



Università degli Studi di Ferrara

Università degli Studi di Ferrara

Dipartimento di Fisica e Scienze della Terra

CORSO DI LAUREA IN FISICA

Statistical learning and simulating the paths of walking pedestrians

Relatori:

Prof. Alessandro Drago

Prof. Federico Toschi

Laureando:

Dario Chinelli

Correlatore:

Dr. Alessandro Corbetta

24 MARZO 2022

ANNO ACCADEMICO 2020 – 2021

Abstract

The dynamics of pedestrians change considerably depending on the surrounding space, not just for the intrinsic random movements that people make while walking, but also due to the reciprocal collisions and environmental conditions. We have considered a variety of scenarios in order to develop models and to create a tool that can give us simulations of the movements of a single pedestrian. To properly simulate pedestrian dynamics, we needed information about the probability of changing direction after every step and in every position of the trajectory. This approach is linked to the path in such a way that given a trajectory, it is possible to say with some probability where the next step will be. This mathematical approach may be computationally expensive, even more so considering the big amount of data we are using. Therefore, we started by implementing a discretized system and an easy model; then we moved to more complex ones. Finally, we got four types of models: two were time-dependent and two were time-independent. The scientific aim has been to create a mathematical framework, inspired by Lattice Boltzmann and Cellular Automata, with which we could learn, starting from real data, the dynamics of pedestrians and quantify them in terms of lattice transition matrices. The fundamental purpose is to succeed in quantifying the probability field found by utilizing different models. This field allows us to study the dynamics and to create simulations of pedestrians and trajectories whose statistics are indistinguishable from the real trajectories' statistics.

Contents

1	Introduction	2
1.1	Pedestrian dynamics background	2
1.2	Assimilating pedestrian dynamics	3
1.2.1	Cellular Automata Model	3
1.2.2	Data-Driven Model	3
1.3	Challenges	4
1.4	Recording technique of measure	4
1.4.1	Xovis 3D sensor	4
1.5	Theoretical tools	5
1.5.1	Markov Chain	5
1.5.2	The model's name and scheme	5
1.6	Presenting dataset	6
2	Propose data assimilation technique	9
2.1	Learning transition matrices from data	9
2.1.1	Model D2Q9	11
2.1.2	Model D2Q9Q9	13
2.1.3	Model TD2Q9	16
2.1.4	Model TD2Q9Q9	17
2.2	Simulated dynamics	17
2.2.1	Trajectories simulation	18
2.2.2	Distribution of probability	19
3	Results	22
3.1	Comparison of 3D histograms	22
3.1.1	3D comparison between real data and simulations	22
3.2	Simulated dynamics with probability models	25
3.2.1	Plots	25
4	Conclusion	29

Chapter 1

Introduction

1.1 Pedestrian dynamics background

Since the late eighteenth century, theories on human walking and pedestrian movement have been developed from many scientific perspectives [1]. Ongoing research has created widespread and diverse knowledge on this subject, branching research into many different specialisms of pedestrian research. Over the years, multiple literature review papers [2, 3] have become available which aid to create taxonomy in the available literature on pedestrian dynamics. In 1895, Gustave Le Bon stated in [4] that the conscious personality of the individual in a crowd is submerged and that the collective crowd mind dominates; crowd behaviour is unanimous, emotional, and intellectually weak. In the second half of the twentieth century, research was focused on social behaviour in crowded situations, by studying e.g. emergency evacuations and the relation to the corresponding domain layout. In the 1970s, analytical formulas for crowd phenomena were derived from empirical data. The following decade, a split in the research activities occurred: experimental work was joined by studies aided by technology (e.g., computer vision) and computational simulations for graphic applications. In this era, simulations evolved from providing basic numerical data outputs to complex three-dimensional virtual environments.

In recent years, technological and scientific advancements have enabled real-life high-accuracy measurements of pedestrian trajectory data. The usage of overhead depth-sensing cameras [5] allows for the anonymous, large-scale acquisition of pedestrian trajectories without compromising quality or privacy. Before the arrival of such data, only qualitative models of pedestrian behaviour were available, but these developments have enabled research on quantitative models. Large-scale trajectory data opens up new possibilities for research on statistical descriptions of pedestrian ensembles, but many other applications have already been published [6, 7, 8] as well.

In the current age, scientific works range from the understanding of single pedestrian behaviour to dynamic crowd interactions. The COVID-19 pandemic has proven that human movements are of extreme relevance for modern society as well [9]. Data collection methods mainly include real-life field observations, controlled experiments, survey-based methods and pedestrian simulation approaches. Recent work shows enormous potential for data collection methods, but restrictions are present as well. For example, field observations are limited by privacy-related issues and controlled experiments often fail to realistically represent real-life scenarios. Research on pedestrian dynamics is characterised by a very large heterogeneity in published works. This is caused by a large range in the branches of science that research pedestrian dynamics, including computer science, engineering, mathematics, physics, psychology and social science. Additionally, different works aim to study different phenomena, e.g., emergence of crowd self-organisation, vibrations in bridges caused by walking crowds or emergency evacuations.

In order to streamline further discussions on pedestrian dynamics, it is beneficial to introduce some definitions on topics presented in this thesis. Currently, there is some disagreement on literature definitions due to a great heterogeneity in published works. In 2019, the Consortium for the Physics and Psychology of Human Crowd Dynamics, constituted a glossary of terms related to crowd research [10]. Their work is not presented as an absolute truth on formal definitions but reflects current views and used interpretations of crowd-related terminology. These definitions will be used as a guideline in this thesis as well. In the glossary, a pedestrian is defined as a person moving on foot in a publicly accessible area. Further refinement of different pedestrian types is possible by including their motivation, such as pedestrian-commuter, pedestrian-shopper, or pedestrian-traveller. There is no clear agreement whether motionless persons should be still considered pedestrians, but nonmoving persons are still considered pedestrians in this thesis.

1.2 Assimilating pedestrian dynamics

The aim of this work is to clarify the possibility to analyze real life data and to generate simulated dynamics of the pedestrian crowd. To do so, it is necessary to collect data with significant precision and high acquisition ratio from a real-world environment. Other works focused on the acquisition of data from a very technical point of view [11], so this object is not taken into account in the work of this thesis. Then, once the data are collected properly, it is possible to *learn* from it. The dynamics of a pedestrian is complex, in which multiple conditions and forces play a role. The motion of a single pedestrian in a crowd is a similarly complex problem. Despite everywhere in the world, it is possible to find and watch walking pedestrian, it is not as simple to acquire data about their motion as it may seem. Therefore, the first issue is the data acquisition. One of the possible solutions to this problem it could be video recording at a given spot. However, this choice leads to more problems, such us privacy violation and object tracking from the video. The Xovis sensor [5] was used to collect the data showed in this work. This type of sensor is capable to solve both problems above. During the last decades, the development of machine learning and imaging recognition has provided more tools to analyze this type of data. This technological advancement has enabled real-life high-accuracy measurements of pedestrian trajectory directly in loco. The data are acquired through the usage of overhead depth-sensing cameras. This approach allows a large-scale anonymous acquisition of pedestrian trajectories without compromising quality or privacy. In this research a statistical approach is used to assimilate the average paths of pedestrians' trajectories. Based on this, four models are being studied to evaluate which one is capable of better predicting the most probable path. Due to the statistical approach, this is also a probabilistic model that could make or could not make possible to achieve a good prediction based on *probabilities*, which are derived by the real-data observation.

1.2.1 Cellular Automata Model

Cellular automata (CA) belong to the family of discretised modelling approaches. The model consists of a discrete spacetime lattice, along with computational capabilities that govern the evolution of the model through space and time. CA approaches often feature a finite amount of physical states per lattice site, but this is not a requirement. CA models are characterised by two main features: locality, ensuring that interactions can only take place between a given set of neighbouring cells, and modularity, which requires every lattice cell to be an independent process. The latter renders CA approaches very suitable for parallelised computing.

In the context of pedestrian dynamics, CA models discretise the pedestrian domain into a grid of cells, where every cell holds information and the presence and walking direction of pedestrians. Cells can also be flagged to be not accessible, to model boundary conditions in the form of objects and obstacles. The model should also have a set of transition rules, governing pedestrian movements between different cells. Such rules are often defined by probabilities and stochastic choice models, hence the close connection with the stochastic modelling category.

Cellular automata were first applied successfully in the context of pedestrian dynamics by Blue and Adler in 1998 [12], simulating one-dimensional pedestrian traffic, which was later extended to two-dimensional traffic flows [13]. CA models have also been applied successfully in the context of evacuation problems [14, 15] and in junction with other modelling categories.

Much of the criticism towards CA-based approaches follows from the method's discrete nature. Since the space-time lattices are often very symmetric, the lattices are considered to be too symmetric for realistic movements. Moreover, the finite number of states and rules per lattice cell cause non-natural homogeneous behaviour, as demonstrated by Bierlaire et al. [16]. Approaches to overcome these limitations have been proposed by Lubas et al. [17], in which the authors created a non-homogeneous and asynchronous CA model with cell-dependent transition rules. Still, the CA model remains a popular platform for studying pedestrian dynamics following its computational simplicity.

1.2.2 Data-Driven Model

The data-driven category distinguishes itself by a strong dependence on real-life measured pedestrian behaviour. In the literature review, two different approaches are concerned, namely data-in-the-loop approaches and data-in-the-model approaches. In the data-in-the-loop models, real pedestrian data (consisting of group behaviour or individual trajectories) are assembled into a collection, which is then used to perform simulations. In Lerner et al. (2007), pedestrian trajectories are captured from video recordings, which are used to generate natural pedestrian behaviour in a virtual environment [18]. In the work by Porzycki (2014), a pedestrian simulation is coupled with a measurement

setup, asdetected pedestrians are initialized as embodied agents in the simulation [19]. In 2010, Ju et al. introduced a crowd generation approach, in which crowd formations and individual trajectories were taken from video recordings [20]. These measurements were then used to create virtual interpolated crowds of different densities. All data-in-the-loop approaches suffer from interpolation artefacts causing non-realistic behaviour, especially in the limit of high densities.

Data-in-the-model are similar to the aforementioned methods, but have one key difference in their workings: the parameters of an existing simulation model are adjusted based on real pedestrian measurements. This category has much more works reported in the literature review, for all modelling categories considered, such as mechanical, cellular automata and stochastic models [21, 22, 23]. The most travails are encountered in the area of data extraction: it is time-consuming work to capture high-quality pedestrian measurements, moreover for large crowds.

1.3 Challenges

This section's scope is to explicitate the research question in a •ry simple and synthetic way.

Which is a *good* data-driven mathematical framework that better represents the original data? In this type of system there is a multitude of factors that determinate the path of a single pedestrian. Thus, let's consider a single pedestrian P who walks in a certain space. The first type of interaction it is the structure where P can or cannot walk through; this structure is defined as the whole domain Ω . The second interaction is between P and the other pedestrians. Every pedestrian needs a personal space all around; due to the circumstance, there is a variable, and it is not easy to analytically determinate it. The third interaction corresponds to random events along the P 's path; for random events we mean real world events. But this is not the end of the list. This work scopes is to learn from real datas without necessarily define and separate those factors above; but define a mathematical framework (MF). The MF has to be able to determinate if a given trajectory is a common one or not. In other words, the MF collect the information about the more probable paths based on what it was used to generate it. This is called a *data-driven* model.

Which is a good representation for a random combination of trajectories? It is possible to plot every single trajectory, but, increasing the number, this leads to an unreadable representation, which then results to be not functional. It is also possible to plot easily the heatmap of a dataset to analyze the most “walked” areas. Although this second plot choice can consider a great number of trajectories and still be readable, it has a problem: it leads to a representation where the time dependency is completely lost. To be able to represent datas that have statistical relevance, it is introduced a 3 dimensional plot that that shows positions in time with the same number of occurrence.

How to built a proper software library? A crucial part of this work is the writing of the software capable of doing all the necessary steps, from the raw data to the results and the simulated dynamics. The idea is to have a Python module that permits to work on the data from a raw structure to the results and the simulated dynamics. This software has to be written from scratch, but, given the nature of the Python language, is pretty easy to embed other features. The name of this library is *“pathintegralanalytics”* (PIA). The PIA takes a raw CSV file, that contains the information about the position in time of the studied trajectories, and returns an object. On this object it's possible to work with the PIA module and analyze the trajectories, get the velocities and collect the information to generate the mathematical framework.

1.4 Recording technique of measure

The recording technique selected to make this work possible, it is the Xovis sensor. The entire field of view is covered using multiple cameras working together. Starting from the raw images, each object is tracked down along its entire path. This is possible by using imaging recognition software. The software gives as an output a data collection with coordinates and time for each pedestrian.

1.4.1 Xovis 3D sensor

The sensor is designed by *Xovis company*, and it is composed of two cameras able to generate a stereo view. The Xovis 3D sensors master every people counting and people flow measurement challenge with high precision. This technology enables people counting and tracking in real-time. A minimal

design and embedded processing are some of the main characteristics which build the base of the Xovis sensors. Further, AI-based algorithms improve the accuracy and flexibility of people counting and people flow management. The signature 3D stereo vision technology permits accurate people counting of up to 99.9%. Over a decade, these sensors have been field-tested and proven to stay true to Swiss precision. The Xovis 3D stereo vision sensor with a powerful on-sensor person tracking engine always guarantees data privacy. Data is only transmitted in text format and without any kind of personally identifiable information. The sensors can be configured to be GDPR compliant. Sensors can work together as one, easily covering large areas and tracking visitor paths.



Figure 1.1: The Xovis 3D sensor.

1.5 Theoretical tools

This section's scope is to introduce some nomenclature and theoretic information for the following chapters.

1.5.1 Markov Chain

A *Markov Chain* (MC) is a stochastic model. It predicts the future outcoming state based on the present one. In other words, the present state determinates the probabilities for every possible future outcome. The MC may be represented as a diagram (Figure 1.2a) [24] in which the arrows are the possible transitions. A number $p \in (0, 1)$ may also be indicated on the arrows to specifies the probability of that transition. Another model's representation is a *stochastic matrix*, also called P matrix. The matrix entries P_{ij} have as row-index i the starting state and as column-index j the ending state of the system. Hence, all the entries are referred to a specific transition. A two-state Markov Chain is the most basic model, which can be used for the illustration of the Markov process. The diagram in (Figure 1.2) represents the possibility that the system must change from both states. For instance, from the state W the system can move to the state B with the big black arrow or can remain in the state W with the small white arrow. The entries in the Markov Matrix in (Figure 1.2b) are positives numbers from 0 to 1; they represent the probability of changing state. The sum on the outgoing arrows must be equal to 1.



Figure 1.2: From the Markov diagram to the Markov matrix of a tow-state system.

1.5.2 The model's name and scheme

The aim of this section is to give the notation used about the models and the mathematical tools.

Model's name (FAI DISEGNI) The D2Q9 is the abbreviation for the "two dimensional space" (D2) and considering the "nine next near cells" (Q9). The name used here of the first considered model is "*D2Q9-model*" and it's also the simplest. The second use the same components of the previous but including also the "nine previous near cells" so that a second (Q9) is added to the end; its name is "*D2Q9Q9-model*". The third model is again similar to the very first one and it also considers "time" (T); its reference name is than "*TD2Q9-model*". The fourth model, and last in this work, is named as "*TD2Q9Q9-model*", so that this model is developed in a "two dimensional space" considering the "nine next near cells" and the "nine previous near cells".

The developed mathematical framework The mathematical framework MF utilized in this work is primarily composed by a tensor of probabilities; that immediately follows from the MC theory. The reference name to this tensor in the following pages is A , where are added indexes, as subscripts.

For the simplest model used here, the *D2Q9-model*, it's a three dimensional tensor named A_{xyk} . For the second, the *D2Q9Q9-model*, it's a four dimensional tensor named A_{xykh} . For the third, the *TD2Q9-model*, it's a four dimensional tensor named A_{txyk} . For the fourth, the *TD2Q9Q9-model*, it's a five dimensional tensor named A_{txykh} .

1.6 Presenting dataset

Utrecht Centraal (Floorefield 10) The data analyzed in this work are given as a CSV file and come from a collaboration with the *ProRail company*. Specifically this data were collected at the Utrecht's train station during one day. The (Figure 1.3) shows the camera's point of view of the analyzed field. This spot offers a great multitude of path's type, due to its *morphology*. It is a rectangular base field, where there are a few obstacle and a lot of entrance and exits. This field is both a corridor from two zones of the station and a cross zone. It also as more than one shops where people may entry to or exit from it. This complex scenario permit to strongly compare all the four models to each others and with the real datas.



Figure 1.3: Utrecht Centraal, cameras point of view (Floorfield 10)

Sample from the dataset In this section are showed a few example of trajectories selected form the dataset. Each trajectory is presented with three types of plot: the line of the path (L), the heatmap generated from that path (H) and the velocities distribution along x and y (V). A great number of collected path are given by pedestrians that have walked from left to right or from right to left, along a *semi-straight* horizontal line.

As first example: the (Figure 1.4 - 1.6) is the path (1) of a random pedestrian that has walked almost straight all the way. The interpretation of the (V) plot in (Figure 1.6) permit to know the direction of that pedestrian. For (1) the velocity distribution is focalized in the negative velocity along x and close to zero along y . That means: the pedestrian goes from right to left along x . The next path (2), in (Figure 1.7 - 1.9) is pretty similar to the (1), but instead it proceed from left to right with positive velocity.

Other paths are like the one in (Figure 1.10 - 1.12), path (3), and (Figure 1.13 - 1.15), path (4). Both path (3) and (4) go straight almost all the length of the field, then (3) turn down and (4) turn up to go exit the field.

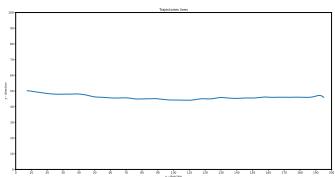


Figure 1.4: Plot L, path 1

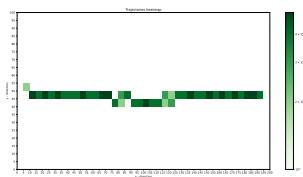


Figure 1.5: Plot H, path 1

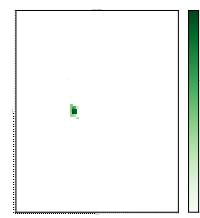


Figure 1.6: Plot V,
path 1

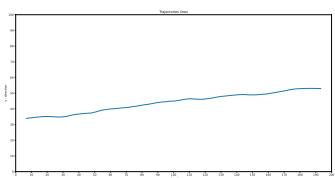


Figure 1.7: Plot L, path 2

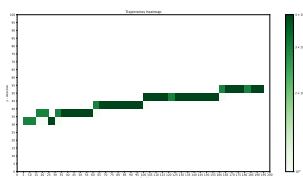


Figure 1.8: Plot H, path 2

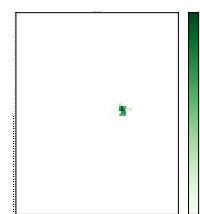


Figure 1.9: Plot V,
path 2

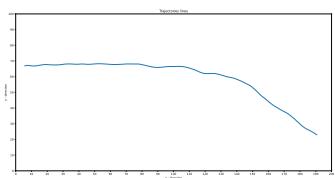


Figure 1.10: Plot L, path 3

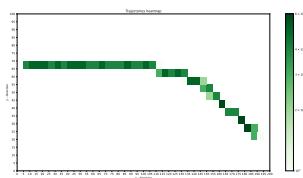


Figure 1.11: Plot H, path 3

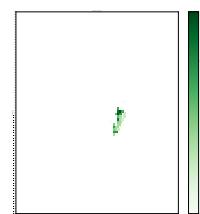


Figure 1.12: Plot V,
path 3

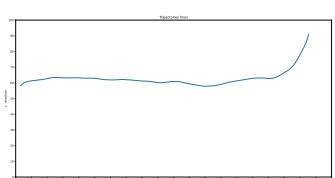


Figure 1.13: Plot L, path 4

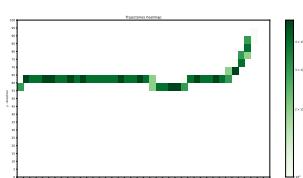


Figure 1.14: Plot H, path 4

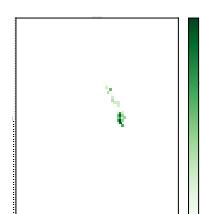


Figure 1.15: Plot V,
path 4

Other paths are stranger than the previous. Some trajectories are showed in (Figure 1.16 - 1.21).

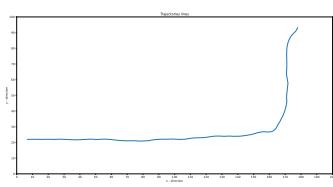


Figure 1.16: Plot L

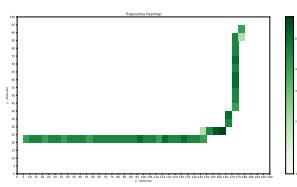


Figure 1.17: Plot H

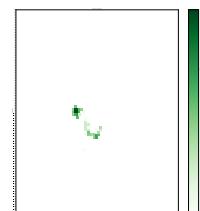


Figure 1.18: Plot V

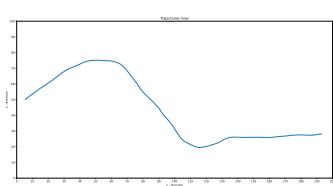


Figure 1.19: Plot L

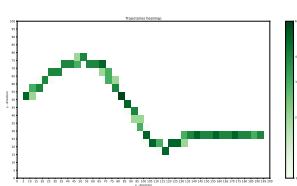


Figure 1.20: Plot H

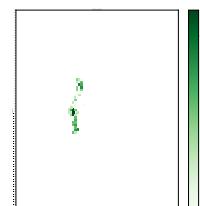


Figure 1.21: Plot V

Chapter 2

Propose data assimilation technique

2.1 Learning transition matrices from data

In this study, a total of four models are considered. Two of them are dependent by the position in space, also called *time-independents*. The others two are dependent by the position and time, also called *time-dependents*. However, there is also a distinction between the D2Q9s and the D2Q9Q9s. For the D2Q9s what it is doing is considering the velocity from a cell to another; therefore, just the change in position. For the D2Q9Q9s, it is also considering the acceleration; hence, the change in velocity. The starting point of each one is the dataset, collected from a real-life situation.

Since each of them are entirely based on real world pedestrian's path in a crowd, those models considers the imposed limit due to the presence of other pedestrians; an example of this limit could be the tendency of other pedestrians to not collide with each other and it also considers the boundary condition given by the structural environment. The strong point of this model is that it is generated by the real-world observation and not build by hand. With the aim of reproducing realistic pedestrians' movements, synthetic paths are created from the models. Every model generates one trajectory that simulate just one pedestrian in a statistical crowd. When simulating more paths, it considers pedestrian who walks alone in the crowd. This model does not consider the interaction made by the others simulated pedestrians.

6	2	5
3	0	1
7	4	8

Figure 2.1: This figure shows the indexes associated to each possible transitions from the center to the another cells. To every transition is associated a number: the index k .

(-1, +1)	(+0, +1)	(+1, +1)
(+1, +0)	(+0, +0)	(+1, +0)
(-1, -1)	(+0, -1)	(+1, -1)

Figure 2.2: Given the initial position at the center square, this is a representation of the change in coordinates to the next cell. The notation represents the variation along x and y axes, as $\Delta x, \Delta y$, from the initial position (x_0, y_0) .

Notation Lets call the observation field Ω_c , defined as a continuous space where pedestrians are tracked. Lets assume $\gamma = \gamma(\vec{x}_c, t)$ a pedestrian's path, where $\vec{x}_c = (x_c, y_c)$ has a bi-dimensional spacial and time dependancy.

The path γ in that space has a start position A and an end position B . Then, the field Ω_c is divided into *rectangular* cells, dividing the real space along x with a maximum extension indicated as L_x in a certain number of cells D_x ; the division happens as well for the y -direction, with obvious notation: L_y and D_y . After this discretization, from Ω_c is obtained a *grid space* called Ω_g . In this grid

space, every path γ is converted from continuous $\gamma = \gamma(x_c, y_c, t)$ to discrete coordinate $\gamma = \gamma(x_g, y_g, t)$ referred to the *grid*. To lighten up the notation, while speaking of *grid space*, it is simply used (x, y) in reference to the discrete grid position.

The standard D2Q9 configuration Referring to (Figure 2.1), the *map* is set for each position (x_0, y_0) in the grid space, and it represents the eight neighbors and the central position where a pedestrian could go. Each direction will be associated to a certain transition probability.

When a trajectory change position, in the grid space (Figure 2.2), from $P_0 = (x_0, y_0)$ to $P_1 = (x_1, y_1)$ is associated a transition. The transition is identified by a number $k = 0, 1, \dots, 8$ such that is unique. It is derived from the series of coordinates for each trajectory and each step in time. When the calculation is made for each step, a transition number is also associated for every position in time; this number represents where the position is going to go in the next step. If this transition is associated to the change in position, it identifies a certain velocity, such as a vector with a certain direction. Iterating this procedure to the entire pedestrian's trajectory, it is possible to get something similar to what is illustrated in (Figure 2.3). In that figure, it is possible to distinguish the path in the continuous space and the discrete path in the grid space. It also shows the direction of the next movement for each position, using arrows that are consistent with the velocity arrows in each position. The numbers are the value of the k -index in each position, and they are consistent with the maps above.

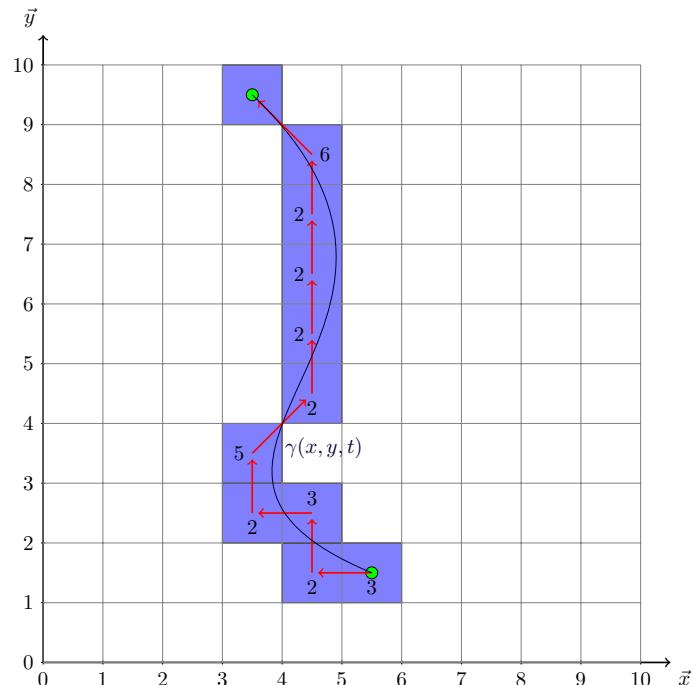


Figure 2.3: This figure represents a trajectory in the *continuous space* as the blue line γ . That path γ is discretized in the *grid space*, represented by the blue cells. The red arrows represents the change from a cell to the next. The numbers are the associated to the D2Q9 indexes to those moves, also called *k-directions* and showed in (Figure 2.1). The coordinate's changes are explicit in (Figure 2.2).

Probability distribution The tool used in this work is a *move probability* tensor. For each position, and eventually also time, it returns a number between 0 and 1 for each element. The sum over every directions must be 1, because of the normalization. This tensor is multidimensional, as described in the previous paragraphs, and its dimension depends on the model. With this tool is possible to plot the map with the corresponding probability for each of the nine directions. It is possible to see along the trajectories where is the more probable direction to take and which is the less. To describe this lets take into account just a few real trajectories, with a common path and opposite directions. To do so, here are considered five pedestrians in (Figure 2.5), with two representations: one is plotting the actual lines in the field (Figure 2.4) and the other is a heat-map that describes where pedestrians passed through (Figure 2.4).

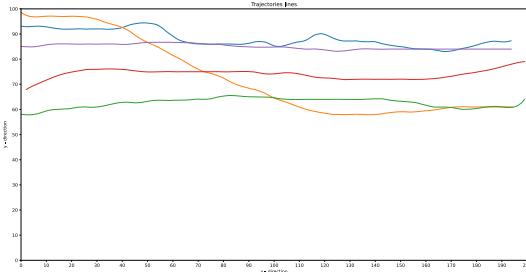


Figure 2.4: Trajectories lines of five “real” pedestrians.

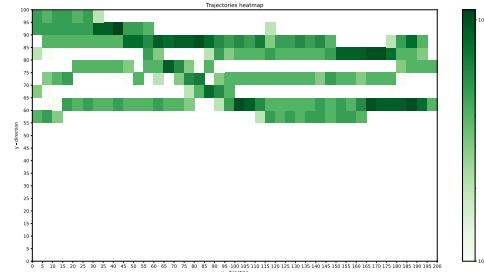


Figure 2.5: Representation of five real trajectories from dataset.

2.1.1 Model D2Q9

The simplest model considered here is called *D2Q9-model*. This model is time-independent, and it considers the velocity of the pedestrians. Given a starting position (x_0, y_0) in the field Ω , it uses the nine closest possible positions a pedestrian could go to from that point. The model is generated by recursively adding every trajectory’s probabilities to the mathematical framework. This MF is a tensor which collects the probabilities information, about all the trajectories, as its elements. Through the D2Q9-model, it is possible to know, for each position (x, y) , the probability to go up, down, left, right or a combination of those movements.

Transitions From the initial position $P_0 = (x_0, y_0)$ to the next closest cell in the grid, P_k is defined by the index k ; in this way, the index k gives the direction of the transition. To explicit all the transitions from P_0 to P_k , those transformations are defined in (Equation 2.1) and represented as diagram (Figure 2.6):

$$\begin{aligned}
 P_0 \rightarrow P_0 : & (x_0, y_0) \rightarrow (x_0, y_0) \\
 P_0 \rightarrow P_1 : & (x_0, y_0) \rightarrow (x_0 + 1, y_0) \\
 P_0 \rightarrow P_2 : & (x_0, y_0) \rightarrow (x_0, y_0 + 1) \\
 P_0 \rightarrow P_3 : & (x_0, y_0) \rightarrow (x_0 - 1, y_0) \\
 P_0 \rightarrow P_4 : & (x_0, y_0) \rightarrow (x_0, y_0 - 1) \\
 P_0 \rightarrow P_5 : & (x_0, y_0) \rightarrow (x_0 + 1, y_0 + 1) \\
 P_0 \rightarrow P_6 : & (x_0, y_0) \rightarrow (x_0 - 1, y_0 + 1) \\
 P_0 \rightarrow P_7 : & (x_0, y_0) \rightarrow (x_0 - 1, y_0 - 1) \\
 P_0 \rightarrow P_8 : & (x_0, y_0) \rightarrow (x_0 + 1, y_0 - 1)
 \end{aligned} \tag{2.1}$$

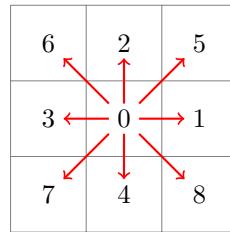


Figure 2.6: The possible transitions, represented as vectors. For the D2Q9 model those vector are also representing the velocity vectors.

Considering the (Figure 2.6), all the transitions are associated to a specific k . This is a particularly simple Markov Chain in which there are a total of *nine* states. Between these states, the transitions always and only start from the P_0 state to go to the others P_k states or itself. The same concept is graphically represented with the diagram in (Figure 2.7). To every transition, it is associated a certain probability of that transition to happen. Formally, this probability is given by the initial and the final states: p_{if} . Since the starting state always stays the same, it is possible to omit it.

Therefore, the probability of the transition from P_0 to P_k is expressed by $p_k(x, y)$, where the index k points to the ending state. It means that for each position in Ω , it is possible to say how likely is

to “step forward” or “turn right” and so on. Then, once it is in the new position, it is again possible to say the most probable direction the pedestrian will choose. The same prediction is applicable to the whole space, mapped by the real data.

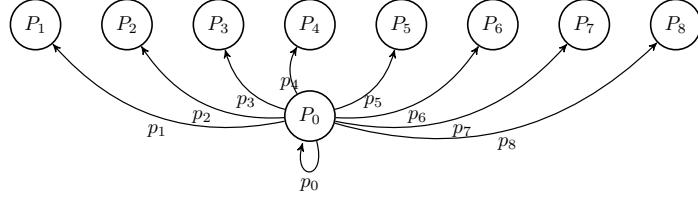


Figure 2.7: The Markov Chain diagram of the system. The states are indicated with circles and labeled with P_k . The transitions are indicated with arrows and labeled with theirs probability p_k

Tensor's dimension With this structure it is possible to create a tensor A with three indices. Taking into account the simplest model, as above, the relative *tensor* is A_{xyk} . Where every entries is the probability p to move along the k direction from the location (x, y) . The total number of elements in A is the product between:

$$\begin{aligned} N(A_{xyk}) &= (\text{dim-x-grid}) \times (\text{dim-y-grid}) \times (\text{dim-k-array}) \\ &= (\text{dim-x-grid}) \times (\text{dim-y-grid}) \times 9 \end{aligned}$$

e.g. in the following paragraphs is used a grid space of 200×100 cells, so the number of entries would became

$$N(A_{xyk}) = 200 \times 100 \times 9 = 180000 .$$

Since the aim of every models is to simulate a pedestrian in the crowd, this tensor is the key to get to the result. In general it's not easy to represent the tensor A in all following models, but it's possible for this first one as plotted in (Figure 2.8). It shows a 3×3 matrix of figures. Each one is referred to a certain value of the k -index. Each figure's position is in reference to the usual *D2Q9* map, similarly as in (Figure 2.1).

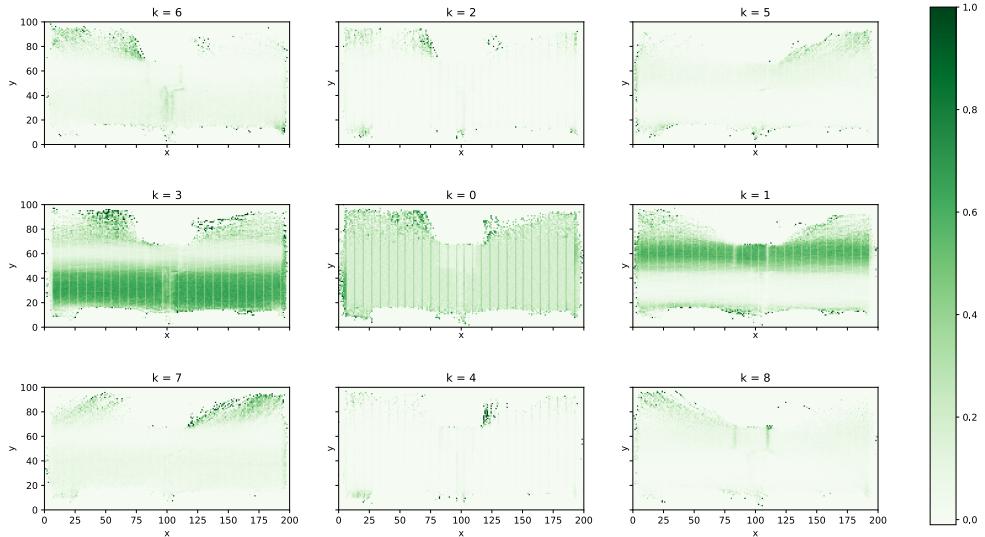


Figure 2.8: Each figure represents the probability in every position to move along a certain direction, defined by the k -index.

2.1.2 Model D2Q9Q9

The conceptual step forward in this work is to consider the next position and the previous one. Similarly to the previous model, this is a *time-independent* model. Hence given a trajectory γ in the grid space of a pedestrian that make a transition for each time step. For each point P_0 of γ it is possible to determinate where it was before at P_{-1} and where is going to be after at P_{+1} . The index that represents the *next* position is k , meanwhile the index that represents the *previous* position is h . For instance it is given the table of the coordinates and the two indexes related to the (Figure 2.7) in the (Table 2.1).

Time step	x_g	y_g	k -index	h -index
1	5	1	3	0
2	4	1	2	1
3	4	2	3	4
4	3	2	2	1
5	3	3	5	4
6	4	4	2	7
7	4	5	2	4
8	4	6	2	4
9	4	7	2	4
10	4	8	6	4
11	3	9	0	8

Table 2.1: This is tabulated the trajectory of the same illustrative pedestrian as above in (Figure 2.7). Here is expressed the position time to time, the index of the following move and the index of the previous move.

Tensor's dimension The *tensor* associated to this model is characterized by a total of four indexes as A_{xykh} . Every element of this tensor is now representing a certain probability to move away from a state to another, but considering also the previous position. The total number of elements in A is the product between:

$$\begin{aligned} N(A_{xykh}) &= (\text{dim-x-grid}) \times (\text{dim-y-grid}) \times (\text{dim-k-array}) \times (\text{dim-h-array}) \\ &= (\text{dim-x-grid}) \times (\text{dim-y-grid}) \times 81 \end{aligned}$$

e.g. in the following paragraphs is used a grid space of 200×100 cells, so the number of entries would became

$$N(A_{xykh}) = 200 \times 100 \times 81 = 1620000 .$$

Taking into account the example in the (Table 2.1), and considering this the only one possible trajectory. It is easy to see that the probability at $A_{xykh} = A_{3,3,5,4} = 1$ is maximum in the position (3, 3). The probability for every other k, h in the same position is zero, $A_{3,3,k,h} = 0$ for $k \neq 5, h \neq 4$. Instead, if two trajectories pass by the same position in the grid but with different directions, this probability is distributed along two directions. This is the scenario represented in (Figure 2.9), where there are two trajectories. Those pass by the same cell at different times, but leave in the model D2Q9Q9 a strong influence. In this case $A_{2,2,5,7} = 1$ and $A_{2,2,6,8} = 1$. For the previous model D2Q9 in the same position it would be, with A_{xyk} , $A_{2,2,5} = 0.5$ and $A_{2,2,6} = 0.5$. In (Table 2.10) are explicitly expressed the positions and the values of k and h for the two illustrative paths.

Cross trajectories A common situation when looking at the trajectories is to find intersections with two different directions. Lets take into account the previous example, where two pedestrians cross the field very differently. One from the left-down to the right-up corners and the other from the right-down to the left-up corners. The intersection that is formed from this two paths is the highlighted green cell in (Figure 2.9). As mentioned before, the simpler model D2Q9 has more problems with this situation than the D2Q9Q9. The reason is that for the D2Q9 it is taken into account the velocity on a particular cell so it is considering instantaneous direction. When simulating pedestrian arrive at that intersection, is inevitable to get a probability to *change* directions and go back. This situation is one of the deeper reason to change model to D2Q9Q9 to take into account also

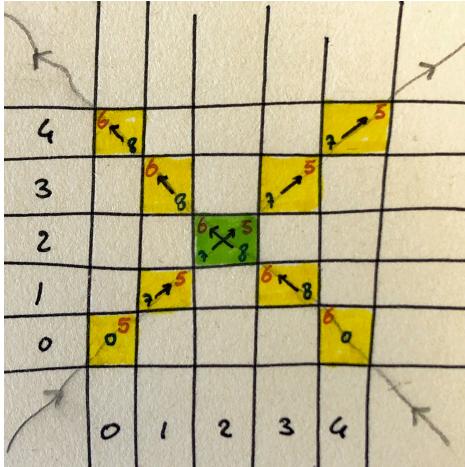


Figure 2.9: Two paths passing by the same cell, in green, with different directions. The numbers in red are defining the values of the k -index for every step. The numbers in blue are the values of the h -index for every step.

Pedestrian	Time step	x	y	k	h
Ped 1	1	0	0	5	0
Ped 1	2	1	1	5	7
Ped 1	3	2	2	5	7
Ped 1	4	3	3	5	7
Ped 1	5	4	4	5	7

Pedestrian	Time step	x	y	k	h
Ped 2	1	0	4	6	0
Ped 2	2	1	3	6	8
Ped 2	3	2	2	6	8
Ped 2	4	3	1	6	8
Ped 2	5	4	0	6	8

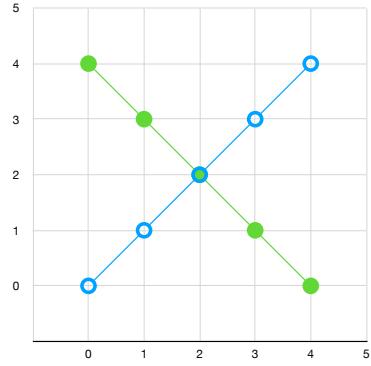


Figure 2.10: Two paths passing by the same cell, with different directions.

the previous position. In fact, in this situation a simulated pedestrian would not change direction for the second model, because it has probability zero to make it.

From (Figure 2.11) to (Figure 2.22), some of the possible changes between cells are shown. Those movements may start going *Up* and evolve in very different ways. The First and the Second example start with the same first transition but diverge in the second movement; this leads to two different ending positions. Anyway, those transitions have something in common even if they end up in different positions. This type of information is contained in the *D2Q9Q9* type of models and not in the *D2Q9* types. This lead to different *interpretations* of what is the expectation in that ending position.

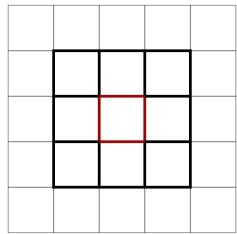


Figure 2.11: Start position.

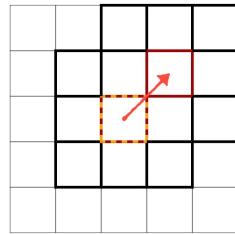


Figure 2.12: First transition Right-Up.

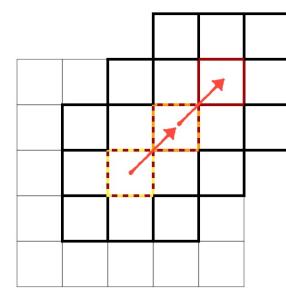


Figure 2.13: Second transition Right-Up.

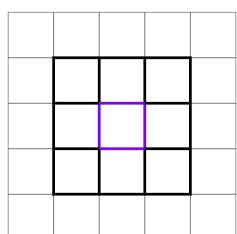


Figure 2.14: Start position.

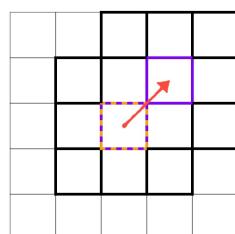


Figure 2.15: First transition Right-Up.

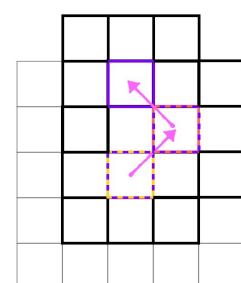


Figure 2.16: Second transition Left-Up.

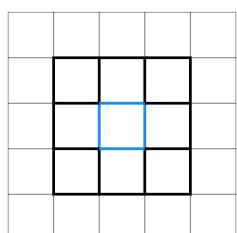


Figure 2.17: Start position.

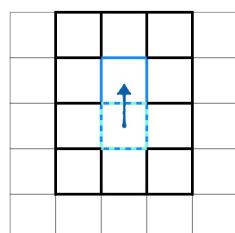


Figure 2.18: First transition Up.

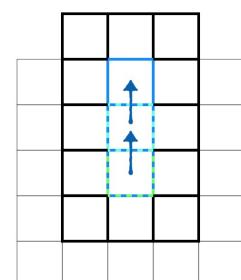


Figure 2.19: Second transition Up.

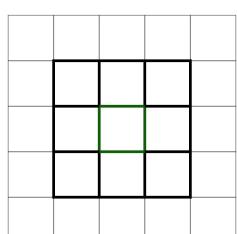


Figure 2.20: Start position.

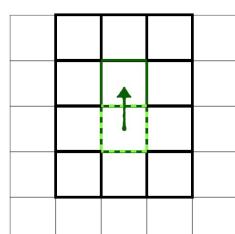


Figure 2.21: First transition Up.

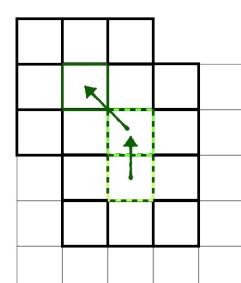


Figure 2.22: Second transition Left-Up.

2.1.3 Model TD2Q9

This module takes into account all the tools offered by the *D2Q9 model*. But time is now relevant and so this is a *time-dependent* model.

Time It is important to describe properly what is *time* in this study. Lets start from what is not: time is not the universal time, like UTC. Time here is discrete and it's defined also as *time step*. It is divided in seconds, using the *unix time* or *UNIX Epoch time*. Every step in time define a new state along the time axes, it is possible to imagine it as a new dimension. For each pedestrian path, time start at the entrance in the field and ends at the exit of it. So time is relative to each trajectories and not global.

The definition of a *state* is not just by the position in space but is given by x, y, t . In this model, pedestrians moves along three axes: two dimensions in space and one in time.

Tensor's dimension The *tensor* representing the probability to move is defined by A_{txyk} . With this structure it is possible to associate the velocity to the time step. The total number of elements in A is the product between:

$$\begin{aligned} N(A_{txyk}) &= (\text{dim-time-grid}) \times (\text{dim-x-grid}) \times (\text{dim-y-grid}) \times (\text{dim-k-array}) \\ &= (\text{dim-time-grid}) \times (\text{dim-x-grid}) \times (\text{dim-y-grid}) \times 9 \end{aligned}$$

e.g. in the following paragraphs is used a grid space of 200×100 cells and the mean dimension in time for significants trajectories is dim-time = 200, so the number of entries would became

$$N(A_{txyk}) = 200 \times 200 \times 100 \times 9 = 36000000 .$$

This gives the possibility to differentiate when a trajectory is going to exit or is just entered, when giving the probability to move. Lets make an example and consider a position close to the map border $P_b = (x_b, y_b)$, something like in (Figure 2.23). If it's not known the time of this position P_b the probabilities to go to the center of the map or out of it are non zero. So it's not possible, given P_b , to really distinguish if the pedestrian is going out or not. But if the time is taken into account it's necessary to distinguish if the pedestrian is at the beginning of it's path or at the end. Lets start again from the position P_b . If it's at the beginning in time steps, the more probable move will be to the center. If some time is passed inside the map, it will have higher probability to go out from the map.

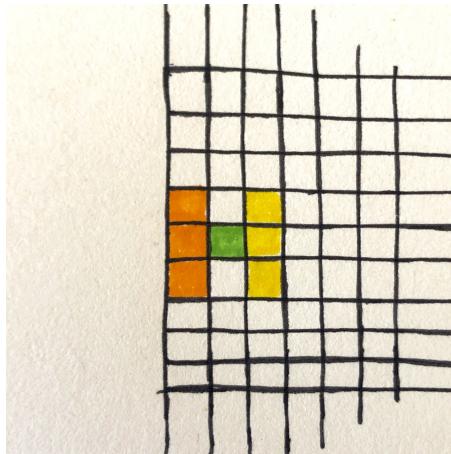


Figure 2.23: The boundary position P_b considered is the green cell. When time is *small*, the trajectory is at the beginning, yellow positions are more likely than orange positions. When time is *big*, the trajectory is at the ending, yellow positions are less likely than orange positions.

2.1.4 Model TD2Q9Q9

This model is the extension of the previous *TD2Q9* and the *D2Q9Q9* method, where time and acceleration are taken into account.

Tensor's dimension With this method it is associated a tensor A_{txykh} , with five dimensions. Using evident notation, in reference to the previous paragraphs, it's dependent on the time t , the position (x, y) , the future position k and the previous position h . With the *TD2Q9Q9*-model is taken into account the information on acceleration in a position combined to the time of corresponding to that position. The total number of elements in A is the product between:

$$\begin{aligned} N(A_{txykh}) &= (\text{dim-time-grid}) \times (\text{dim-x-grid}) \times (\text{dim-y-grid}) \times (\text{dim-k-array}) \times (\text{dim-h-array}) \\ &= (\text{dim-time-grid}) \times (\text{dim-x-grid}) \times (\text{dim-y-grid}) \times 81 \end{aligned}$$

e.g. in the following paragraphs is used a grid space of 200×100 cells and the mean dimension in time for significants trajectories is $\text{dim-time} = 200$, so the number of entries would became

$$N(A_{txyk}) = 200 \times 200 \times 100 \times 81 = 324000000 .$$

Time As before, time is the proper time of each pedestrian. It define the time of a certain step along the whole trajectory.

This very last model studied in this work is the most complex but may lead to a more appropriate simulation. It's also the most computationally expensive because of the great number of items and because it needs a big number of trajectories to completely *fill* the tensor.

2.2 Simulated dynamics

Velocities plot A significant plot to understand those paths is the one that compare the velocity along the two axes x and y . In this example it describes how some trajectories are walking left and others are going right, see the (Figure 2.24). This plot is made as heat-map, that means each cell gives the intensity of that unique combination of velocities.

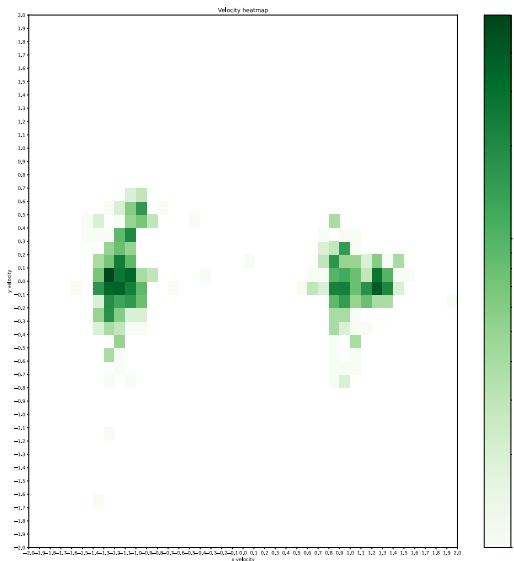


Figure 2.24: Comparison between velocity along the two directions v_x and v_y .

D2Q9 representation Another significant plot is the 3×3 matrix of figures that follows in (Figure 2.25), it is composed by nine images. All those images are referred to the same field, with the same dimensions. In each of those is plotted the move probability along just one direction. The positions of those images is oriented as the *D2Q9* map, showed in (Figure 2.1). So that the center figure represents the probability to stand still, meanwhile the right-center figure is the probability to move right and so on.

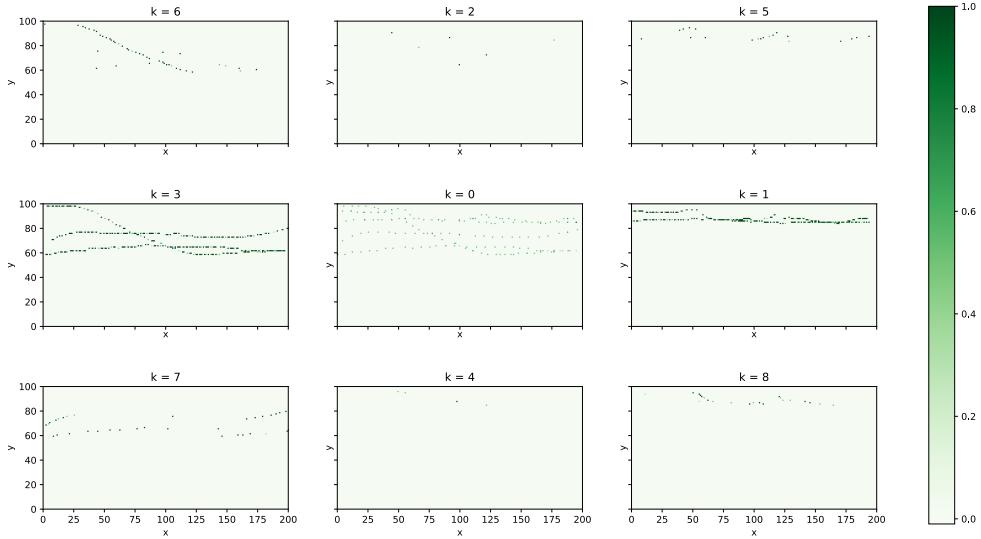


Figure 2.25: Representation of the D2Q9 model. Every plot shows the move probability for each associated direction.

2.2.1 Trajectories simulation

The aim of a good simulation is to be capable of recreate a realistic path. In other words, the aim is to make a good prediction on the path chosen by a pedestrian. To do so, it is necessary to *understand* the trajectories, to *learn* the motion from real life experiment, before trying to simulate it.

Start positions The first step is always the hardest to make, others follows. The simulation has a start on a cell that is considered part of an group of cells with certain characteristics. To define this group is necessary to analyze where a real life trajectory start. Lets consider a raw trajectory, a discrete path made from consecutive points. At every time is associated a position on the x ax and on the y ax. So that a trajectory is described as a group of points in three dimensions, one in time and two in space. With this definition is easy to define a starting position, asking where is the position when the time is minimum. The group of the possible *start positions* is created going through all the trajectories and select the points that correspond to the minimum time for each of those.

Once the group of start position is created, it is possible to assign to a synthetic pedestrian its initial position. In this work the assignation is made by a random sort from the group of named before. It is possible to select a region of interest in the field. Combining an arbitrary portion of space and the group of start position and making a new sub-group. Than the random choose is made from that secondary sub-group.

Step The step from the initial position to the second is essentially made with the same procedure as all further steps. The algorithm take as input the position, in space and time if necessary. The tensor A is used to get the probability for each of the nine directions. So that the initial position, chosen from the group of the start positions, is associated to the time $t = 0$. When this input is given to the algorithm it read the array of the possible transitions from the actual cell to the next. Then it run a Monte Carlo trough that array and returns the corresponding direction randomly chosen with different probabilities. For the second position it will assign time $t = 1$ with the new coordinates, running another step. And so on, one step at the time, moving through the field and increasing the time for each synthetic pedestrian. It is possible to simulate one trajectory or hundred, if more than one it will not consider the interactions between those new synthetic. This fact may be useful to analyze different scenarios in the same environment. As also said before, this model takes into account a real-life environment and make possible the simulations consequentially to the selected scenario. Choosing a different one lead to very different simulations. Choosing a scenario and running a multitude of simulations lead to a complete tree of possible paths. The path that will be followed more will be the most probable one.

Examples to explain the algorithm The (Figure 2.26) represents a scenario where in a certain position it is associated a distribution of probability that make certain the evolution of the system. In the figure is described that is not possible to move anywhere except to the Right direction. For the

second example in (Figure 2.27) is given a different probability distribution. If in a certain position (x_0, y_0) is associated this type of distribution the randomization will be between going Right or going Down with the same probability. For the third example in (Figure 2.28) lets assume every entry non-zero. In this case some of the future positions will have a really low probability to happen and others very high. So that simulating a great number of trajectories will lead to get some of them "choosing" also the less probable directions. For sure the most probable choice is to go Right, the second is to go Down and the third in order of probability is to go Right-Down. All the other directions follows as less probable, but with a non-zero probability.

$p_6 = 0$	$p_2 = 0$	$p_5 = 0$
$p_3 = 0$	$p_0 = 0$	$p_1 = 1$
$p_7 = 0$	$p_4 = 0$	$p_8 = 0$

Figure 2.26: First example of probability distribution for a certain position. Always right.

$p_6 = 0$	$p_2 = 0$	$p_5 = 0$
$p_3 = 0$	$p_0 = 0$	$p_1 = 0.5$
$p_7 = 0$	$p_4 = 0.5$	$p_8 = 0$

Figure 2.27: Second example of probability distribution for a certain position. Always right or down.

$p_6 = 0.02$	$p_2 = 0.01$	$p_5 = 0.05$
$p_3 = 0.10$	$p_0 = 0.01$	$p_1 = 0.40$
$p_7 = 0.05$	$p_4 = 0.20$	$p_8 = 0.16$

Figure 2.28: Third example of probability distribution for a certain position. None zero probability.

Stop the step The simulation of a singular pedestrian has to be stopped by some kind of trigger. The first trigger is applied when the synthetic pedestrian touches the border of the field. The other trigger used in this work is made by setting the maximum value for the proper time of each synthetic pedestrian. Both those triggers must stop the counting of synthetic pedestrian's time and stop calculating the next move for those trajectories. This may lead to a distribution of the trajectories' length. That is force cut at the upper limit, imposed by the simulation setup, and depend on when every trajectory touches the border.

2.2.2 Distribution of probability

The analysis on the probability distribution tensor make possible to determinate *which trajectory is more likely to be chosen*. This method propose an approximation for the continuous general problem. This method is built on a discrete system of time, space and "directions" of the momentum. With this approximation it is possible to evaluate the probability for a trajectory, starting from a certain position. So lets assume the initial position as (x_0, y_0) , the trajectory γ that start from that point which path well follows? If a *tensor* of the probability was created before it is possible to calculate the most probable γ starting from that point. Could be also very interesting to change the question to: how likely is this γ that i'm watching? The answer to the last question may be satisfied by multiplying the value of each transition from the starting point to the end.

Simplistic example Lets take into account the D2Q9-model, so that it's defined by a tensor A_{xyk} , in reference to the (Chap. 2.1.1). Assume a finite grid of cells, a 6×3 matrix, where x is horizontal and y is vertical. Assume that for each position (x, y) is given a vector of *nine* entries with index k . Assume a finite number of possible move distributions, (Equation 2.2). Lets represent the vector in the form of a matrix, referencing to the (Figure 2.1), to help visualization. And call them: A, B, C, D , with the following values:

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.3 \\ 0 & 0 & 0.7 \end{pmatrix} \quad C = \begin{pmatrix} 0 & 0 & 0.7 \\ 0 & 0 & 0.3 \\ 0 & 0 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (2.2)$$

remembering that the sum of the vector's entries must be 1. Vector A force the movement to go Right. Vector B allows two possible directions but going Right is less probable than going Right-Down. Similarly to the previous, vector C allows two possible directions but going Right is less probable than going Right-Up. The last vector D imposes to stand still and for the scope of this example is useful to stop the steps.

The discrete space Ω_g of this example is formed by 18 cells and it's represented in (Figure 2.29), in which one is set the probability distribution. Lets assume that the first position of a synthetic

pedestrian starts from a cell in the first column from the left. Lets also assume that the scope of this simulation is two start from the left side of Ω_g and arrive to the right side. Not all trajectories are permitted, instead only a few are possible. All the possible path are showed in (Figure 2.30) with different colors. When the first position is the middle-left cell the simulation could only evolve in one path, the one represented in blue in the (Figure 2.31). Defining the path in the figure as γ_0 and it's probability as p_{γ_0} . This path as probability to happen equal to $p_{\gamma_0} = 1$ and no other path are allowed from this cell. Meanwhile from the upper-left cell and from the bottom-left cell, three paths are possible, as showed in (Figure 2.32), but not with the same probability. Lets set names for all the trajectories from this cell:

- γ_0 : blue path
- γ_1 : green path
- γ_2 : orange path
- γ_3 : red path

the notation for theirs probability is p_{γ_i} . Each probability can be derived from the series of products of the corresponding transitions values. So that in the previous case would be:

$$p_{\gamma_0} = 1 \times 1 \times 1 \times 1 \times 1 = 1 = 100\%$$

and in fact γ_0 is the only one possible path. In the second case would be instead:

$$\begin{aligned} p_{\gamma_1} &= 1 \times 1 \times 0.3 \times 0.3 \times 1 = 0.09 = 9\% \\ p_{\gamma_2} &= 1 \times 1 \times 0.3 \times 0.7 \times 1 = 0.21 = 21\% \\ p_{\gamma_3} &= 1 \times 1 \times 0.7 \times 1 \times 1 = 0.70 = 70\% \end{aligned} \quad (2.3)$$

The (Equation 2.3) explicitly shows all the possibilities. With this result it's clear witch one is the most probable path in this space.

A	A	B	B	A	D
A	A	A	A	A	D
A	A	C	C	A	D

Figure 2.29: Space of the simplistic example. Every letter correspond to a specific distribution of possible transitions, referred to the (Equation 2.2).

A	A	B	B	A	D
A	A	A	A	A	D
A	A	C	C	A	D

Figure 2.31: A straight path. This path is forced to go straight right because of the distribution A that permits only this movement.

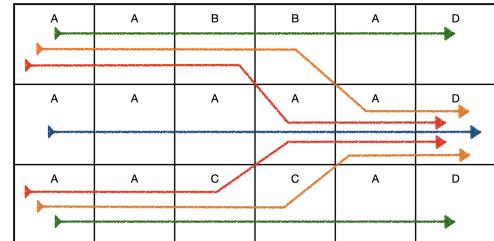


Figure 2.30: All the possible paths that are permitted to travel from the left to the right side of the map Ω_g . Different colors represents different probabilities.

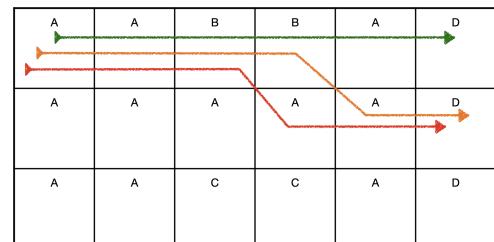


Figure 2.32: The three possible paths when starting from the upper-left cell. This situation is specular to when starting from the bottom-left cell.

Real-life situation The previous example is a simplification of a real study case. In fact it is easy enough to calculate by hand the possibles paths. The dataset showed before in (Figure 2.4) has a grid space dimensions of 200×100 cells. In that real scenario the field Ω_g is larger than the simplified example of the previous paragraph. The grid dimensions depend always on two factor: the physical space dimensions (in meters) and the choice of the grid size. The (Figure 2.25) express the values of each direction in correlation to the map position. That figure is strictly liked to the matrices showed in (Equation 2.2), but this is referred to the real case. Even this scenario is simplified because it takes into account only five trajectories. For comparison lets consider the entire dataset, the (Figure 2.8) shows how complex may became the representation when this transitions are calculated for a grater multitude of real pedestrian trajectories.

Chapter 3

Results

3.1 Comparison of 3D histograms

The discrete path integral delineated with the models above is capable of great predictions. But this structures are hard to represents, moreover the models that includes more than 3 indexes is impossible to represents. To solve this situation here is used a 3D-histogram. As histogram this plot express the statistic relevance of a certain data that occurred in dataset. The 3D space dimensions are: the x direction, the y direction and the *time* dimension that goes upward. A 2D surface here represents points in the space-time that have the same occurrence, it's also called *isosurface*. The following dataset is selected from the Floorfield10 - Utrecht Station.

3.1.1 3D comparison between real data and simulations

In the following results it's distinctive the surface shape, a *tube* isosurface given by the same statistical occurrence. The shape is the result of a symmetrical distribution around the center of the most probable path.

Whats follows doesn't represents the maximum probability nor the minimum, it shows a certain occurrence in the middle that may be useful to compare different datasets. In this work are considered the **real data** and the four simulations: **simD2Q9**, **simD2Q9Q9**, **simTD2Q9**, **simTD2Q9Q9**. Simulations are generated starting from respectively the four models: *D2Q9*, *D2Q9Q9*, *TD2Q9*, *TD2Q9Q9*.

In the figures below are showed the isosurfaces colored by type:

- BLACK: RealData;
- LIGHT-BLUE: Simulation D2Q9;
- PURPLE: Simulation D2Q9Q9;
- GREEN: Simulation TD2Q9;
- RED: Simulation TD2Q9Q9.

Plot of the RealData alone In (Figure 3.1) and (Figure 3.2) it's represented a *tube* surface made from the same statistical occurrence in real data.



Figure 3.1: Real data. Top view: where the time direction is pointing out of the plot, it's clearly visible the shape along the spaces directions.

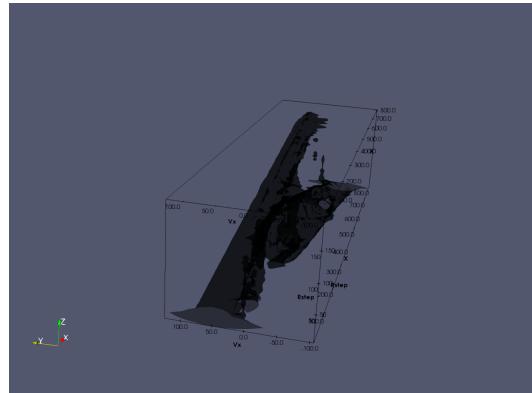


Figure 3.2: Real data. Side view: it's possible to distinguish the three dimensions. The shape goes UP in time and move horizontally in space.

Comparison between RealData and Simulations The following 8 figures, from (Figure 3.3) to (Figure 3.10), show the same statistical occurrence in datasets. From the first to the last model it's over and over clearer the good overlay between the simulation and the real data. For the firsts two models is pretty difficult to see a good overlaying, even if it's not null. For the lasts two models is easy to see the improve made by adding the *time* information.

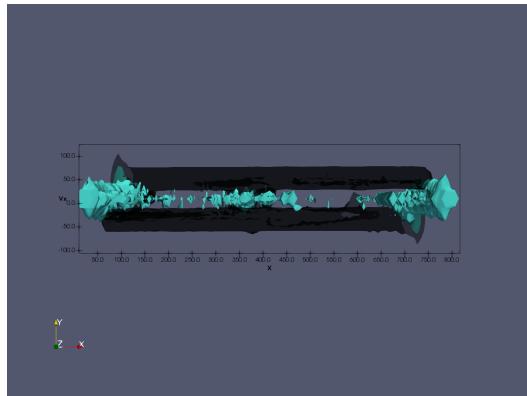


Figure 3.3: Top view: model D2Q9 in light blue and real data in black.

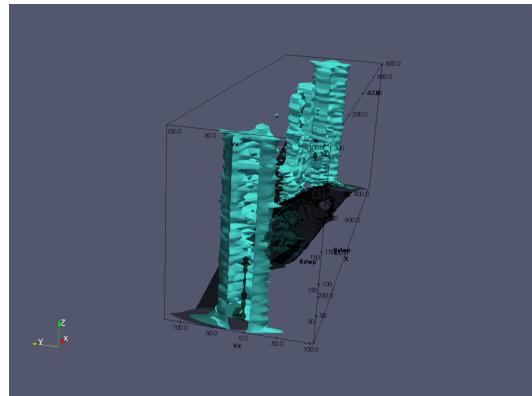


Figure 3.4: Side view: model D2Q9 in light blue and real data in black.

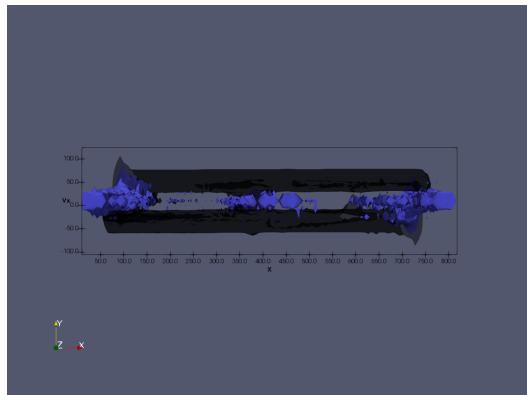


Figure 3.5: Top view: model D2Q9Q9 in purple and real data in black.

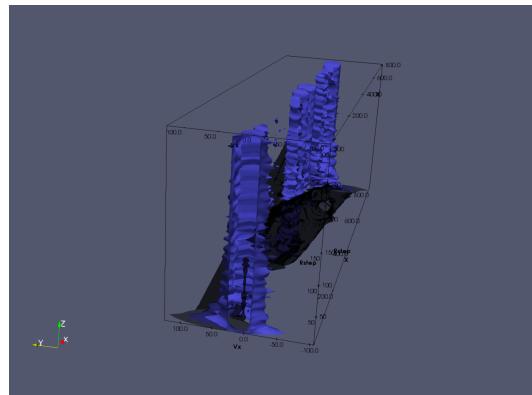


Figure 3.6: Side view: model D2Q9Q9 in purple and real data in black.

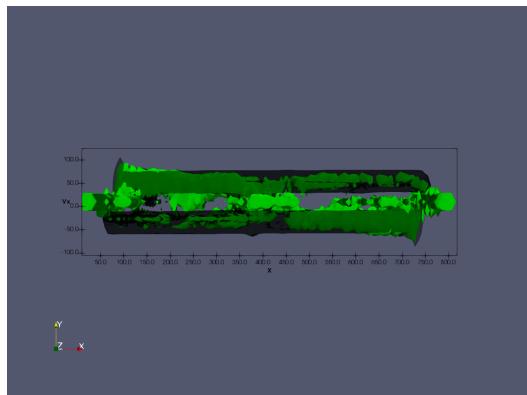


Figure 3.7: Top view: model TD2Q9 in green and real data in black.

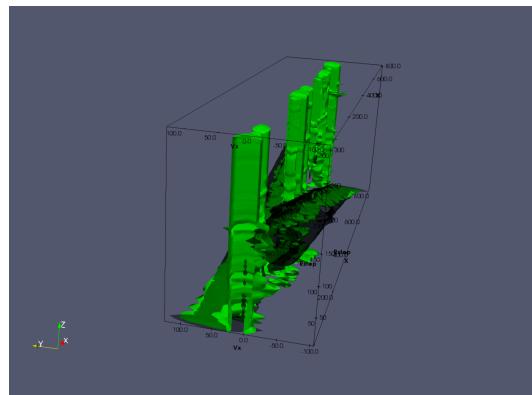


Figure 3.8: Side view: model TD2Q9 in green and real data in black.

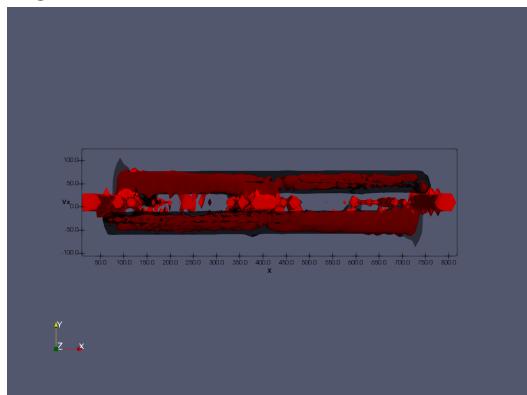


Figure 3.9: Top view: model TD2Q9Q9 in red and real data in black.

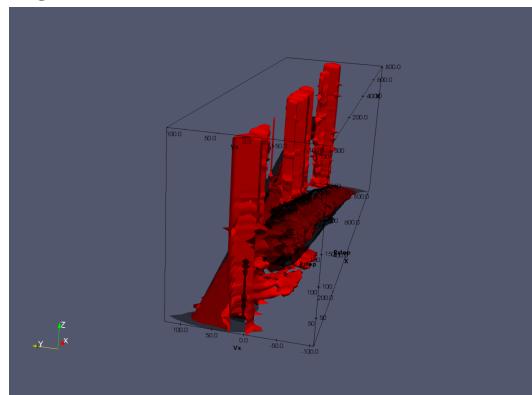


Figure 3.10: Side view: model TD2Q9Q9 in red and real data in black.

3.2 Simulated dynamics with probability models

The aim of this paragraph is the comparison between the real dataset utilized to produce the models and some simulations. The following quantities are taken into account to compare the results. Those are plotted as histograms, to empathize the statistical approach.

Different quantities has been considered in the following plot types:

- (i) The magnitude of the velocity vector along the \vec{x} axes, plotted as 1-dimensional histogram;
- (ii) the magnitude of the velocity vector along the \vec{y} axes, plotted as 1-dimensional histogram;
- (iii) the correlation between the position along the \vec{x} axes and the magnitude of the velocity vector along the same axes, plotted as heat-map or 2-dimensional histogram;
- (iv) the correlation between the position along the \vec{y} axes and the magnitude of the velocity vector along the same axes, plotted as heat-map or 2-dimensional histogram.
- (v) the heat-map of the positions along \vec{x} and \vec{y} axis of all paths that have passed though, plotted as 2-dimensional histogram;

3.2.1 Plots

In this section are presented the results as plots, generated form the real data and from the dynamic simulations made from all the four models explained in Chapter 2.

The plots form (Figure 3.11) to (Figure 3.15) represents the data about the (i) quantity. Plots from (Figure 3.16) to (Figure 3.20), that represents the (ii) quantity. Those quantities (i) and (ii) are representing the velocity distribution on theirs own axes.

A good correspondency between those images characterize a good modelization. Reading those from left to right makes possible to distinguish an increasing in the quality of the simulated dynamics. But these quantities are not too significant by themself, because they take in to account just the magnitude of the velocity vector.

The plots form (Figure 3.21) to (Figure 3.25) represents the data about the (iii) quantity.

The (Figure 3.21) shows two main velocities, this represents a very strong bi-directional flow. These plots represents the distribution of the velocity in correspondence to the position along x .

The increasing of the model's complexity shows, from left to right, an increasing of the accuracy. The dynamic simulated data doesn't shows the main velocities as defined as it is in the real condition. This may depending on the number of simulated trajectories, that could be convenient to increase in future research.

The plots form (Figure 3.26) to (Figure 3.30) represents the data about the (iv) quantity. These plots represents the distribution of the velocity in correspondence to the position along y .

The (Figure 3.26) shows one main velocity around the zero value, it means that a major quantity of trajectories proceed horizontally (x -direction) and not vertically (y -direction). These plots shows from left to right an increasing accuracy, even more readable than the previous on the other direction.

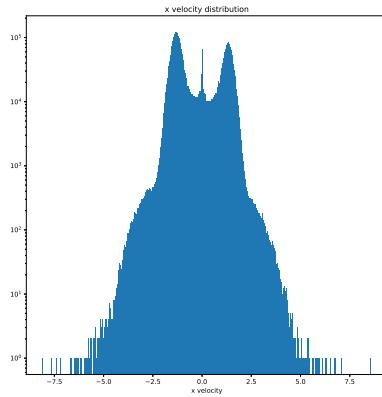


Figure 3.11: Plot type (i). Real-life data distribution

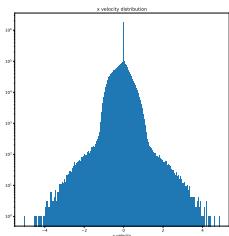


Figure 3.12:
Plot type (i).
Simulated dynamic
using D2Q9

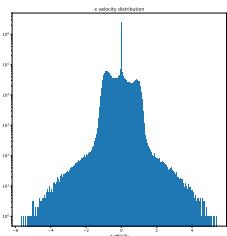


Figure 3.13:
Plot type (i).
Simulated dynamic
using D2Q9Q9

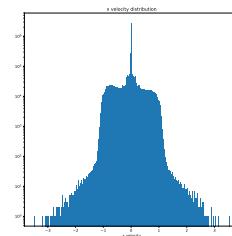


Figure 3.14:
Plot type (i).
Simulated dynamic
using TD2Q9

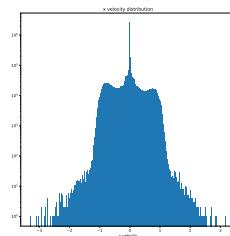


Figure 3.15:
Plot type (i).
Simulated dynamic
using TD2Q9Q9

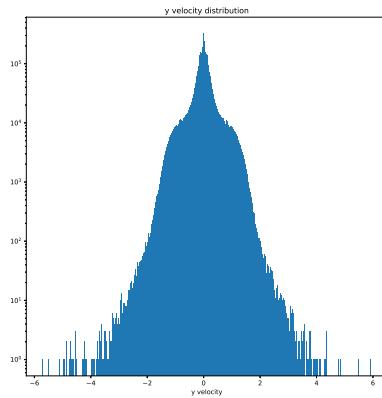


Figure 3.16: Plot type (ii). Real-life data distribution

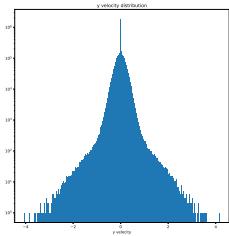


Figure 3.17:
Plot type (ii).
Simulated dynamic
using D2Q9

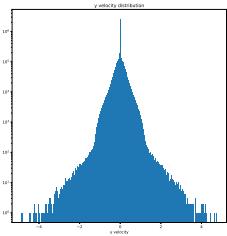


Figure 3.18:
Plot type (ii).
Simulated dynamic
using D2Q9Q9

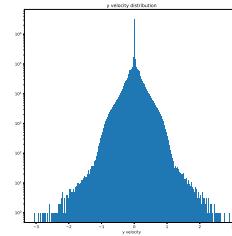


Figure 3.19:
Plot type (ii).
Simulated dynamic
using TD2Q9

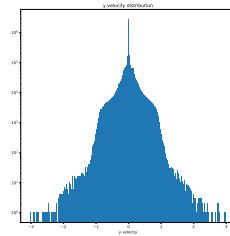


Figure 3.20:
Plot type (ii).
Simulated dynamic
using TD2Q9Q9

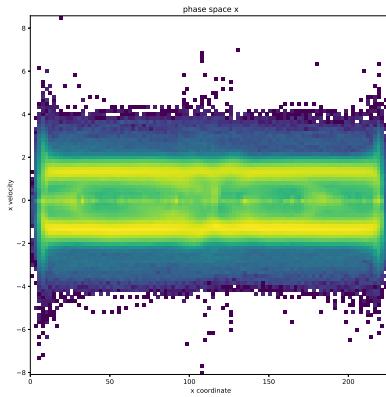


Figure 3.21: Plot type (iii). Real-life data distribution

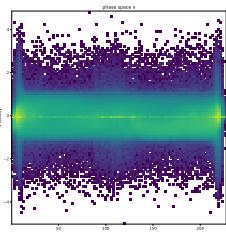


Figure 3.22:
Plot type (iii).
Simulated dynamic
using D2Q9

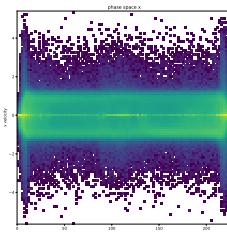


Figure 3.23:
Plot type (iii).
Simulated dynamic
using D2Q9Q9

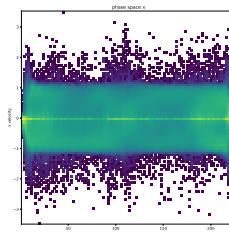


Figure 3.24:
Plot type (iii).
Simulated dynamic
using TD2Q9

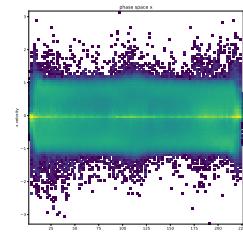


Figure 3.25:
Plot type (iii).
Simulated dynamic
using TD2Q9Q9

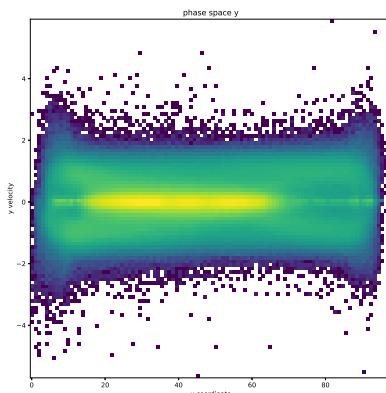


Figure 3.26: Plot type (iv). Real-life data distribution

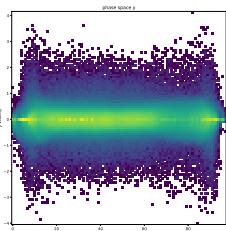


Figure 3.27:
Plot type (iv).
Simulated dynamic
using D2Q9

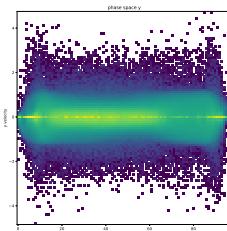


Figure 3.28:
Plot type (iv).
Simulated dynamic
using D2Q9Q9

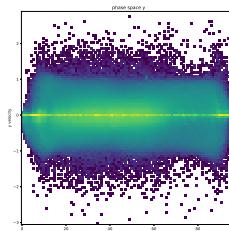


Figure 3.29:
Plot type (iv).
Simulated dynamic
using TD2Q9

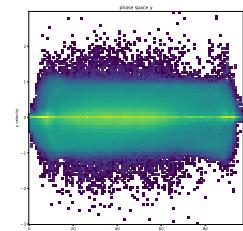


Figure 3.30:
Plot type (iv).
Simulated dynamic
using TD2Q9Q9

The plots from (Figure 3.31) to (Figure 3.35) represent the data about the (v) quantity. The first plot in (Figure 3.31) shows where it is more probable to find a pedestrian, in the considered field, based on real data. The others from left to right represent the distribution of the position for every simulated data.

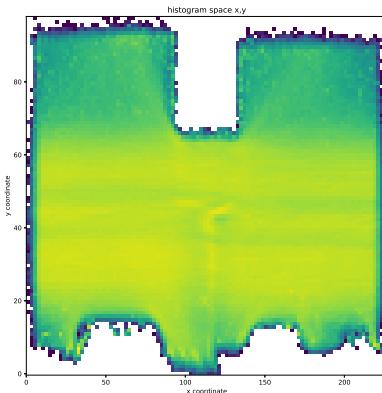


Figure 3.31: Plot type (v). Real-life data distribution

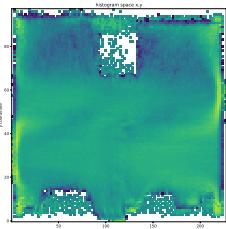


Figure 3.32:
Plot type (v).
Simulated dynamic
using D2Q9

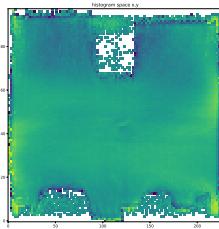


Figure 3.33:
Plot type (v).
Simulated dynamic
using D2Q9Q9

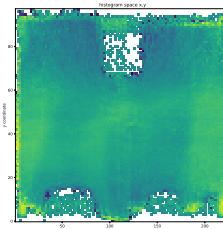


Figure 3.34:
Plot type (v).
Simulated dynamic
using TD2Q9

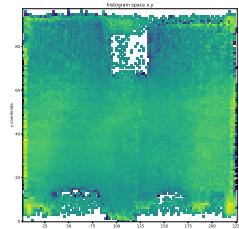


Figure 3.35:
Plot type (v).
Simulated dynamic
using TD2Q9Q9

In this study case there is a strong x component in the pedestrian's motion. A great number of trajectories go through the space from left to right or vice-versa with a *quasi*-straight horizontal path. A minor number of pedestrians make they way going from up to down or vice-versa through the field with a *quasi*-straight vertical line. Some others trajectories make a visible change in direction, e.g. from going straight horizontal to right-up when exiting the area. This dataset has a *good* variety of cases, that sufficiently represents how pedestrians may move along the field. With this it's possible to generate synthetic pedestrian made by simulating theirs dynamic, entirely based on real observation.

Chapter 4

Conclusion

This work is leaded by a statistical approach, which is possible thanks to the access at a large quantity of data. Its focus was to generate a model, a mathematical framework, that learns the probability distribution based on real world situation. A total of four different models have been studied, during the period of study in Eindhoven, and are presented in Chapter 2. The results in Chapter 3 were given generating the dynamic simulation for each model, finding similar paths to the real original data. The approach in this thesis is applied to the pedestrian problem and, for a statistical and data-driven point of view, it is really capable in learning the behaviour of those paths. The technological tools developed to make this work possible are themself a positive result for this thesis.

This work has proven that the concept of a statistical approach to this problem may lead to positives results and applications. Firstly, the dynamic simulation generate pedestrians that may move only where real data have moved through. So it is not necessary to implement boundary conditions, excepts for borders' map, because the framework does it *by construction*. This concept has a deep and strong impact on the study's results and it also simplify the model itself. Secondly, it was shown that this tools may be applied to a great number of information and still running on an average portable computer. So that it opens the possibilities to in-loco solutions for new experiments. Of course, by increasing the number of data and the complexity of the model it would require a greater amount of calculation capacity.

In future works, the same approach may be applied to other systems, that require the same statistical and data-driven point of view: to study the satellites' orbit around Earth, the paths given by GPS datas may be easily analyzed, it's also possible the study the path given by the neurons activity. In general, this solution is applicable where a complex system is nor deterministic nor ordinated and the statistical approach may lead to a solution or a part of it.

Bibliography

- [1] A. Mayer, T. Skowroneck and R. Blanton, “*The Science of Walking*”, University of Chicago Press, 2020. doi: 10.7208/chicago/9780226352480.001.0001. [Online]. Available: <https://doi.org/10.7208/chicago/9780226352480.001.0001>.
- [2] F. Martinez-Gil, M. Lozano, I. García-Fernández and F. Fernández, “*Modeling, evaluation, and scale on artificial pedestrians*”, ACM Computing Surveys, vol. 50, no. 5, pp. 1–35, Nov. 2017. doi: 10.1145/3117808. [Online]. Available: <https://doi.org/10.1145/3117808>.
- [3] D. C. Duives, W. Daamen and S. P. Hoogendoorn, “*State-of-the-art crowd motion simulation models*”, Transportation Research Part C: Emerging Technologies, vol. 37, pp. 193–209, Dec. 2013. doi: 10.1016/j.trc.2013.02.005. [Online]. Available: <https://doi.org/10.1016/j.trc.2013.02.005>.
- [4] G. Le Bon, “*The Crowd: A Study of the Popular Mind*”, Courier Corporation, 2002.
- [5] Xovis AG (Switzerland), Xovis USA Inc. (United States of America), “*Rethinking People Flow*”, www.xovis.com, 2022.
- [6] A. Corbetta, L. Bruno, A. Muntean and F. Toschi, “*High statistics measurements of pedestrian dynamics*”, Transportation Research Procedia, vol. 2, pp. 96–104, 2014.
- [7] A. Corbetta, J. Meeusen, C. min Lee and F. Toschi, “*Continuous measurements of real-life bidirectional pedestrian flows on a wide walkway*”, Proceedings of Pedestrian and Evacuation Dynamics (Special Issue on Collective Dynamics), pp. 18–24, 2016.
- [8] A. Corbetta, W. Kroneman, M. Donners, A. Haans, P. Ross, M. Trouwborst, S. Van De Wijdeven, M. Hultermans, D. Sekulovski, F. Van Der Heijden et al., “*A large-scale real-life crowd steering experiment via arrow-like stimuli*”, Collective Dynamics, vol. 5, pp. 61–68, 2020.
- [9] C. A. S. Pouw, F. Toschi, F. van Schadewijk and A. Corbetta, “*Monitoring physical distancing for crowd management: Real-time trajectory and group analysis*”, PLOS ONE, vol. 15, no. 10, D. R. Chialvo, Ed., e0240963, Oct. 2020. doi: 10.1371/journal.pone.0240963. [Online]. Available: <https://doi.org/10.1371/journal.pone.0240963>.
- [10] J. Adrian, N. Bode, M. Amos, M. Baratchi, M. Beermann, M. Boltes, A. Corbetta, G. Dezecache, J. Drury, Z. Fu, R. Geraerts, S. Gwynne, G. Hofinger, A. Hunt, T. Kanders, A. Kneidl, K. Konya, G. Köster, M. Küpper, G. Michalareas, F. Neville, E. Ntontis, S. Reicher, E. Ronchi, A. Schadschneider, A. Seyfried, A. Shipman, A. Sieben, M. Spearpoint, G. Sullivan, A. Templeton, F. Toschi, Z. Yücel, F. Zanlungo, I. Zuriguel, N. van der Wal, F. van Schadewijk, C. von Krüchten and N. Wijermans, “*A glossary for research on human crowd dynamics*”, English, Collective Dynamics, vol. 4, pp. 1–13, 2019, issn: 2366-8539. doi: 10.17815/CD.2019.19.
- [11] A. Corbetta, “*Multiscale crowd dynamics: physical analysis, modeling and applications*”, 2016, isbn 978-90-386-4014-3, [online] Available: https://pure.tue.nl/ws/files/13621142/20160201_Corbetta.pdf.
- [12] V. J. Blue and J. L. Adler, “*Emergent fundamental pedestrian flows from cellular automata microsimulation*”, Transportation Research Record: Journal of the Transportation Research Board, vol. 1644, no. 1, pp. 29–36, Jan. 1998. doi: 10.3141/1644-04. [Online]. Available: <https://doi.org/10.3141/1644-04>.
- [13] V. J. Blue and J. L. Adler, “*Cellular automata microsimulation of bidirectional pedestrian flows*”, Transportation Research Record: Journal of the Transportation Research Board, vol. 1678, no. 1, pp. 135–141, Jan. 1999. doi: 10.3141/1678-17. [Online]. Available: <https://doi.org/10.3141/1678-17>.

- [14] L. Yang, D. Zhao, J. Li and T. Fang, “*Simulation of the kin behavior in building occupant evacuation based on cellular automaton*”, Building and Environment, vol. 40, no. 3, pp. 411–415, Mar. 2005. doi: 10.1016/j.buildenv.2004.08.005. [Online]. Available: <https://doi.org/10.1016/j.buildenv.2004.08.005>.
- [15] A. Varas, M. Cornejo, D. Mainemer, B. Toledo, J. Rogan, V. Muñoz and J. Valdivia, “*Cellular automaton model for evacuation process with obstacles*”, Physica A: Statistical Mechanics and its Applications, vol. 382, no. 2, pp. 631–642, Aug. 2007. doi: 10.1016/j.physa.2007.04.006. [Online]. Available: <https://doi.org/10.1016/j.physa.2007.04.006>.
- [16] M. Bierlaire, “*Biogeme: A free package for the estimation of discrete choice models*”, in Swiss transport research conference, 2003.
- [17] R. Lubas, J. Wa.s and J. Porzycki, “*Cellular automata as the basis of effective and realistic agent-based models of crowd behavior*”, The Journal of Supercomputing, vol. 72, no. 6, pp. 2170–2196, Apr. 2016. doi: 10.1007/s11227-016-1718-7. [Online]. Available: <https://doi.org/10.1007/s11227-016-1718-7>.
- [18] A. Lerner, Y. Chrysanthou and D. Lischinski, “*Crowds by example*”, Computer Graphics Forum, vol. 26, no. 3, pp. 655–664, Sep. 2007. doi: 10.1111/j.1467-8659.2007.01089.x. [Online]. Available: <https://doi.org/10.1111/j.1467-8659.2007.01089.x>.
- [19] J. Porzycki, R. Lubas, M. Mycek and J. Wa.s, “*Dynamic data-driven simulation of pedestrian movement with automatic validation*”, in Traffic and Granular Flow 13, Springer International Publishing, Nov. 2014, pp. 129–136. doi: 10.1007/978-3-319-10629-8_15. [Online]. Available: https://doi.org/10.1007/978-3-319-10629-8_15.
- [20] E. Ju, M. G. Choi, M. Park, J. Lee, K. H. Lee and S. Takahashi, “*Morphable crowds*”, ACM Transactions on Graphics, vol. 29, no. 6, pp. 1–10, Dec. 2010. doi: 10.1145/1882261.1866162. [Online]. Available: <https://doi.org/10.1145/1882261.1866162>.
- [21] 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*”, Computer Graphics Forum, vol. 31, no. 2pt2, pp. 489–498, May 2012. doi: 10.1111/j.1467-8659.2012.03028.x. [Online]. Available: <https://doi.org/10.1111/j.1467-8659.2012.03028.x>.
- [22] S. Kim, A. Bera, A. Best, R. Chabra and D. Manocha, “*Interactive and adaptive data-driven crowd simulation*”, in 2016 IEEE Virtual Reality (VR), IEEE, Mar. 2016. doi: 10.1109/vr.2016.7504685. [Online]. Available: <https://doi.org/10.1109/vr.2016.7504685>.
- [23] K. H. Lee, M. G. Choi, Q. Hong and J. Lee “*Group behavior from video: A data-driven approach to crowd simulation*”, in Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ser. SCA ’07, San Diego, California: Eurographics Association, 2007, 109118, isbn: 9781595936240.
- [24] Paul A. Gagniuc, “*Markov chains: from theory to implementation and experimentation*”, Hoboken, NJ : John Wiley & Sons, 2017, ISBN 9781119387572