Università degli Studi di Ferrara
Corso di Laurea in Fisica

Statistical learning and simulating the paths of walking pedestrians

Relatori:
Prof. Alessandro Drago
Prof. Federico Toschi

Correlatore:
Dr. Alessandro Corbetta

Laureando:
Dario Chinelli

24 marzo 2022

Anno Accademico 2020 – 2021

Abstract

[ENG] The dynamics of pedestrian changes considerably depending on the surrounding space, not just for the intrinsic chaotic movements that people does walking but also due to the reciprocal collisions and environment condition. We have considered some scenarios to implement models and a tools that can gives us simulations of the movements of a single pedestrian. In order to properly simulate a pedestrians' dynamic, is to have information about the probability to change direction after every step, in every positions of the trajectory. This approach is linked to the path integral in the way that: given a trajectory, it's possible to say with certain probability where the next step is. This mathematical approach is computationally expensive, even more with the big amount of data we are using. So we started implementing a discrete system and a easy model and than we moved to more complex model. In total we get four types of models: two time dependent and tow independent. From now on we'll call those: D2Q9 and D2Q9Q9 the firsts two; TD2Q9 and TD2Q9Q9 the others two.

[ITA] L'obiettivo scientifico è stato creare un metodo, ispirato al Lattice-Boltzman, con cui apprendere, a partire da dati reali, la dinamica pedonale e quantificarla in termini di matrici di transizione su reticolo. L'obiettivo fondamentale è riuscire a quantificare il campo di probabilità, trovato utilizzando diversi modelli. Questo campo ci permette di studiare la dinamica e creare simulazioni di pedoni e traiettorie le cui statistiche sono indistinguibili per costruzione dalle statistiche delle traiettorie reali

Contents

Chapter 1

Introduction

The aim of this work is to clarify the possibility to analyze real life datasets and simulate the pedestrian crowd starting from the Lattice model.

1.1 Assimilating pedestrian dynamics

The dynamics of pedestrians is essentially a chaotic motion in witch multiple conditions and forces are applied to the system. The motion of a single pedestrian in a crowd is a similarly complex problem. Since everywhere in the world is possible to find and watch walking pedestrian, is not as simple to acquire data about their motion. So the first problematic issue is the data acquisition. A possible, and first, solution to this is the video recording of a spot. However this choice lead to others problem such us privacy leak and the object tracking from the video. Still the first issue doesn't have a real solution except if faces are blurred. During the lasts decades the development of machine learning and imaging recognition has provided more tools to analyze this type of data. So that is now possible to elaborate the image of a video surveillance system and obtain analytic datas. Other recent technological advancements have also enabled real-life high-accuracy measurements of pedestrian trajectory. The data are acquired through the usage of overhead depth-sensing cameras. This second 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 is better in prediction. The assumption is that a crowd produces an *effective potential.* Due to the statistical approach this potential is also a probabilistic model, that make, or not, possible a good prediction based on probabilities. The *probability* is inducted by the real-data observation. The path of synthetic pedestrian is given by a Monte Carlo simulation that defines the probability to move in every direction given a position. To make this study possible the space and time discretization is essential.

1.2 Challenges

Starting from real data how can we define a good model to simulate a pedestrian in the crowd flow? In this type of system there is a multitude of *forces* that determinate the path of a single pedestrian. So let's take into account a single pedestrian P that walks in a certain space. The first type of interaction is the structure where P can or cannot walk thought, that 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, that is variable due the circumstance and it is not easy to be analytically determinate. A third type of interaction are random events along the P's path, real world events. ...

How to visualise the pedestrian's path using a multi-dimensional histogram? It is possible to plot every single trajectory, but this lead to a chaotic data representation and not so functional nor readable. It is also easily possible to plot the *heatmap* of a dataset to analyse the most "walked" areas. Even if this second plot choice can takes into account more trajectories than the first and still be readable, it has a problem. This second lead to a representation where the time dependency is completely lost. ...

1.3 Relevance

Chapter 2

Pedestrian dynamics measurements

2.1 Idea of the recording technique

The recording techniques employed are two: Kinect based technology and ProRail security cameras. In both cases the field of view is covered using multiple cameras working together. Starting from the row images, each object is tracked down along its entire path. This is possible using imaging recognition software. The software give as output a data-frame with coordinates and time for each pedestrian.

2.1.1 Kinect: depth map based pedestrian tracking

2.1.2 ProRail cameras

2.2 Presenting datasets

2.2.1 Glow experiment

The experiment was realized during the Glow Expo in 2017. The idea is that people that enter from a single entrance have to chose between two possible exits, because of the obstacle in the middle. In the (Figure 2.1) is represented a digital reconstruction of the domain.

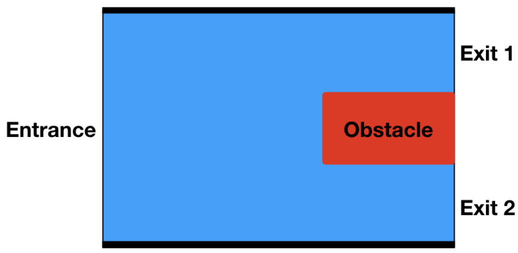


Figure 2.1: Reconstruction of the *glow* experiment, point of view from above.

2.2.2 Utrecht Centraal (Floorefield 10)

In collaboration with ProRail - the company responsible of the train's stations in Netherlands - we had the possibility to use data from the Utrecht's train station. The (Figure 2.2) represents the camera's point of view of the analyzed domain in the station. This is an interesting spot given that this square has three free sides where people could walk through. It's also a huge corridor and an highly crossed spot, that increase the statistic. With this domain we could study if the simulations we're doing are correct also in cases where people with different directions cross the same coordinates in the map. In fact - as described in the following paragraphs - the representation with the 3-dimensional histogram shows us a sort of *cross* X.

Figure 2.2: Utrecht Centraal, cameras point of view (Floorfield 10)

2.3 Technique of measure

Technique of measure documentation intro here

2.3.1 Kinect sensor

Technique of measure with Kinect sensor documentation intro here

2.3.2 ProRail data

Technique of measure with ProRail data documentation intro here

Chapter 3

Background

3.1 Markov Chain

A *Markov Chain* is a stochastic model. It predicts the future outcoming state based on the present state. In other words, the present state determinates the probabilities for every possible future outcomes. The MC may be represented as a diagram, (Figure 3.1a), where the arrows are the possible transitions. A number $p \in (0, 1)$ may also be indicated on the arrows, it specify the probability of that transition. Another model's representation is a *stochastic matrix*, from now on called P matrix. The matrix's entries P_{ij} have as row-index i the starting state and as column-index j the ending state of the system. So that every entries are referred to a specific transition. A two-state Markov

(a) The diagram of a two-state Markov Chain

(b) The transition matrix, also named the Markov matrix

Figure 3.1: From the Markov diagram to the Markov matrix of a tow-state system

chain is the most basic model which can be used for the illustration of the Markov process. The diagram in (Figure 3.1) represents the possibility that the system has to 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 **??**) are positives numbers from 0 to 1 that represent the probability of changing state. The sum on the outgoing arrows must be equal to 1.

3.2 Cellular-Automata

Cellular-Automata documentation here

3.3 Lattice-Boltzman

Lattice-Boltzman documentation here

Chapter 4

Propose data assimilation technique

4.1 Approximation of path integrals: 3D histogram

Approximation of path integrals: 3D histogram documentation here

4.2 Learning transition matrices from data

In this study there is a total of four models: *D2Q9*, *D2Q9Q9*, *TD2Q9*, *TD2Q9Q9*. The firsts two are only dependent by the position in space, also called *time-independents*. The others two are dependent by the position and time, also called *time-dependents*. Whereas there is also a distinction between the D2Q9s and the D2Q9Q9s. For the D2Q9s what it's doing is considering the velocity from a cell to another, so just the change in position. For the D2Q9Q9s it's also considering the acceleration, so the change in velocity. The starting point of each one is the dataset, collected form a real life situation. Since each of them are entirely based on real world pedestrian's path in a crowd, those models simulates an *effective potential* (EP). This potential considers the imposed limit due to the presence of others pedestrians, such as pedestrians tend to not collide each others. It also considers the boundary condition given by the structural environment. The strong point of this EP is that is generated by the real world observation and not built by hand. With the aim of reproducing realistic pedestrians movements, synthetic paths are created from the models. Every model generate one trajectory that simulate just one pedestrian in a statistical crowd. When simulating more paths it consider pedestrian that walks alone in the crowd. This model doesn't consider the interaction made by the others simulated pedestrians.

Notation Lets assume $\gamma = \gamma(\vec{x}_c)$ a pedestrian's path, where $\vec{x}_c = (x_c, y_c)$ has a bi-dimensional spacial dependancy. Given a field Ω_c, the continuous space where pedestrians are tracked, the path γ in that space has a start position A and an end position B. The field Ω is than divided into *rectangular* cells, dividing the real space along x, whit maximum extension indicated as L_x, in a certain number of cells D_x; as well for the y-direction, with obvious notation: L_y and D_y. After this discretization is obtained a *grid space* Ω_g. Where every path γ is converted from continuous $\gamma = \gamma(x_c, y_c)$ to discrete coordinate $\gamma = \gamma(x_g, y_g)$, referred to the *grid*. To lighten up the notation when speaking of *grid space* it is simply used (x, y) in reference to the discrete grid position.

The standard D2Q9 configuration In reference to the (Figure 4.1). This *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 4.2), 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 $k = 0, 1, ..., 8$ 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, for every position in time is also associated a transition number, that represents where is going to go in the next step. If this transition is associated to the change in position it identify a certain velocity, as vector, with a certain direction.

In reference to images from (Figure 4.4) to (Figure 4.15), are shown some possible changes between cells.

6	2	5
3	0	1
7	4	8

Figure 4.1: Index associated to possible movements from the center cell to another. Every transition is associated with a number for the index k. The figure represents how to identify the nine cells with the k index.

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

Figure 4.2: 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).

Iterating this procedure to the entire pedestrian's trajectory it is possible to get something like what's illustrated in the (Figure 4.3). 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 with arrows that are consistent with the velocity arrows in each position. The numbers are the value of the k-index in each position, it is solid with the maps above. This lead the discussion directly to the first model $D2Q9$ in the next paragraph.

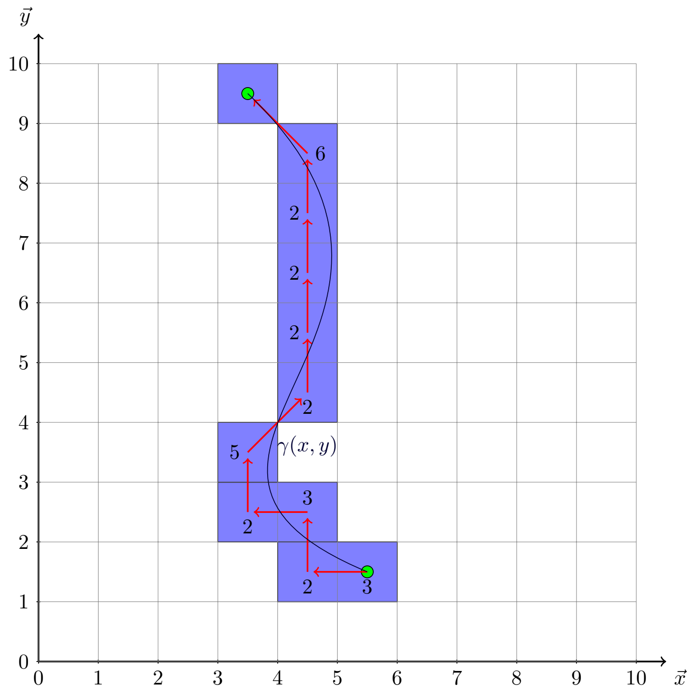


Figure 4.3: This illustration represents a trajectory in the *continuous space* as the blue line γ. That path γ is discretized in the *grid space*, represented by the blue cells. The red arrows represents the change from a cell to the next. The numbers are the associated to the D2Q9 indexes to those moves, also called *k-directions*

Tensor's dimension Those tensors A analyzed in the following paragraphs are conceptually similar to each other, but theirs dimensions are pretty different. Considering the same field with same dimensions it is easy to see that the order of magnitude of the entries' number rapidly increase changing the acquisition method. As example is set a field space of 200×100 cells and a time space of 200 time-steps, to compare all the four model's tensors by theirs intrinsic dimensions.

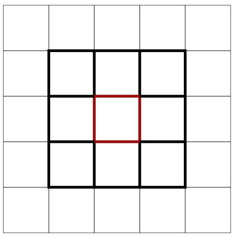


Figure 4.4: Start position.

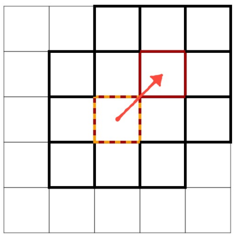


Figure 4.5: First transition Right-Up.

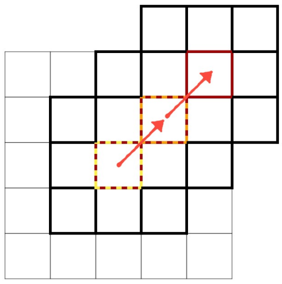


Figure 4.6: Second transition Right-Up.

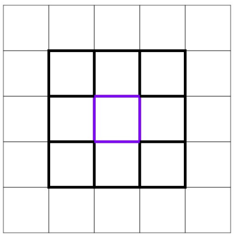


Figure 4.7: Start position.

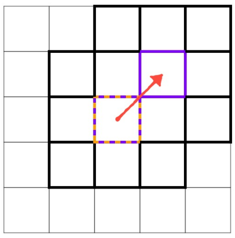


Figure 4.8: First transition Right-Up.

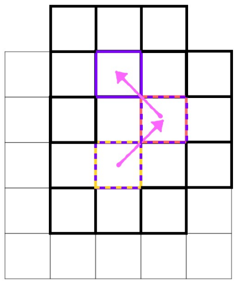


Figure 4.9: Second transition Left-Up.

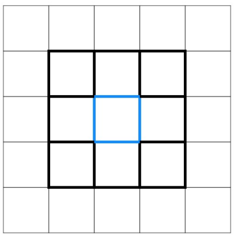


Figure 4.10: Start position.

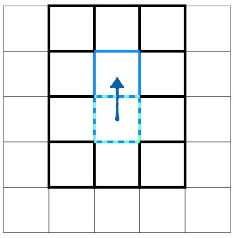


Figure 4.11: First transition Up.

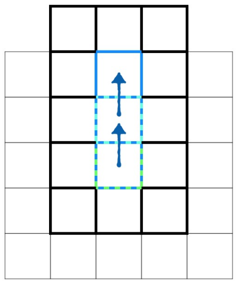


Figure 4.12: Second transition Up.

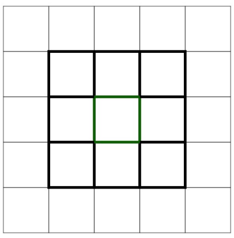


Figure 4.13: Start position.

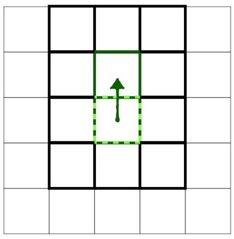


Figure 4.14: First transition Up.

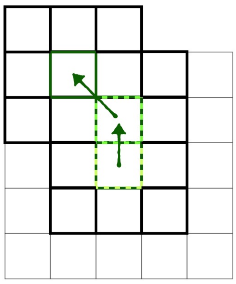


Figure 4.15: Second transition Left-Up.

4.2.1 Model D2Q9

The simplest model considered here is called $D2Q9 - model$. This model is a time-independent and it consider the velocity of the pedestrian. Given a starting position (x_0, y_0) in the field Ω. It uses the nine closest possible positions where a pedestrian could go from that point. With the $D2Q9 - model$ is than possible to know, for each position (x_0, y_0), 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 are defined by the index k. So that the index k gives the direction of the transition. To explicit all the transitions from P_0 to P_k, those transformation are defined in (Equation 4.1) and represented as diagram in (Figure 4.16):

$$\begin{aligned}
P_0 \to P_0 :& \quad (x_0, y_0) \to (x_0, y_0) \\
P_0 \to P_1 :& \quad (x_0, y_0) \to (x_0 + 1, y_0) \\
P_0 \to P_2 :& \quad (x_0, y_0) \to (x_0, y_0 + 1) \\
P_0 \to P_3 :& \quad (x_0, y_0) \to (x_0 - 1, y_0) \\
P_0 \to P_4 :& \quad (x_0, y_0) \to (x_0, y_0 - 1) \\
P_0 \to P_5 :& \quad (x_0, y_0) \to (x_0 + 1, y_0 + 1) \\
P_0 \to P_6 :& \quad (x_0, y_0) \to (x_0 - 1, y_0 + 1) \\
P_0 \to P_7 :& \quad (x_0, y_0) \to (x_0 - 1, y_0 - 1) \\
P_0 \to P_8 :& \quad (x_0, y_0) \to (x_0 + 1, y_0 - 1)
\end{aligned} \tag{4.1}$$

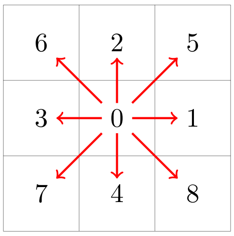


Figure 4.16: The possible transitions, represented as vectors. For the D2Q9 model those vector are also representing the velocity vectors.

Considering the (Figure 4.16) all the transitions are associated to a specific k. This is a particular Markov Chain (Ref.Chap. 3.1) where 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 4.17). To every transition is associated a certain probability to happen. Formally this probability is given by the initial and the final states: p_{if}. Since there always is the same starting state, it is possible to omit it. So that the probability of the transition from P_0 to P_k is expressed by p_k, where the index k points to the ending state. It means that for each position in Ω it's possible to say how likely is to "step forward" or "turn right" and so on. Then, once in the new position, it's again possible to say the most probable direction that the pedestrian will choose. The same prediction is applicable to the whole space, mapped by the real datas.

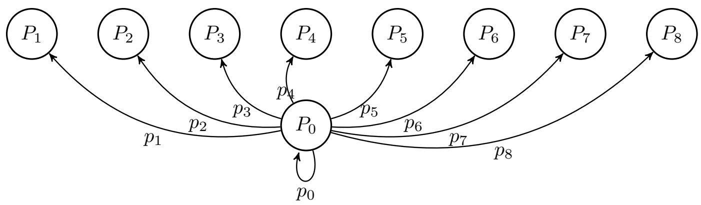


Figure 4.17: The Markov Chain diagram of the system. The states are indicated with circles and labeled with P_k. The transitions are indicated with arrows and labeled with theirs probability p_k

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 4.18). 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 4.1).

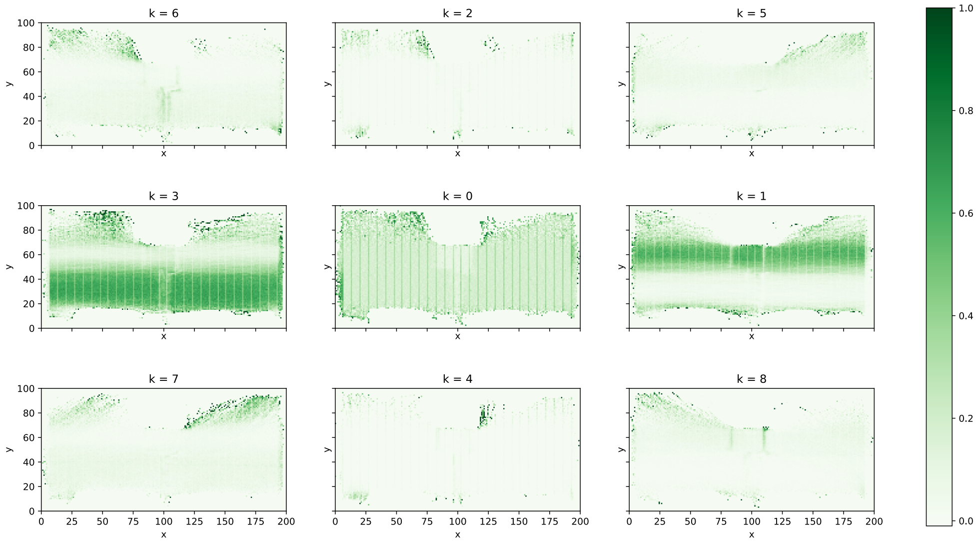


Figure 4.18: Each figure represents the probability in every position to move along a certain direction, defined by the k-index.

4.2.2 Model D2Q9Q9

The conceptual step forward of the study is to consider the next position, but also 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 4.17) in the (Table 4.1).

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

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

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 4.1: This is tabulated the trajectory of the same illustrative pedestrian as above in (Figure 4.17). Here is expressed the position time to time, the index of the following move and the index of the previous move.

Taking into account the example in the (Table 4.1), 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, $A_{3,3,k,h} = 0 \quad for\, k \neq 5,\, h \neq 4$. 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 4.19), 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 4.20) are explicitly expressed the positions and the values of k and h for the two illustrative paths.

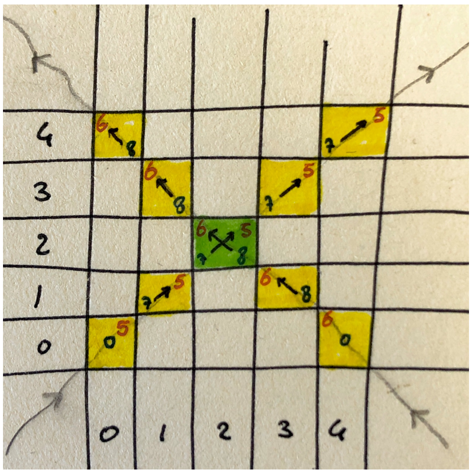


Figure 4.19: (TO DO WITH TIKZ) Two paths passing by the same cell, in green, with different directions. The numbers in red are defining the values of the k-index for every step. The numbers in blue are the values of the h-index for every step.

Cross trajectories A common situation when looking at the trajectories is to find intersections with two different initial directions. Taking 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 4.19). 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 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.

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

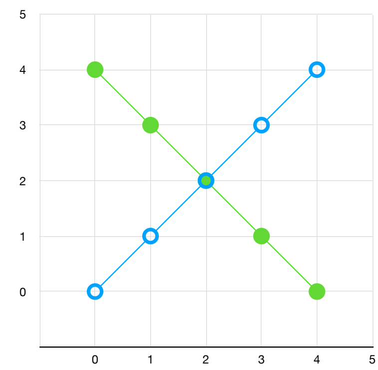


Figure 4.20: (TO DO WITH TABLES) Two paths passing by the same cell, with different directions.

4.2.3 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 4.21). 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.

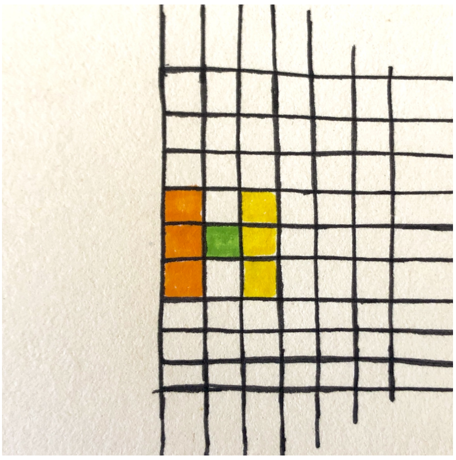


Figure 4.21: (TO DO WITH TIKZ) The boundary position P_b considered is the green cell. When time is *small*, the trajectory is at the beginning, yellow positions are more likely than orange positions. When time is *big*, the trajectory is at the ending, yellow positions are less likely than orange positions.

4.2.4 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. With 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 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 *fill* it all.

4.3 Simulations

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 and opposite directions. To do so, here are considered five pedestrians in (Figure 4.23), with two representations: one is plotting the actual lines in the field (Figure 4.22) and the other is a heat-map that describes where pedestrians passed through (Figure 4.22).

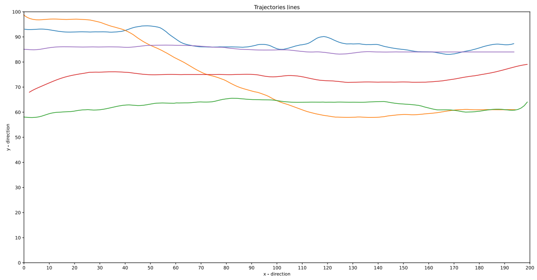


Figure 4.22: Trajectories lines of five ”real” pedestrians.

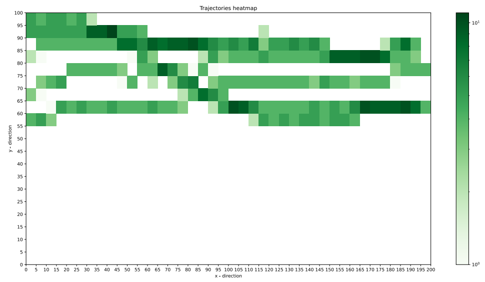


Figure 4.23: Representation of five real trajectories from dataset.

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 4.24). This plot is made as heat-map, that means each cell gives the intensity of that unique combination of velocities.

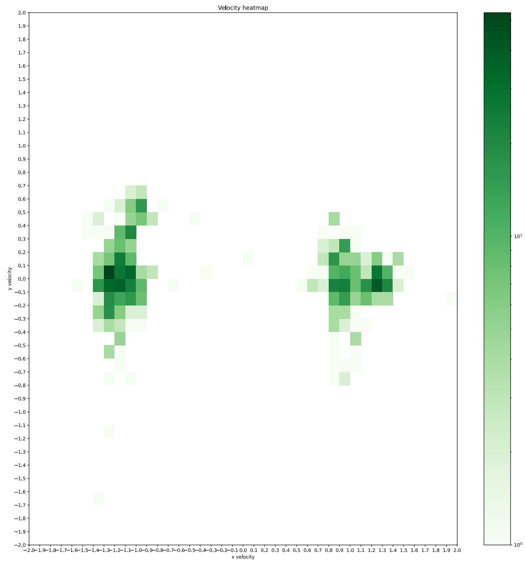


Figure 4.24: Comparison between velocity along the two directions v_x and v_y.

D2Q9 representation Another significant plot is the 3×3 matrix of figures that follows in (Figure 4.25), 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 4.1). 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.

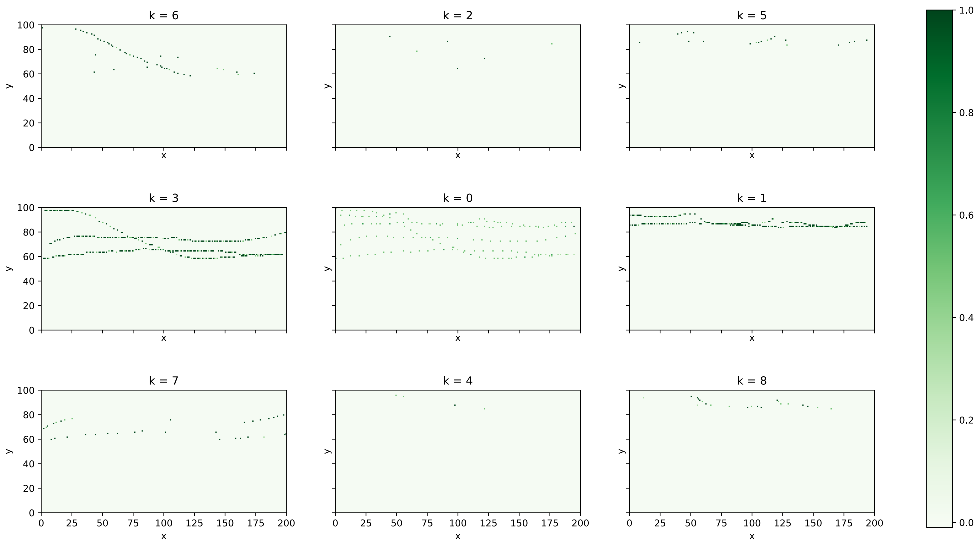


Figure 4.25: Representation of the D2Q9 model. Every plot shows the move probability for each associated direction.

4.3.1 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 to 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 crowd. 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 4.26) 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 4.27) 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 4.28) 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 4.26: 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 4.27: 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 4.28: 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.

4.3.2 Distribution simulation

The analysis on the probability distribution tensor make possible to determinate *which trajectory is more likely to be chosen*. This method propose an approximation of the general problem called *path integral*. 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. 4.2.1). 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 4.2). Lets represent the vector in the form of a matrix, referencing to the (Figure 4.1), to help visualization. And call them: A, B, C, D, with the following values:

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.3 \\ 0 & 0 & 0.7 \end{pmatrix} \quad C = \begin{pmatrix} 0 & 0 & 0.7 \\ 0 & 0 & 0.3 \\ 0 & 0 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.2}$$

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 4.29), 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 4.30) 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 4.31). 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 three path are possible, as showed in (Figure 4.32), but not all with the same probability. Lets set a name 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:

$$\begin{aligned} p_{\gamma_1} &= 1 \times 1 \times 0.3 \times 0.3 \times 1 = 0.09 = 9\% \\ p_{\gamma_2} &= 1 \times 1 \times 0.3 \times 0.7 \times 1 = 0.21 = 21\% \\ p_{\gamma_3} &= 1 \times 1 \times 0.7 \times 1 \times 1 = 0.70 = 70\% \end{aligned} \tag{4.3}$$

The (Equation 4.3) explicitly shows all the possibilities. With this is clear witch one is the more probable in this space.

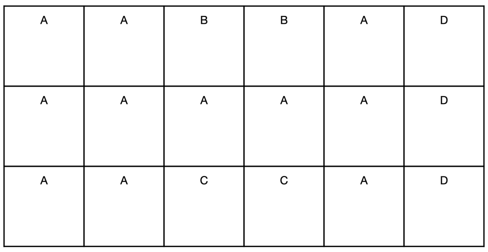


Figure 4.29: Space of the simplistic example. Every letter correspond to a specific distribution of possible transitions, referred to the (Equation 4.2).

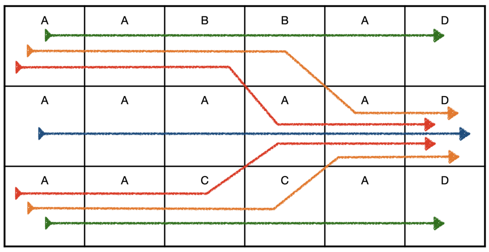


Figure 4.30: All the possible paths that are permitted to travel from the left to the right side of the map Ω_g. Different colors represents different probabilities.

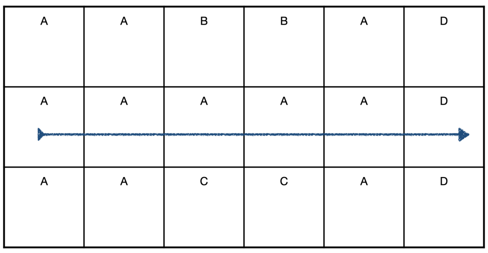


Figure 4.31: A straight path. This path is forced to go straight right because of the distribution A that permits only this movement.

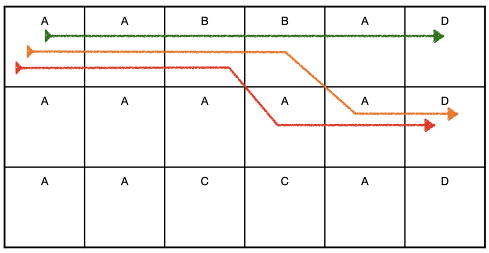


Figure 4.32: The three possible paths when starting from the upper-left cell. This situation is specular to when starting from the bottom-left cell.

Real-life situation The previous example is an extreme simplification of a real case study. In fact it is easy enough to calculate by hand the possibles paths. First of all because the field Ω_g is larger and its dimensions depend only on two factor: the real spaces dimensions (in meters) and the choice of the grid size. The dataset showed before in (Figure 4.22) has an approx. grid space dimensions of 200×100 cells. The (Figure 4.25) express the values of each direction in correlation to the map position. Also this scenario is simplified because it takes into account only five trajectories. In comparison the (Figure 4.18) shows how complex may became the representation when this transitions are calculated for a grater multitude of real pedestrian trajectories.

To solve this problem it's essential the approach using computer's computation.

Chapter 5

Results

5.1 Comparison of assimilation methods

5.1.1 Real data - D2Q9

5.1.2 Real data - D2Q9Q9

5.1.3 Real data - TD2Q9

5.1.4 Real data - TD2Q9Q9

5.2 Simulation with generative 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.

(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;

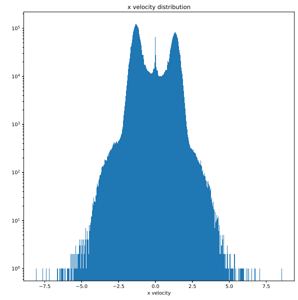
(iii) the correlation between the position along the $\vec{x}$ axes and the magnitude of the velocity vector along the same axes, plotted as heat-map or 2-dimensional histogram;

(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.

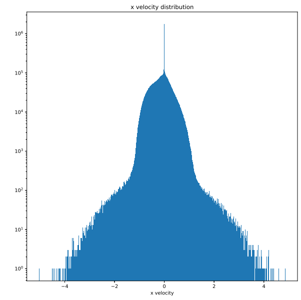
(v) the heat-map of the positions along $\vec{x}$ and $\vec{y}$ axis of all paths that have passed though, plotted as 2-dimensional histogram;

5.2.1 Real data - D2Q9

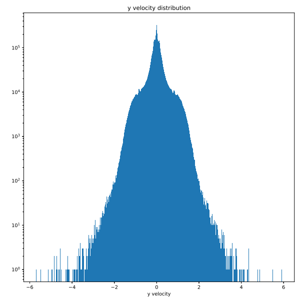
This paragraph's reference are the following: (Figure 5.1), (Figure 5.2), (Figure 5.3).



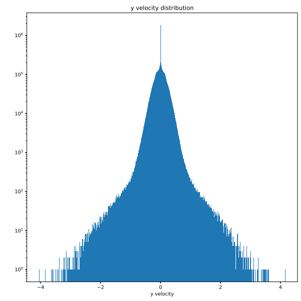
(a) (i) Real data hist vx



(b) (i) Simulation data hist vx

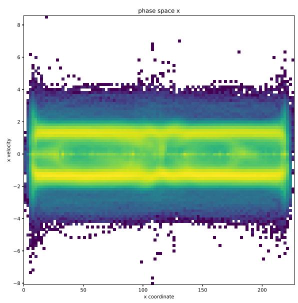


(c) (ii) Real data hist vy

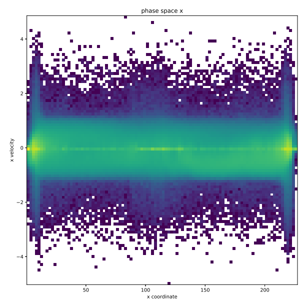


(d) (ii) Simulation data hist vy

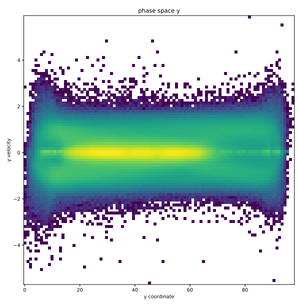
Figure 5.1: (i) & (ii) - simD2Q9 - The magnitude of the velocity vector along the $\vec{x}$ and $\vec{y}$ axis, plotted as 1-dimensional histograms.



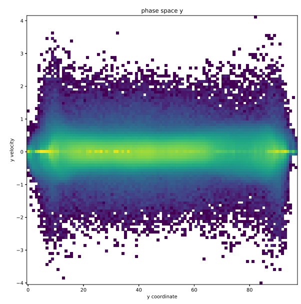
(a) (iii) Real data hist2d xVx



(b) (iii) Simulation data hist2d xVx

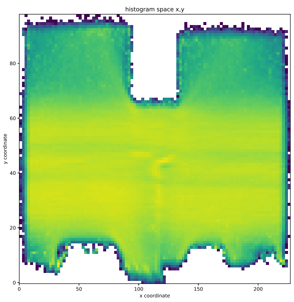


(c) (iv) Real data hist2d yVy

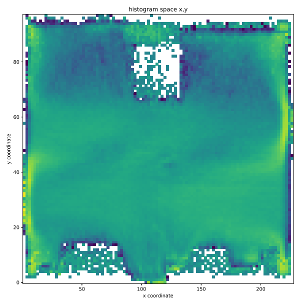


(d) (iv) Simulation data hist2d yVy

Figure 5.2: (iii) & (iv) - simD2Q9 - The correlation between the position along the $\vec{x}$ and $\vec{y}$ axis and the magnitude of the velocity vector along the same correspondent axes, plotted as heat-map or 2-dimensional histograms.



(a) (v) Real data hist2d Pxy

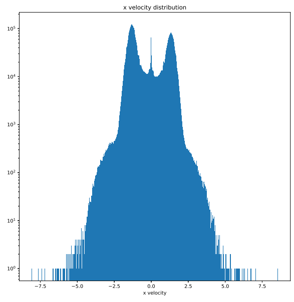


(b) (v) Simulation data hist2d Pxy

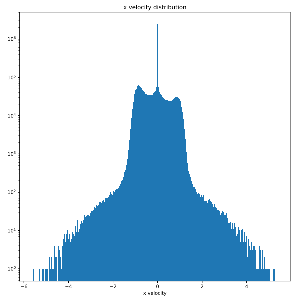
Figure 5.3: (v) - simD2Q9 - The heat-map of the positions along $\vec{x}$ and $\vec{y}$ axis of all paths that have passed though, plotted as 2-dimensional histogram.

5.2.2 Real data - D2Q9Q9

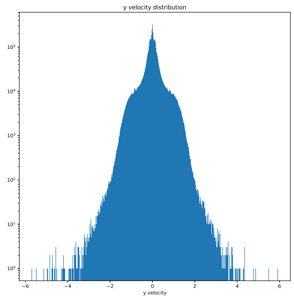
This paragraph's reference are the following: (Figure 5.4), (Figure 5.5), (Figure 5.6)



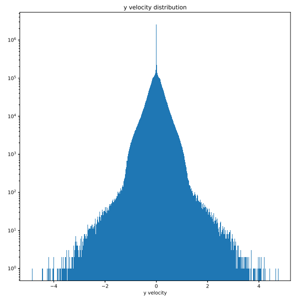
(a) (i) Real data hist vx



(b) (i) Simulation data hist vx

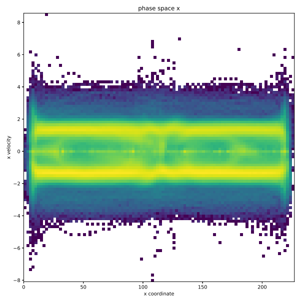


(c) (ii) Real data hist vy

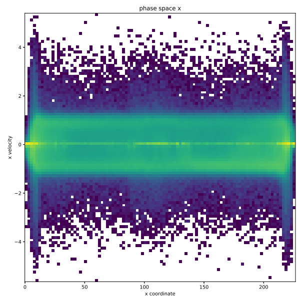


(d) (ii) Simulation data hist vy

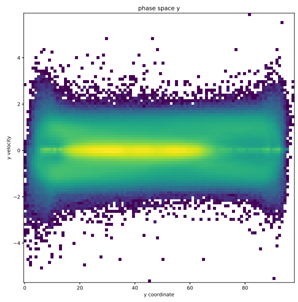
Figure 5.4: (i) & (ii) - simD2Q9Q9 - The magnitude of the velocity vector along the $\vec{x}$ and $\vec{y}$ axis, plotted as 1-dimensional histograms.



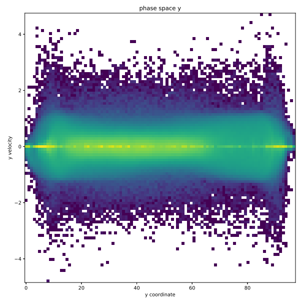
(a) (iii) Real data hist2d xVx



(b) (iii) Simulation data hist2d xVx

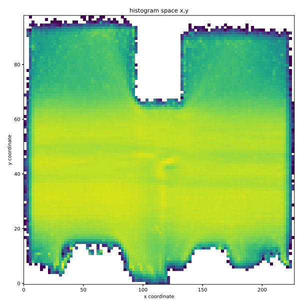


(c) (iv) Real data hist2d yVy

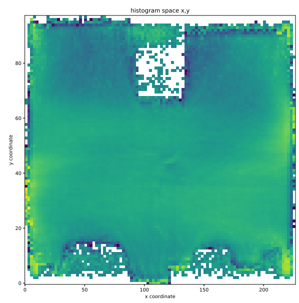


(d) (iv) Simulation data hist2d yVy

Figure 5.5: (iii) & (iv) - simD2Q9Q9 - The correlation between the position along the $\vec{x}$ and $\vec{y}$ axis and the magnitude of the velocity vector along the same correspondent axes, plotted as heat-map or 2-dimensional histograms.



(a) (v) Real data hist2d Pxy

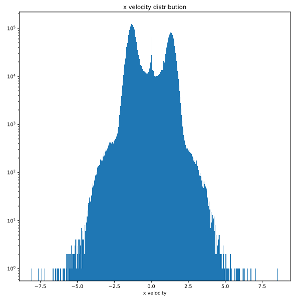


(b) (v) Simulation data hist2d Pxy

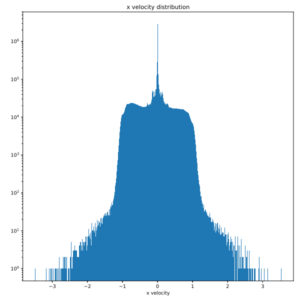
Figure 5.6: (v) - simD2Q9Q9 - The heat-map of the positions along $\vec{x}$ and $\vec{y}$ axis of all paths that have passed though, plotted as 2-dimensional histogram.

5.2.3 Real data - TD2Q9

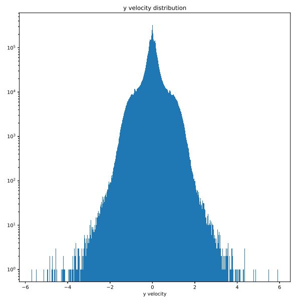
This paragraph's reference are the following: (Figure 5.7), (Figure 5.8), (Figure 5.9)



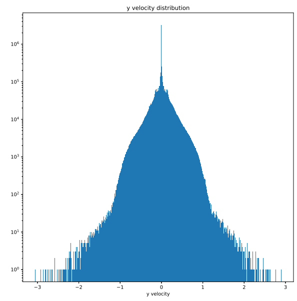
(a) (i) Real data hist vx



(b) (i) Simulation data hist vx

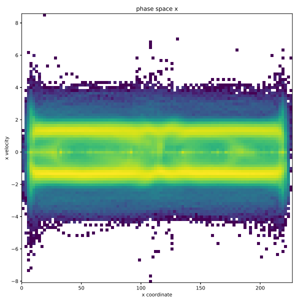


(c) (ii) Real data hist vy

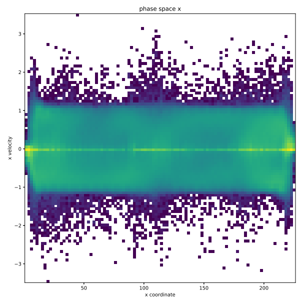


(d) (ii) Simulation data hist vy

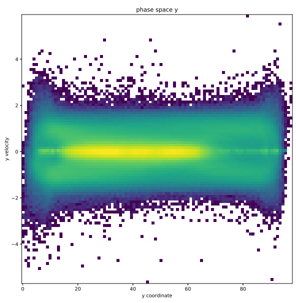
Figure 5.7: (i) & (ii) - simTD2Q9 - The magnitude of the velocity vector along the $\vec{x}$ and $\vec{y}$ axis, plotted as 1-dimensional histograms.



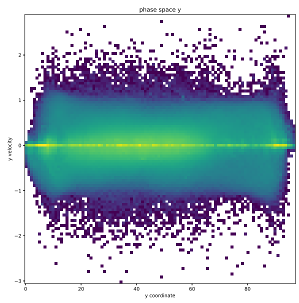
(a) (iii) Real data hist2d xVx



(b) (iii) Simulation data hist2d xVx

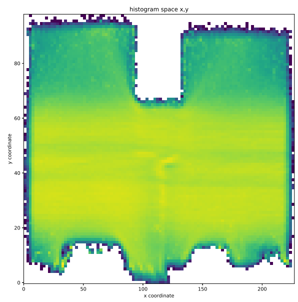


(c) (iv) Real data hist2d yVy

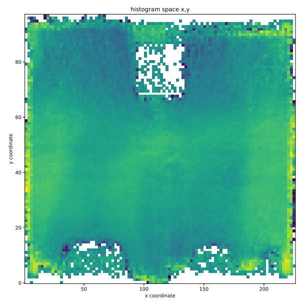


(d) (iv) Simulation data hist2d yVy

Figure 5.8: (iii) & (iv) - simTD2Q9 - The correlation between the position along the $\vec{x}$ and $\vec{y}$ axis and the magnitude of the velocity vector along the same correspondent axes, plotted as heat-map or 2-dimensional histograms.



(a) (v) Real data hist2d Pxy

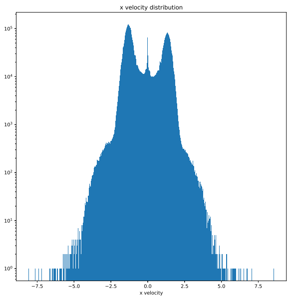


(b) (v) Simulation data hist2d Pxy

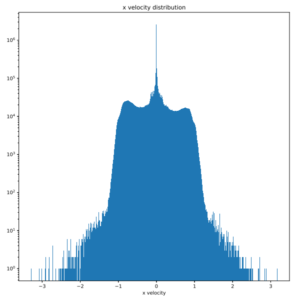
Figure 5.9: (v) - simTD2Q9 - The heat-map of the positions along $\vec{x}$ and $\vec{y}$ axis of all paths that have passed though, plotted as 2-dimensional histogram.

5.2.4 Real data - TD2Q9Q9

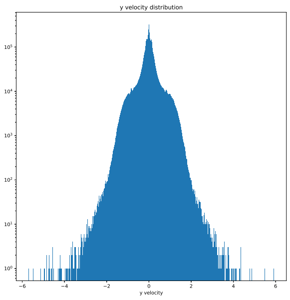
This paragraph's reference are the following: (Figure 5.10), (Figure 5.11), (Figure 5.12)



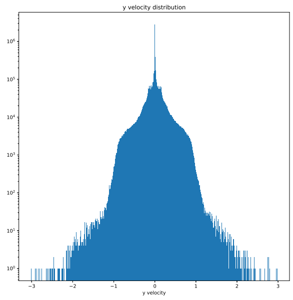
(a) (i) Real data hist vx



(b) (i) Simulation data hist vx

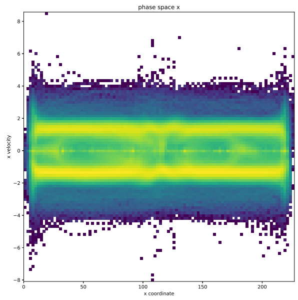


(c) (ii) Real data hist vy

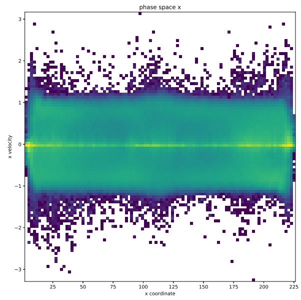


(d) (ii) Simulation data hist vy

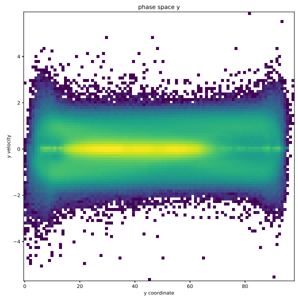
Figure 5.10: (i) & (ii) - simTD2Q9Q9 - The magnitude of the velocity vector along the $\vec{x}$ and $\vec{y}$ axis, plotted as 1-dimensional histograms.



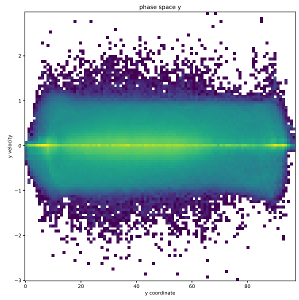
(a) (iii) Real data hist2d xVx



(b) (iii) Simulation data hist2d xVx

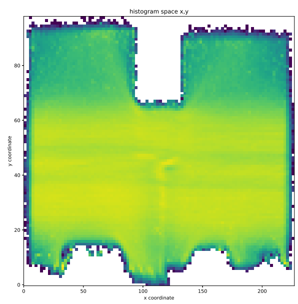


(c) (iv) Real data hist2d yVy

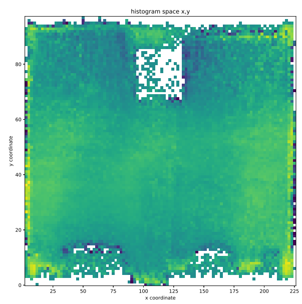


(d) (iv) Simulation data hist2d yVy

Figure 5.11: (iii) & (iv) - simTD2Q9Q9 - The correlation between the position along the $\vec{x}$ and $\vec{y}$ axis and the magnitude of the velocity vector along the same correspondent axes, plotted as heat-map or 2-dimensional histograms.



(a) (v) Real data hist2d Pxy



(b) (v) Simulation data hist2d Pxy

Figure 5.12: (v) - simTD2Q9Q9 - 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.

Chapter 6

Discussion

Discussion documentation here

Chapter 7

Appendix

7.1 Package pathintegralanalytics - code description

Package pathIntegralAnalytics documentation here

7.1.1 Explanation of the library

Explanation of the library documentation here

7.1.2 UML diagram

UML pathIntegralAnalytics documentation code here