For each trajectory segment, a number of user-defined metrics are computed on three levels: individual, between two agents, and for a group of agents. A distance function determines the distance between any two segments, given the chosen metrics. When querying a simulation segment against the database, the anomaly score is determined by looking at the closest k segments in the database. The distance to the k-th neighbor determines whether the queried trajectory segment can be explained with data, or is showing outlying behavior.

Wolinski's approach is able to tune an input scenario to real data, but uses a basic representation of agent state. Due to the varying nature of crowd scenarios, the reference data is often not reusable, and specific data would have to be available for every simulation. Charalambous and Chrysanthou's approach can recognize unnatural behavior, but gives no suggestion on how to fix this. By combining the approaches, both methods' weak sides are tackled. The combined method should be more flexible and give better performance.

References

[1] G.K. Still. *Introduction to Crowd Science.* CRC Press, 2014.

[2] B.D. Greenshields. A study of highway capacity. In *Proceedings Highway Research Record.*

[3] W.L. Wang, S.B. Liu, S.M. Lo, and L.J. Gao. Passenger ship evacuation simulation and validation by experimental data sets. *Procedia engineering*, 71:427–432, 2014.

[4] M.L. Isenhour and R. Löhner. Verification of a pedestrian simulation tool using the nist recommended test cases. *Transportation Research Procedia*, 2:237–245, 2014.

[5] F.P. Navin and R.J. Wheeler. Pedestrian flow characteristics. *Traffic Engineering, Inst Traffic Engr*, 39, 1969.

[6] S.J. Older. *Movement of pedestrians on footways in shopping streets.* Traffic engineering & control, 1968.

[7] B.D. Hankin and R.A. Wright. Passenger flow in subways. *OR*, pages 81–88, 1958.

[8] D. Oeding. *Verkehrsbelastung und Dimensionierung von Gehwegen und anderen Anlagen des Fußgängerverkehrs.* Bundesminister für Verkehr, Abt. Strassenbau, 1963.

[9] J. Zhang. *Pedestrian fundamental diagrams: Comparative analysis of experiments in different geometries*, volume 14. Forschungszentrum Jülich, 2012.

[10] J.J. Fruin. *Pedestrian planning and design.* Metropolitan Association of Urban Designers and Environmental Planners, 1971.

[11] D. Wolinski, S.J. Guy, A-H. Olivier, M. Lin, D. Manocha, and J. Pettré. Parameter estimation and comparative evaluation of crowd simulations. In *Computer Graphics Forum*, volume 33, pages 303–312. Wiley Online Library, 2014.

[12] S. Lemercier, A. Jelic, R. Kulpa, J. Hua, J. Fehrenbach, P. Degond, C. Appert-Rolland, S. Donikian, and J. Pettré. Realistic following behaviors for crowd simulation. In *Computer Graphics Forum*, volume 31, pages 489–498. Wiley Online Library, 2012.

[13] S. Curtis. *Pedestrian velocity obstacles: Pedestrian simulation through reasoning in velocity space.* PhD thesis, The university of North Carolina at Chapel Hill, 2013.

[14] V.M. Predtechenskii, A. Milinskiĭ, and Ivanovich. *Planning for foot traffic flow in buildings.* National Bureau of Standards, US Department of Commerce, and the National Science Foundation, Washington, DC, 1978.

[15] S.J. Guy, J. van den Berg, W. Liu, R. Lau, M.C. Lin, and D. Manocha. A statistical similarity measure for aggregate crowd dynamics. *ACM Transactions on Graphics (TOG)*, 31(6):190, 2012.

[16] B. Banerjee and L. Kraemer. Evaluation and comparison of multi-agent based crowd simulation systems. In *Agents for games and simulations II*, pages 53–66. Springer, 2011.

[17] Z. Jin and B. Bhanu. Optimizing crowd simulation based on real video data. In *ICIP*, pages 3186–3190, 2013.

[18] M. Moussaïd, D. Helbing, and G. Theraulaz. How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences*, 108(17):6884–6888, 2011.

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

[20] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In *Computer Graphics Forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

[21] P. Charalambous and Y. Chrysanthou. The pag crowd: A graph based approach for efficient data-driven crowd simulation. In *Computer Graphics Forum*, volume 33, pages 95–108. Wiley Online Library, 2014.

[22] P. Charalambous, I. Karamouzas, S.J. Guy, and Y. Chrysanthou. A data-driven framework for visual crowd analysis. In *Computer Graphics Forum*, volume 33, pages 41–50. Wiley Online Library, 2014.

[23] Jacques Jr, J.C. Silveira, A. Braun, J.Soldera, S. Musse, and C. Jung. Understanding people motion in video sequences using voronoi diagrams. *Pattern Analysis and Applications*, 10(4):321–332, 2007.

[24] E. Ronchi, E. Kuligowski, P. Reneke, R. Peacock, and D. Nilsson. The process of verification and validation of building fire evacuation models. *NIST Technical Note*, 2013.

[25] S. Singh, M. Kapadia, P. Faloutsos, and G. Reinman. Steerbench: a benchmark suite for evaluating steering behaviors. *Computer Animation and Virtual Worlds*, 20(5-6):533–548, 2009.

[26] M. Kapadia, S. Singh, B. Allen, G. Reinman, and P. Faloutsos. Steerbug: an interactive framework for specifying and detecting steering behaviors. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 209–216. ACM, 2009.

[27] M. Kapadia, M. Wang, S. Singh, G. Reinman, and P. Faloutsos. Scenario space: characterizing coverage, quality, and failure of steering algorithms. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 53–62. ACM, 2011.

[28] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3):507–525, 2001.

[29] PTV Group. Viswalk - http://vision-traffic.ptvgroup.com/en-us.

[30] T. Kretz, S. Hengst, and P. Vortisch. Pedestrian flow at bottlenecks-validation and calibration of vissim's social force model of pedestrian traffic and its empirical foundations. *arXiv preprint arXiv:0805.1788*, 2008.

[31] J. Henningsson and M.J. Blomstrand. Verification and validation of viswalk for building evacuation modelling. *LUTVDG/TVBB*, 2015.

[32] Legion. Legion - http://www.legion.com.

[33] AnyLogic Company. Anylogic - http://www.anylogic.com.

[34] Oasys. Massmotion - http://www.oasys-software.com.

[35] Arup. The verification and validation of massmotion for evacuation modelling. 2015.

[36] R. Löhner and E. Haug. On critical densities and velocities for pedestrians entering a crowd. *Transportation Research Procedia*, 2:394–399, 2014.

[37] D. Oberhagemann. Static and dynamic crowd densities at major public events. Technical report, Technical Report vfdb TB 13-01, German Fire Protection Association, 2012.

[38] D. Helbing, A. Johansson, and H.Z. Al-Abideen. Dynamics of crowd disasters: An empirical study. *Physical review E*, 75(4):046109, 2007.

[39] B. Steffen and A. Seyfried. Methods for measuring pedestrian density, flow, speed and direction with minimal scatter. *Physica A: Statistical mechanics and its applications*, 389(9):1902–1910, 2010.

[40] G. Voronoï. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les parallélloèdres primitifs. 1908.

[41] F.P. Preparata. The medial axis of a simple polygon. In *Mathematical Foundations of Computer Science 1977*, pages 443–450. Springer, 1977.

[42] H. Rohnert. Moving a disc between polygons. *Algorithmica*, 6(1-6):182–191, 1991.

[43] J-C. Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.

[44] R. Geraerts. Planning short paths with clearance using explicit corridors. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1997–2004. IEEE, 2010.

[45] N. Jaklin, W. van Toll, and R. Geraerts. Way to go-a framework for multi-level planning in games. In *Planning in Games Workshop*, volume 11.

[46] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344, 1987.

[47] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666–673. IEEE, 1999.

[48] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):743–761, 2012.

[49] B. Zhan, D. Monekosso, P. Remagnino, S.A. Velastin, and L-Q. Xu. Crowd analysis: a survey. *Machine Vision and Applications*, 19(5-6):345–357, 2008.

[50] S. Paris, J. Pettré, and S. Donikian. Pedestrian reactive navigation for crowd simulation: a predictive approach. In *Computer Graphics Forum*, volume 26, pages 665–674. Wiley Online Library, 2007.

[51] W. Daamen and S.P. Hoogendoorn. Qualitative results from pedestrian laboratory experiments. *Pedestrian and evacuation dynamics*, pages 121–132, 2003.

[52] A. Seyfried, O. Passon, B. Steffen, M. Boltes, T. Rupprecht, and W. Klingsch. New insights into pedestrian flow through bottlenecks. *Transportation Science*, 43(3):395–406, 2009.

[53] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 261–268. IEEE, 2009.

[54] B. Pushkarev, J.M. Zupan, B. Pushkarev, and J.M. Zupan. Capacity of walkways. *Transportation Research Record*, 538:1–15, 1975.

[55] W.H.K. Lam, J.Y.S. Lee, K.S. Chan, and P.K. Goh. A generalised function for modeling bi-directional flow effects on indoor walkways in hong kong. *Transportation Research Part A: Policy and Practice*, 37(9):789–810, 2003.

[56] U. Weidmann. *Transporttechnik der Fussgänger: transporttechnische Eigenschaften des Fussgängerverkehrs ; Literaturauswertung*. Schriftenreihe des IVT. IVT, 1993.

[57] A. Seyfried, M. Boltes, J. Kähler, W. Klingsch, A. Portz, T. Rupprecht, A. Schadschneider, B. Steffen, and A. Winkens. Enhanced empirical data for the fundamental diagram and the flow through bottlenecks. In *Pedestrian and Evacuation Dynamics 2008*, pages 145–156. Springer, 2010.

[58] A. Seyfried, B. Steffen, W. Klingsch, and M. Boltes. The fundamental diagram of pedestrian movement revisited. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(10):P10002, 2005.

[59] Matthias Plaue, Günter Bärwolff, and Hartmut Schwandt. On measuring pedestrian density and flow fields in dense as well as sparse crowds. In *Pedestrian and Evacuation Dynamics 2012*, pages 411–424. Springer, 2014.

[60] W. van Toll, N. Jaklin, Norman, R. Geraerts, et al. Towards believable crowds: A generic multi-level framework for agent navigation. 2015.

[61] J. Van Den Berg, S.J. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *Robotics research*, pages 3–19. Springer, 2011.

[62] Unity Technologies. Unity game engine, June 2005.

[63] Arthur van Goethem, Norman Jaklin, Atlas Cook IV, Roland Geraerts, et al. On streams and incentives: A synthesis of individual and collective crowd motion. *Technical report/Department of Computer Science*, 2015, 2015.

A Crowd Simulation Case Study

At the beginning of the study we performed crowd simulations for the start of the 2015 edition of the Tour de France, *Le Grand Départ* in Utrecht. The Tour de France is the largest annual sporting event in the world. The cycling race consists of 21 stages, attracting over 12 million visitors over the course of three weeks. The majority of the race occurs in France, with some stages visiting nearby countries. In the 2015 edition the first two stages started and finished in the Netherlands. The Tour started with a time trial through the city, starting and finishing in the area around Utrecht's central train station, see Figure 25. The area was the center of action, with many events occurring in and around the Jaarbeurs area. With the complexity of the resulting situation it was important to see how the crowd management could be optimized to reduce safety risks and improve crowd flow.

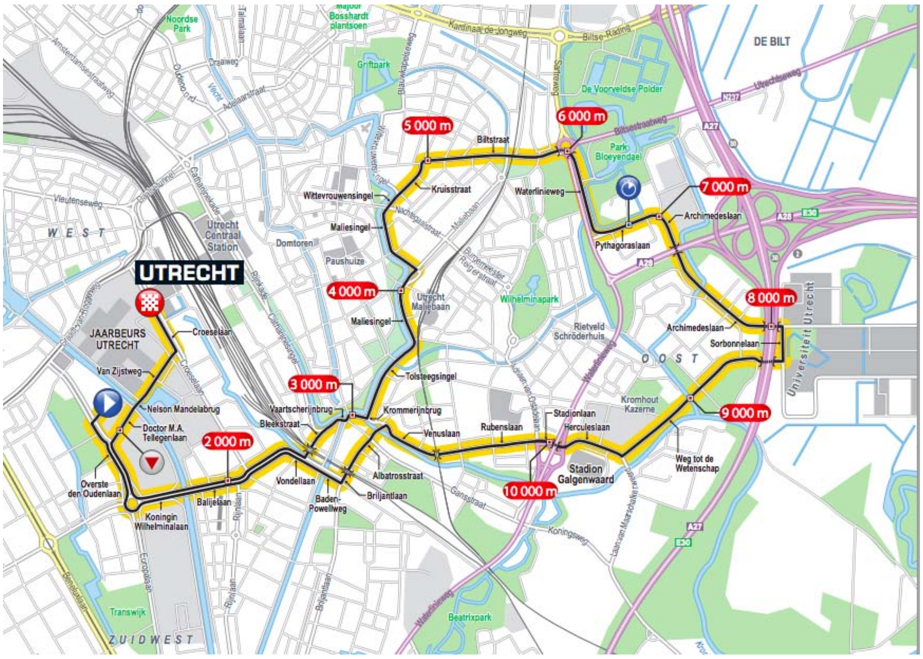


Figure 25: Overview of the first stage.

A.1 Goals and Assumptions

Complications

The crowd situation for the Grand Départ is complicated for several reasons. The event is spread over a large area, with the route being 13.8 km long. It is located in the fourth largest city of the Netherlands, easily accessible from anywhere in the country within a couple of hours. This means many people will take a last-minute decision on whether or not they visit. Since the event is freely accessible and a large part of the target audience does not need to arrange accommodation it is hard to estimate the number of expected visitors beforehand. Unpredictable factors such as bad weather can have a large impact on the number of visitors.

The crowd present at the event will be very diverse. People from all over the world visit the Tour. Both devoted cycling fans and first-time watchers will be present. It is hard to judge the individual behavior of these people. Some will do a lot of effort to find a good position on the first row, while others can be expected to take the first free spot they see.

It is also hard to estimate how people will travel to the event. Bus, train, tram or car are all viable options for people arriving from outside the city, while people already in the city will mostly arrive by bike or foot. Some part of the visitors will arrive days before the event, some part early on the day, and a final part right before the event starts. People arriving by public transport will arrive in bursts. Whenever a train or bus arrives the crowd flow will peak for a while. This means peak crowd density will be very different from the average density.

Assumptions
Due to the complicated structure of the entire environments we made several assumptions. Rather than simulating the complete scenario we focused on stress testing several key situations. This prevents a butterfly effect where the entire simulation ends up being inaccurate due to a single wrong assumption. For the selected environments we want to find the throughput it supports before a critically unsafe density occurs. We chose this critical value to be four people per square meter, in line with numbers found by [1].

We assumed people keep moving in the simulated areas. In practice groups of people tend to stand still on some places, but the areas chosen for simulation are the most dense in the environment, so it is reasonable to assume constant motion. During the event the organization had placed electronic signs at key locations that could be remotely controlled to show the route towards points of interest, such as the stage start or the central train station. One of the desired results of our simulations was to help decide the suggested route.

A.2 Environment

The area we were asked to simulate mostly concerned the ingress into the city. During egress, crowds were expected to reach the central station through alternative entrances. The main flow in the environment went from the public transport hubs towards the starting point of the stage. This was arranged to be a place of interest throughout the day, and was expected to draw many visitors. Spectators could walk a path around the team buses here and watch the cyclists prepare for their race. As this event was situated along the main route towards most of the course, a large part of the visitors was expected to follow this route.

We identified two potentially dangerous situations along this route:

Bottleneck
This situation concerns the flow between the public transport hubs and the Jaarbeurs halls, shown in Figure 26. The scenario focuses on a bottleneck created by a number of buildings on one side, and a fenced off area used by the organization on the other. This creates a funnel with a relatively small width of

11 meters. Given the expected number of visitors, as well as the bi-directional flow in this point we saw this as a potentially unsafe place. If indeed found to be dangerous this could be solved by either moving the fencing, allowing for a wider passage, or blocking entrance from the other side, creating a one-directional flow.

Canal Crossing
This environment concerns the flow from the Jaarbeurs halls to the starting area across the river. The bridge is around 9.5 meters wide, but could become a choke point for larger crowds. If found to be potentially dangerous compared to the expected number of visitors a temporary bridge could be placed. The organization wanted to use simulations to verify the effect of an extra bridge on expected throughput.

A.3 Implementation

The simulations were executed using a library implementation of the ECM Framework and visualized using Unity3D [62]. The ECM Framework takes a 2D environment representation and a set of characters as input, and returns the next position of each character.

The 2D representation of the environment is created using a navigation mesh of the walkable space. Using an overview of the environment as it was scheduled to be on the day of the event the walkable area was modeled. The representation used is binary; space is either walkable or not walkable. Preference for terrain types is not taken into account. This simplification was done to reduce the complexity of the environment.

The 3D representation was created with Unity. The buildings were modeled using a height map containing the height of each building in the area. Landmark buildings in the area, such as the *Beatrix Theatre* and the *Central Station* were manually modeled. To further increase recognizability the buildings are placed on a street map of the area. A screen capture of the result is shown in Figure 30.

The agents in the crowd use the Explicit Corridor Map Method [44] with side preference for path planning. For local collision avoidance the ORCA method [61] is used. Agents parameters are set to be normally distributed, see Table 5 for an overview of the settings used. Streams, as described in [63] were enabled.

A scenario consists of an environment represented by a 2D navigation mesh, a set of regions, and a set of connections between regions. A region consist of a bounding box and a parameter $AgentsPerSecond$ controlling the number of agents spawning from this region per second. A connection consists of a *Spawn* and a *Goal* region, and a weight. The weight determines the ratio of agents traveling towards each goal when multiple goal regions are defined for a single spawn region.

To test the capacity of the environments described in A.2 we measured density and flow for each simulation. For density we were mostly interested in the maximal density occurring in a simulation. As we were testing the capacity of specific situations, it was important to find the amount of people with which a critical density would first be reached. We measured the density as local density in a 1 meter radius around each agent. This means the minimal density for an

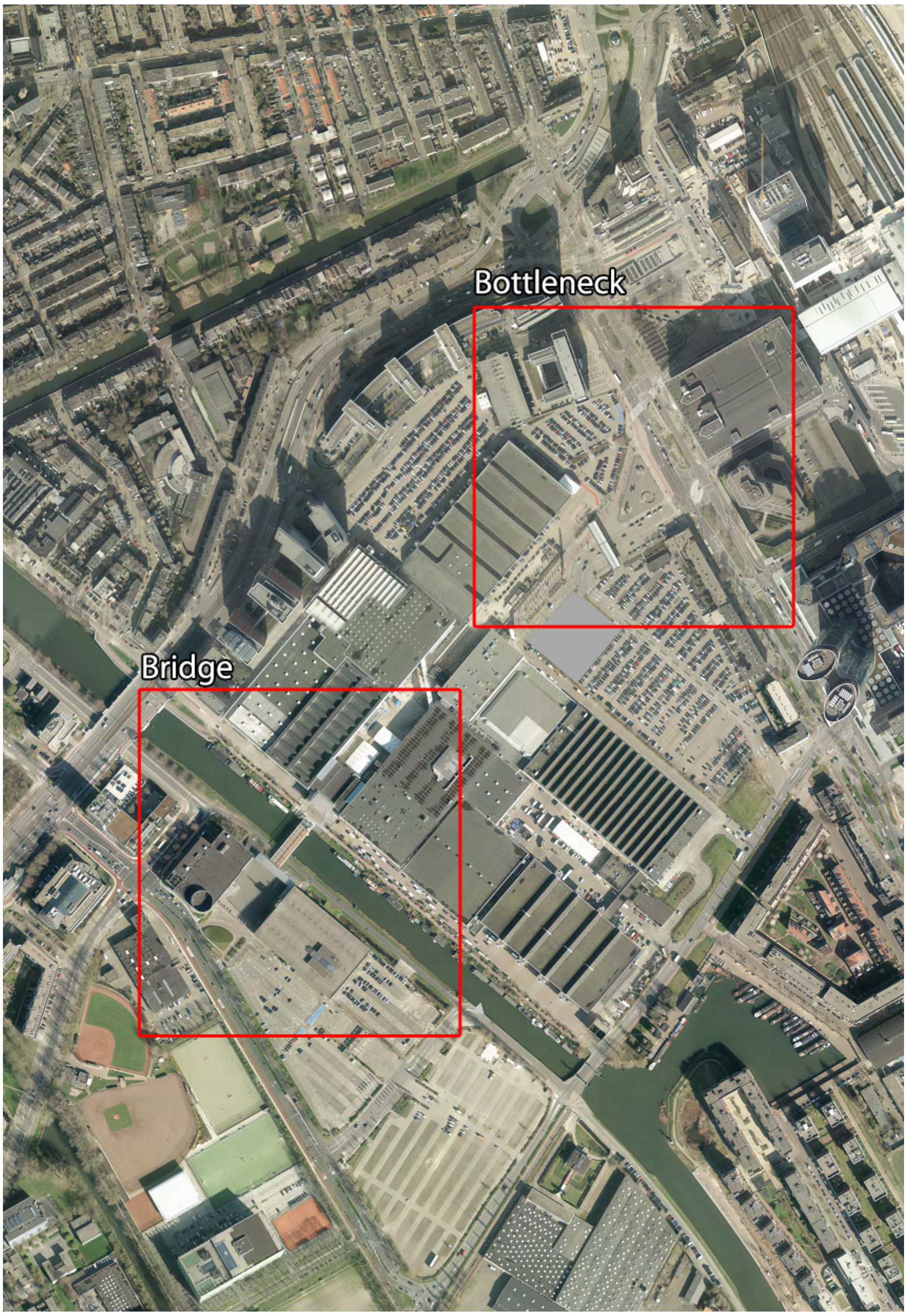


Figure 26: The simulated areas.

agent is always at least one, including itself. It is a simplified way of measuring density, but sufficient for our needs. For each time step in the simulation the maximal density value was saved, resulting in a graph that shows the evolution of maximal density over time. For flow we were interested in visualizing choke points. To do this we rasterized the environment and measured average speed

Parameter	μ	σ
Radius	0.24	0.024
Maximum speed	1.40	0.20
Side preference	0.30	0.30
Preferred clearance	0.20	0.05
Preferred personal space	2.00	0.20

Table 5: Parameter settings used for the agents.

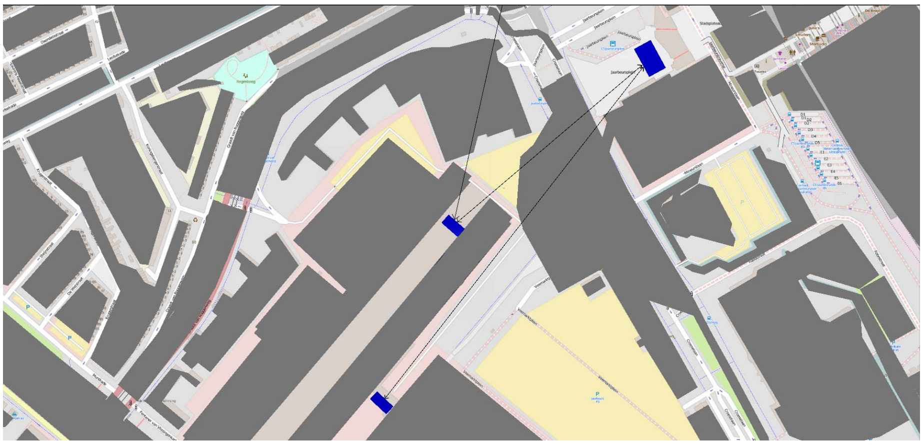


Figure 27: A screen capture from the scenario used for the *Bottleneck* simulation.

per cell across the simulation. This results in a heat map visualizing overall throughput for a simulation.

As a baseline for the simulations we used the numbers produced by the organization of the event: around 450 000 visitors on Saturday, of which a maximum of 250 000 arrived by public transport. As we were mostly concerned with the flows from public transport towards the starting area, we used 250 000 over the course of the day as a base line. A small number of visitors was expected to arrive as early as 7 am, but most were expected between 12 and 14 pm, when the race started. The organization expected as many as 25% to arrive in these two hours, meaning an average of 8.7 people arriving per second.

A.4 Results

As a measure for critically dangerous density we adhered to the numbers presented by Still [1] of four per square meter, for moving crowds.

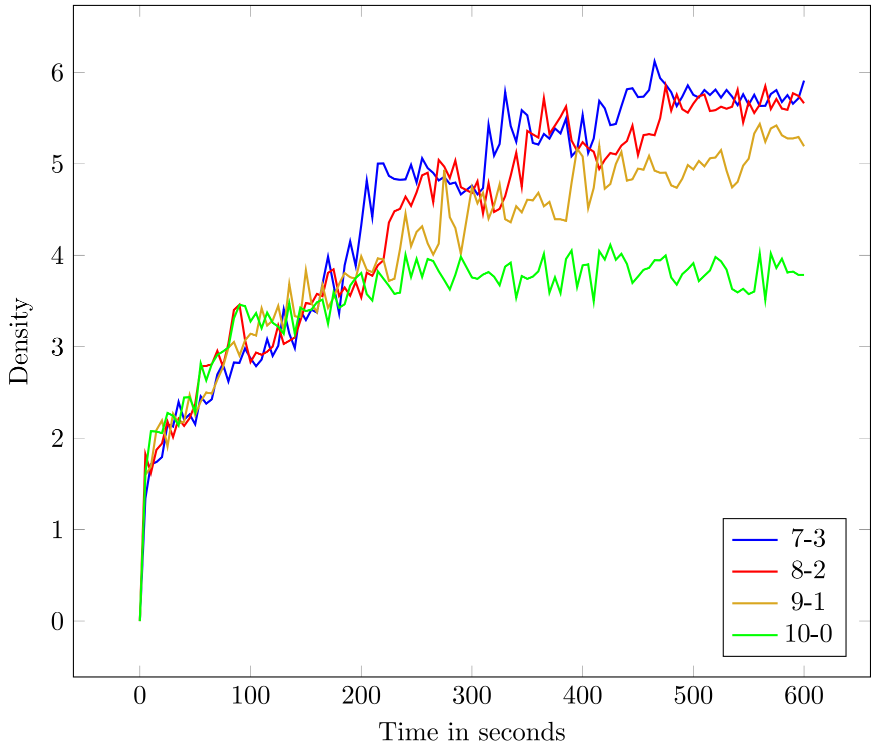


Figure 28: Density over time for the Jaarbeurs scenario with a total of 10 people per second. This graph shows the effect of increasing the counterflow from 0 to 3 people per second, while keeping the total number of people spawning the same.

Bottleneck

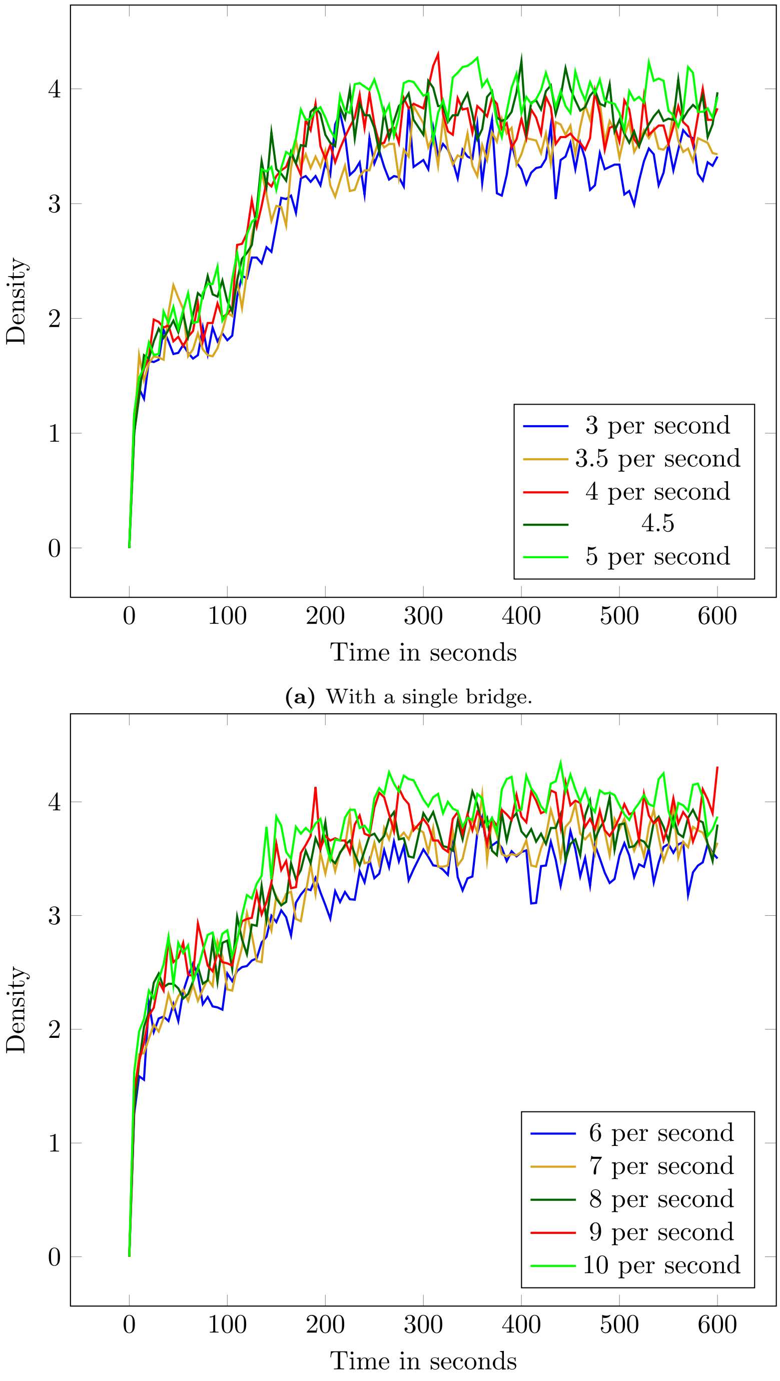
For the Jaarbeurs situation we compared unidirectional to bidirectional flow through the bottleneck. We tested maximal density in the crowd for a flow rate of 8, 9 and 10 people per second. For each of these rates we tested varying counter flow of 1, 2 or 3 people per second. We compared the result of each test to tests with the same total number of people per second. The results can be seen in Figures 28, 39a and 39b.

For each of the tests it is clear that adding any counterflow will result in a significantly higher density. When ten people per second pass through the bottleneck, density remains stable at just below $4/m^2$, as seen in Figure 28. When any counterflow is added the throughput through the bottleneck decreases and density increases up to 6, for a counterflow of 3 per second. In the simulation results it can be seen that the crowd eventually reaches a deadlock in the bottleneck, due to the lower throughput. This happened even in the scenario where 7 agents encounter a counterflow of one per second, as seen in Figure 39a.

While the numbers give no guarantees for the real situation, it is clear that the capacity of the bottleneck significantly decreases when a counterflow is added. Because of the width of the area, it is unlikely that problems will occur with just a unidirectional flow. This was useful information for the organization, since they could use signs to steer the crowd flow towards the central station away from this area. As an added safety measure, the fencing was also moved to allow for a wider passage.

Crossing
In this scenario we tested the effect of adding a second (temporary) bridge for crossing the canal. We tested the capacity of the first bridge, in people per second, before a critical density of four or higher has occurred. As seen in Figure 29a, density is relatively stable compared to size of the flow. Based on these tests the capacity of the single bridge seems to be around 4 people per second, before a density of 4 per square meter is reached. When a second bridge is added, this capacity more than doubles to 9 people per second.

Although the overall width of the crossing only increased by a factor of 1.8, the number of people supported before a density of $4/m^2$ is reached increased by a factor of 2.25. The organization of the event used this information when making a decision, and decided to use a temporary bridge.



(a) With a single bridge.

(b) With a double bridge.

Figure 29: Density over time for the Crossing scenario.

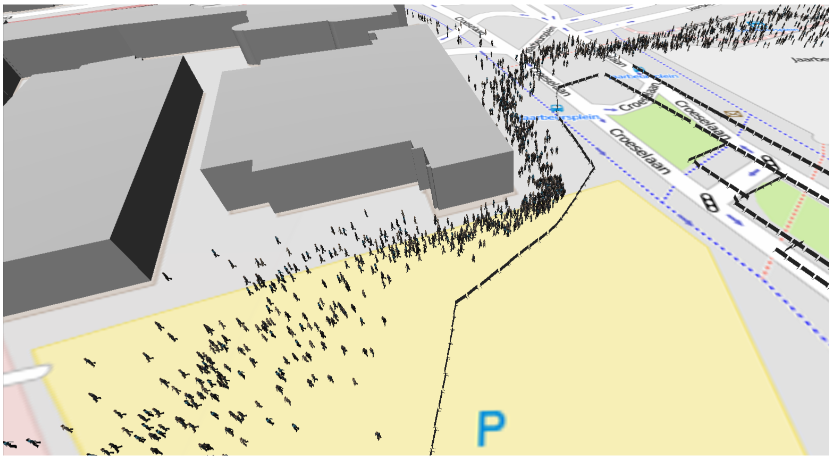


Figure 30: A screen capture of the crowd simulation result.

A.5 Comparison to Reality

The event occurred without major incidents. During the event the Jaarbeurs area was less crowded than expected. It turned out many people arranged accommodation within the city. Many stayed with family or slept in a hotel for a night. The heat wave prior to the event and the many warnings issued by the organization resulted in people either staying away or being well prepared. Because of the warm weather, many people planned an entire day around the event, rather than visiting purely the race. This meant the crowd arrived spread out over the day more so than expected. The expected peak right before the race start was spread out over the entire morning.

The Bottleneck situation was significantly less crowded than expected. Due to the relatively low capacity the organization advised people to take an alternative route to get to a spot along the route. It was still part of the route towards the events in the Jaarbeurs, but this area was mostly of interest before the start of the race. As a result of precautions taken by the organization and the crowd ingress being spread out over an entire morning, a peak close to what we simulated never occurred for this area.

Since the Bottleneck situation was less dense than expected, this resulted in less flow by the bridges as well. The area was still relatively crowded, as everyone who wanted to visit the starting area had to pass through, but it is hard to say whether the second bridge was actually required.

Right after the race officially ended, and the yellow jersey was awarded to the race leader, crowd flow into the train station became problematic. The detour into the train station suggested by the electronic signs was discarded by many people, as they knew the route they arrived by was significantly shorter. This resulted in a high amount of people trying to enter the train station, where they often had to wait for some time. The organization was prepared for this however, and utilized stewards to block the entrance and send people towards the intended route once the flow threatened to be problematic. Because of this the situation never became dangerous.

The simulations allowed the event organization to recognize potentially problematic situations in an early stage. The outcome of the simulations helped the organization motivate the extra costs involved in placing a temporary bridge, and were used as a guideline for the suggested walking routes. Although it is hard to say whether the precautions prevented a disaster, it definitely decreased the risk that dangerous situations developed.

B Density Experimental Results

B.1 Simulation Settings

Parameter	Value
Radius	0.24m
Max speed	1.4m/s
Preferred clearance	0.25m
Streams enabled	true
Collision avoidance	ORCA
Preferred personal space	0.5m
Side preference	0.0
Side preference deviation	0.9
time step	0.1s

Table 6: Simulation parameter settings used for the agents in the unidirectional hallway scenario.

Parameter	Value
Radius	0.24m
Max speed	1.4m/s
Preferred clearance	0.25m
Streams enabled	true
Collision avoidance	ORCA
Preferred personal space	0.5m
Side preference	0.5
Side preference deviation	0.5
time step	0.1s

Table 7: Simulation parameter settings used for the agents in the bidirectional hallway scenario.

B.2 Variance Tables

Method	Temporal	Spatial	Agent
Classic 1	$\mu = 1.999, \sigma = 0.872$	$\mu = 2.115, \sigma = 1.256$	$\mu = 1.641, \sigma = 0.730$
Classic 2	$\mu = 1.669, \sigma = 0.396$	$\mu = 1.789, \sigma = 0.498$	$\mu = 1.735, \sigma = 0.227$
Gaussian	$\mu = 1.472, \sigma = 0.293$	$\mu = 1.531, \sigma = 0.317$	$\mu = 1.536, \sigma = 0.150$
Voronoi 1	$\mu = 1.232, \sigma = 0.543$	$\mu = 1.398, \sigma = 0.449$	$\mu = 1.614, \sigma = 0.365$
Voronoi 2	$\mu = 1.532, \sigma = 0.668$	$\mu = 1.511, \sigma = 0.610$	$\mu = 1.689, \sigma = 0.479$

Table 8: Variance in the Hallway scenario

Method	Temporal	Spatial	Agent
Classic 1	$\mu = 1.201, \sigma = 0.815$	$\mu = 1.047, \sigma = 0.608$	$\mu = 1.941, \sigma = 0.446$
Classic 2	$\mu = 1.397, \sigma = 0.397$	$\mu = 1.322, \sigma = 0.406$	$\mu = 1.698, \sigma = 0.469$
Gaussian	$\mu = 1.257, \sigma = 0.325$	$\mu = 1.166, \sigma = 0.313$	$\mu = 1.544, \sigma = 0.344$
Voronoi 1	$\mu = 1.094, \sigma = 0.424$	$\mu = 0.879, \sigma = 0.296$	$\mu = 1.765, \sigma = 0.529$
Voronoi 2	$\mu = 1.147, \sigma = 0.343$	$\mu = 1.183, \sigma = 0.479$	$\mu = 1.847, \sigma = 0.692$

Table 9: Variance in the Bidirectional Hallway scenario

B.3 Global Density

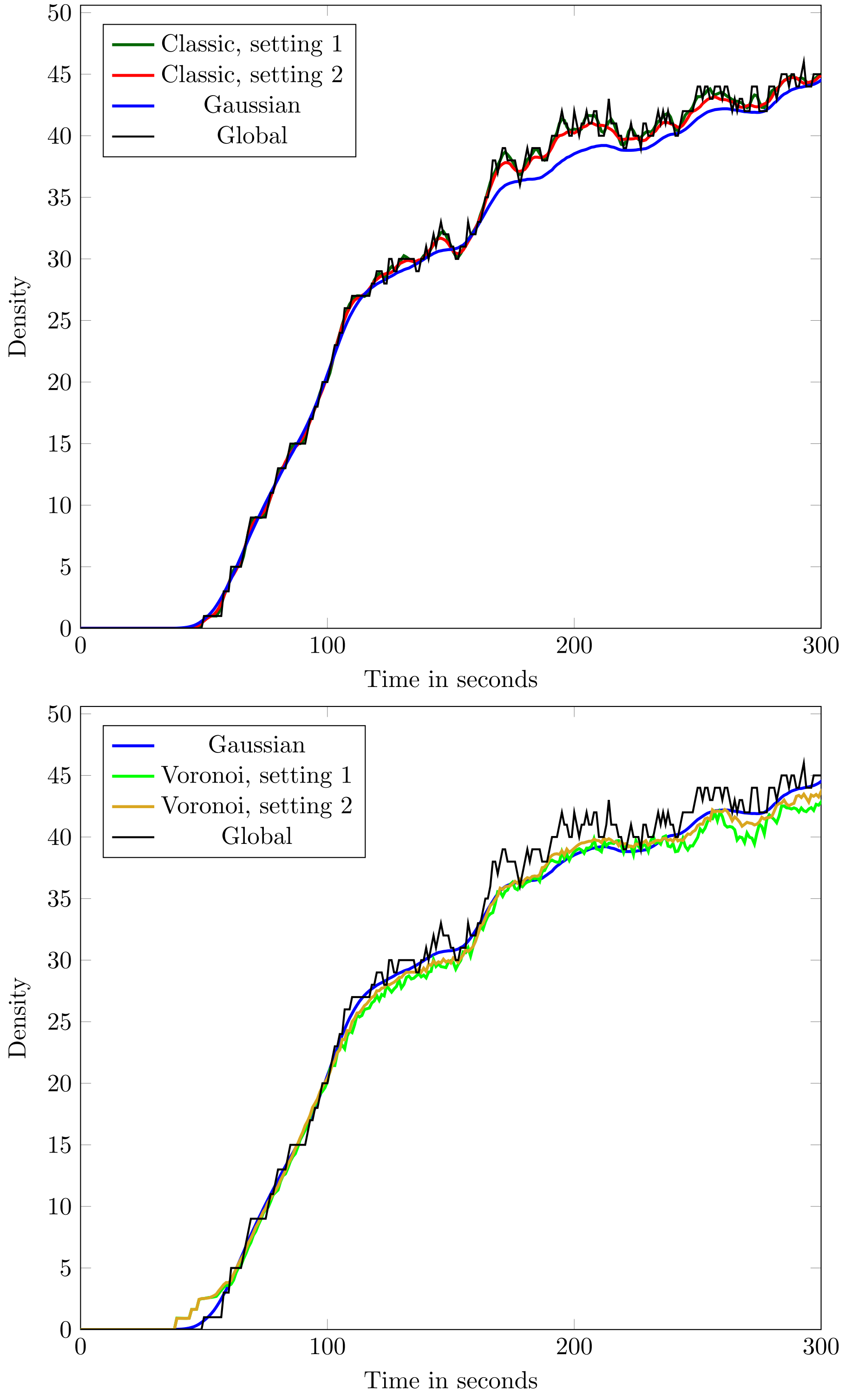


Figure 31: Density methods compared to global density in the bidirectional hallway scenario.

B.4 Variance over Time

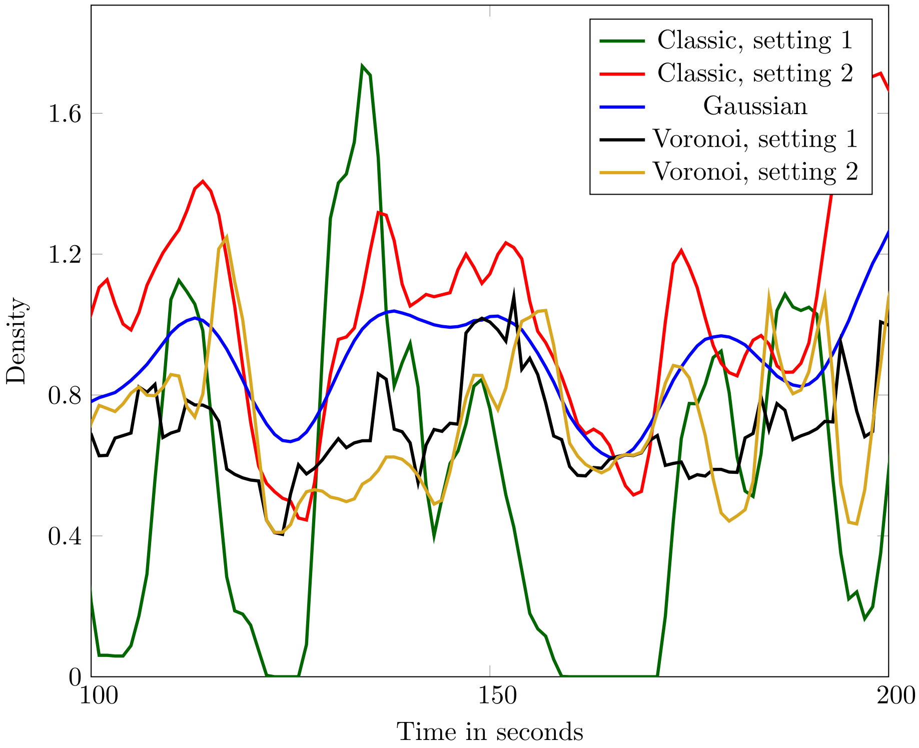


Figure 32: Variance over time for the Bidirectional Hallway scenario.

B.5 Variance over Space

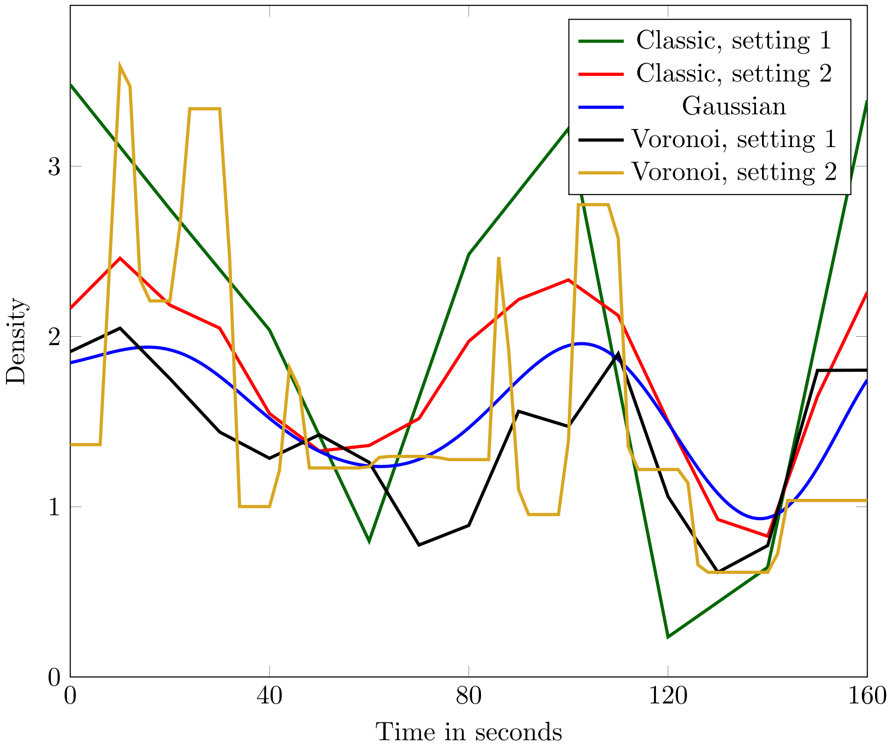


Figure 33: Variance over space for the hallway environment.

B.6 Agent Variance

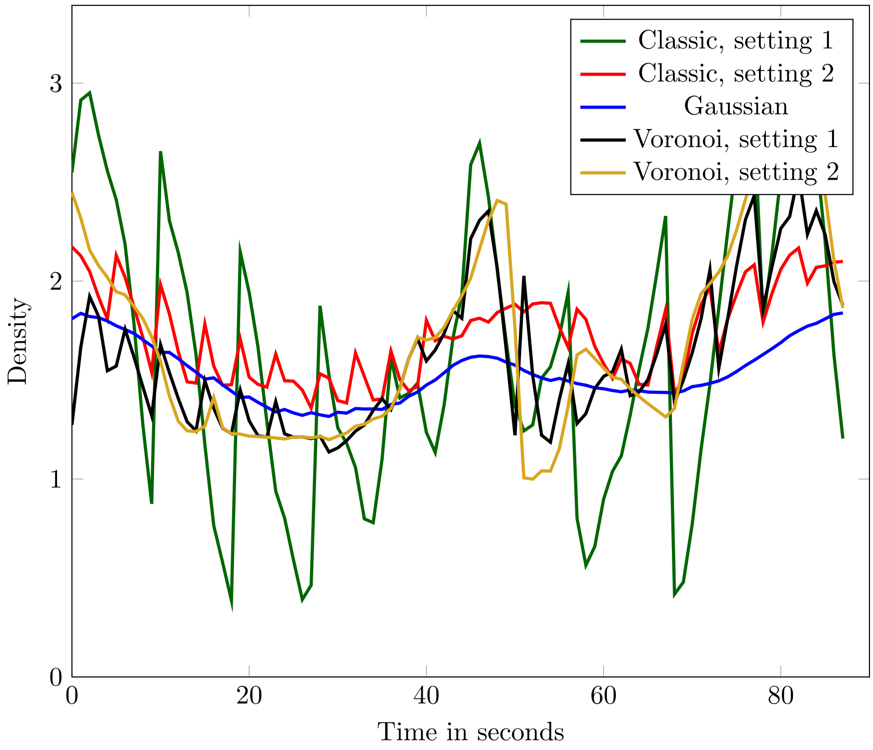
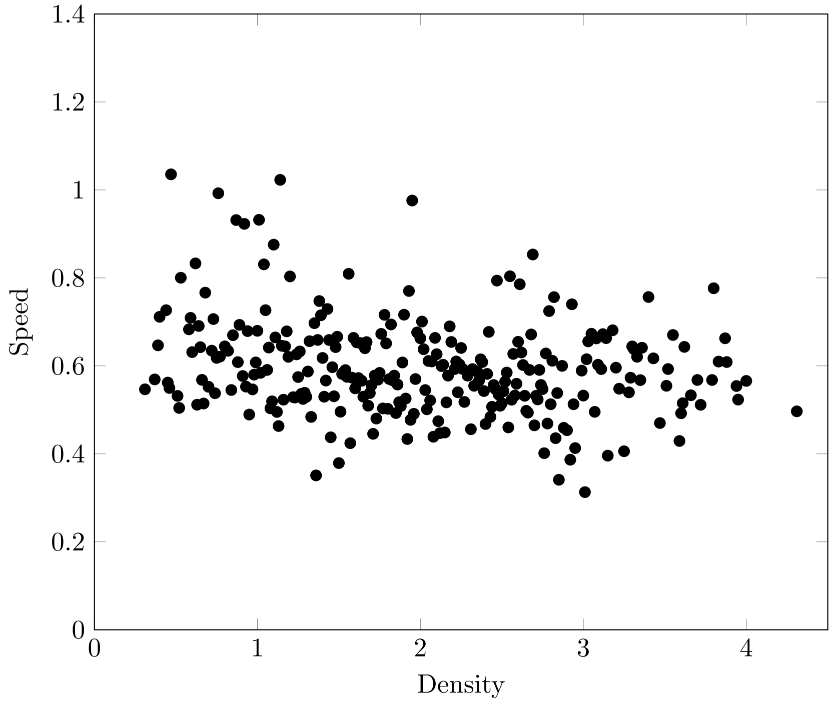
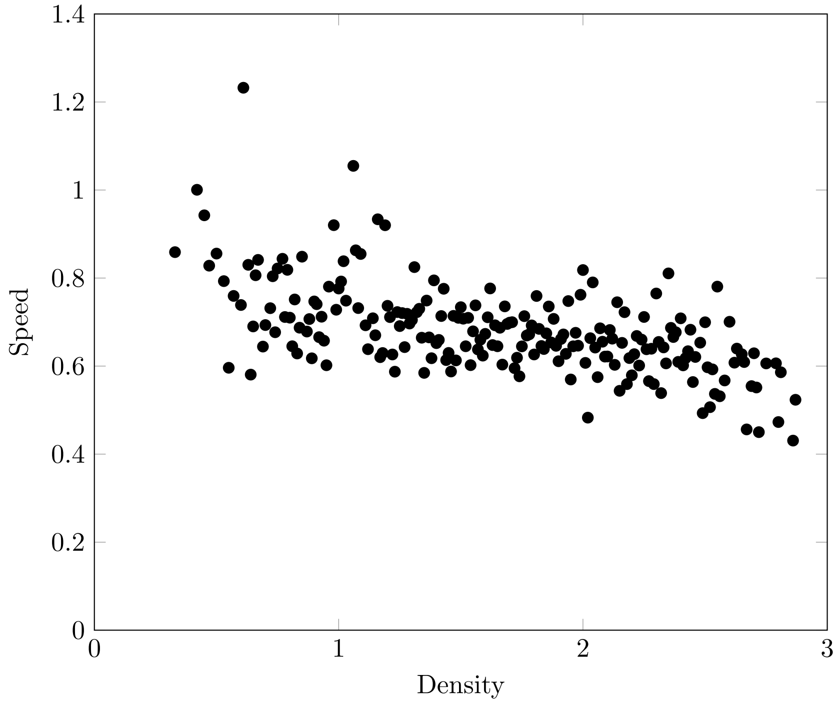


Figure 34: Agent variance in the bidirectional environment.

C Fundamental Diagram Graphs

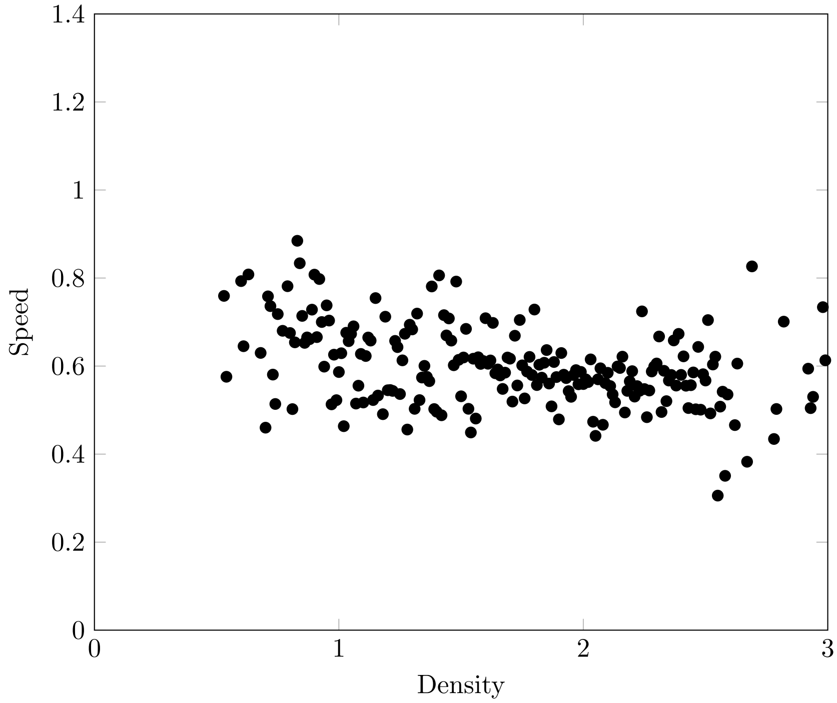


(a) Classic, setting 1

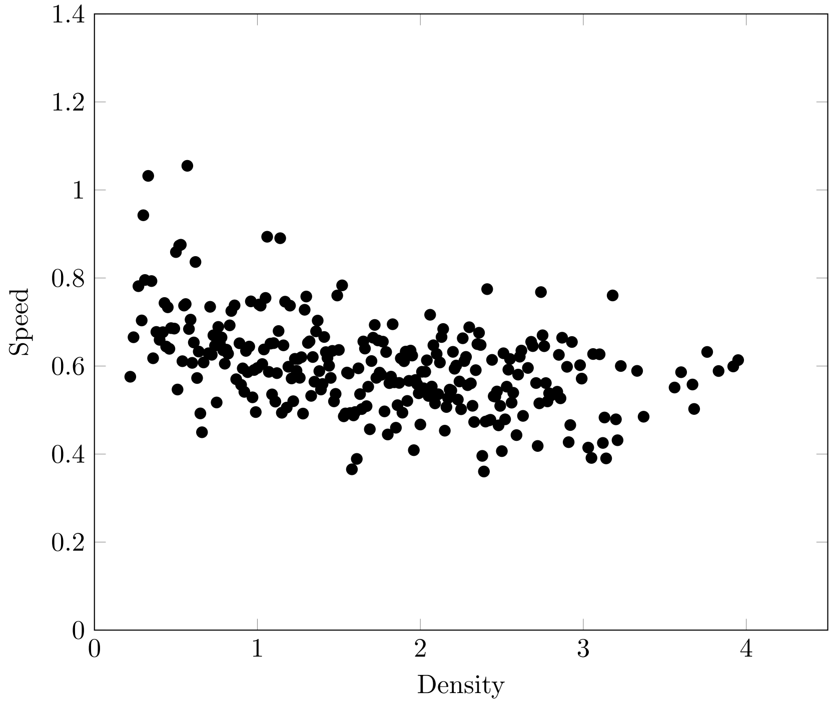


(b) Classic, setting 2

Figure 35: Fundamental diagrams created by different density measurement methods.

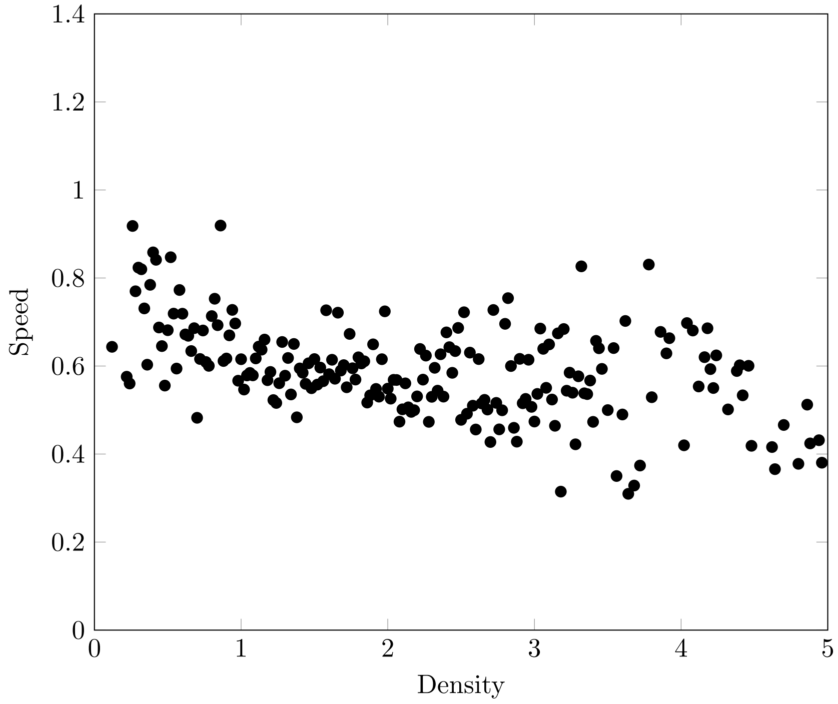


(a) Gaussian



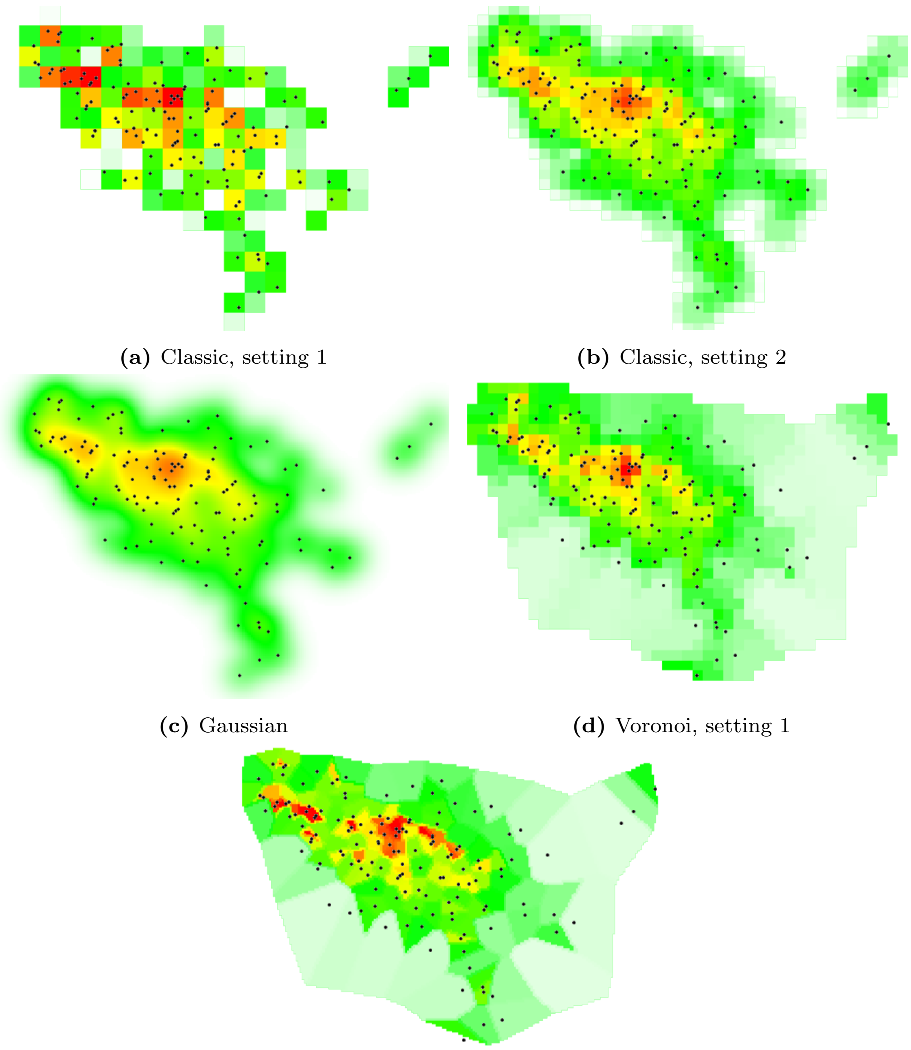
(b) Voronoi, setting 1

Figure 36: Fundamental diagrams created by different density measurement methods.



(a) Voronoi, setting 2

Figure 37: Fundamental diagrams created by different density measurement methods.



(a) Classic, setting 1

(b) Classic, setting 2

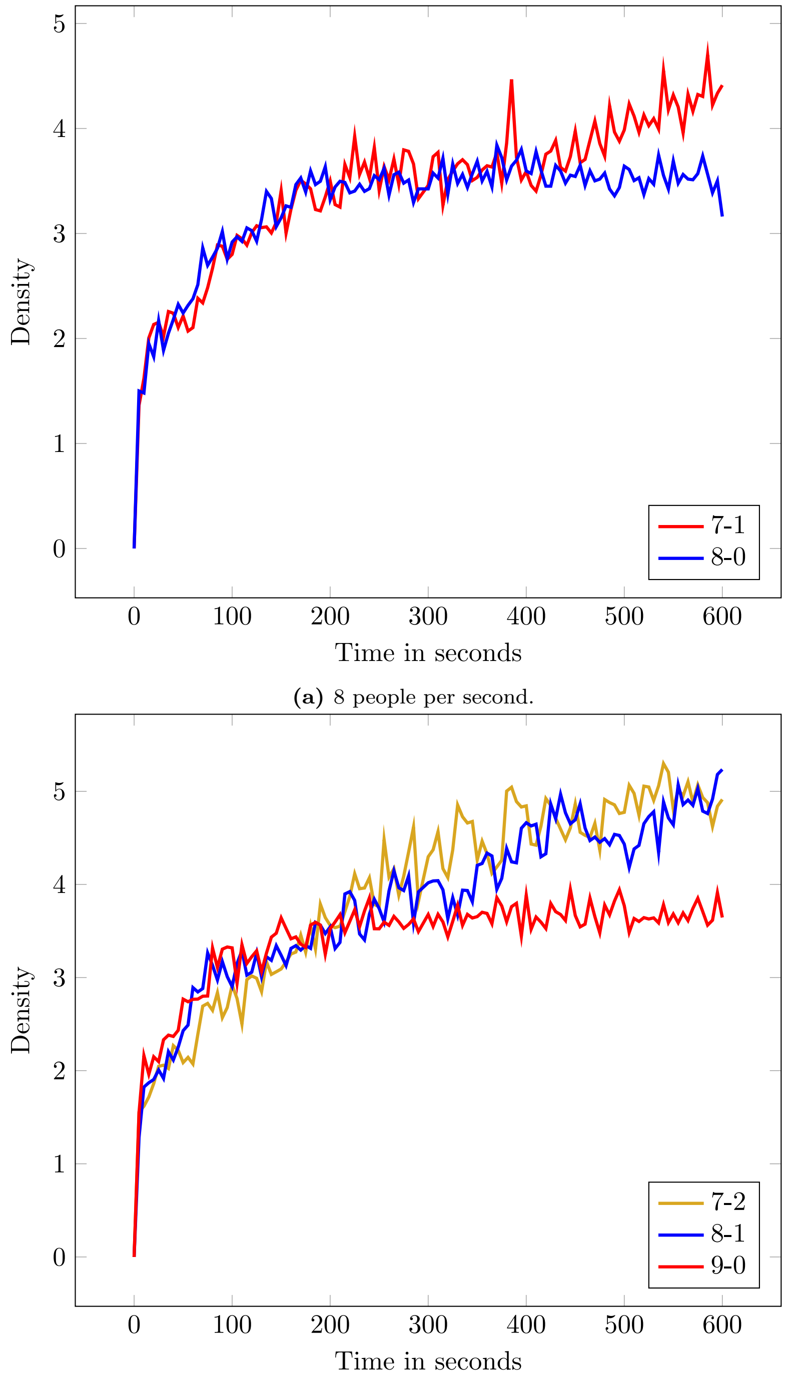
(c) Gaussian

(d) Voronoi, setting 1

(e) Voronoi, setting 2

Figure 38: Snapshots of the density measurements in the tracked crowd.

D Case Study Graphs



(a) 8 people per second.

(b) 9 people per second.

Figure 39: Density over time for the Jaarbeurs scenarios.