

# **Lanes Discovery using Hausdorff distance and Spectral Clustering**

Sistemi Complessi: Modelli e Simulazione

**Cezar Angelo Sas - 781563**

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>                        | <b>2</b>  |
| <b>2</b> | <b>State of the Art</b>                    | <b>3</b>  |
| 2.1      | Pedestrian Self-Organization . . . . .     | 3         |
| 2.2      | Trajectories Clustering . . . . .          | 5         |
| 2.2.1    | Distance measures . . . . .                | 5         |
| 2.2.2    | Clustering Algorithms . . . . .            | 6         |
| 2.3      | Lanes Discovery using Clustering . . . . . | 8         |
| <b>3</b> | <b>Proposed Solutions</b>                  | <b>10</b> |
| 3.1      | Hausdorff Distance . . . . .               | 10        |
| 3.2      | Spectral Clustering . . . . .              | 11        |
| 3.3      | Lane Discovery . . . . .                   | 12        |
| <b>4</b> | <b>Experiments</b>                         | <b>13</b> |
| 4.1      | Datasets . . . . .                         | 13        |
| 4.1.1    | Gorrini et al. . . . .                     | 13        |
| 4.1.2    | Zhang et al. . . . .                       | 14        |
| 4.2      | Performance Measures . . . . .             | 16        |
| 4.3      | Results . . . . .                          | 18        |
| 4.3.1    | Gorrini et al. . . . .                     | 18        |
| 4.3.2    | Zhang et al. . . . .                       | 22        |
| <b>5</b> | <b>Conclusions</b>                         | <b>24</b> |

# Chapter 1

## Introduction

Pedestrian flow analysis aims to provide insight into human movement patterns in buildings or outdoor areas. These analyses provide valuable information to building designers and other decision makers. A substantial body of research is based on pedestrian behavior and the interaction of pedestrians with their environment and other pedestrians.

In order to help this analysis we propose a pedestrian trajectories clustering system for lane discovery. Our solution uses the work of Atev et al. [1] on clustering of vehicular trajectories applied to our domain. We evaluate our solution on multiple dataset.

In the next chapter we will take a look at the state-of-the-art of pedestrian self-organization, trajectories clustering, and lane discovery. In the third chapter we present our solution, while in the fourth we discuss the performance of our system. Finally, in the last chapter we present our conclusion and present some future works.

# Chapter 2

## State of the Art

In this section we will introduce some state-of-the-art research regarding self-organization in pedestrian flow and also for trajectories clustering. In the last section of this chapter we will present some solutions for the lane discovery problem.

### 2.1 Pedestrian Self-Organization

Designing of walking infrastructure requires the understanding of pedestrian's behavior and their interaction with the environment. A good understanding of patterns and different behaviors under different circumstances allows to blueprint safer buildings, public areas, and cities.

In [2] S. Hoogendoorn and W. Daamen performed an experimental research analyzing the problem of pedestrian self-organization. Their theory is based on the psychological *principle of the least effort*. According to this principle each pedestrian aims to minimize his/hers predicted disutility of walking (i.e. the pedestrian economicus [3]). This means that a pedestrian will always try to minimize the disutility of his actions (e.g. accelerating, decelerating, change direction, do nothing).

Moreover, under specific condition, differential game theory predicts the occurrence of the Nash equilibrium solution to the n-person non-cooperative differential game [4]. This means no pedestrian can improve his/hers condition by unilateral undertaking an action. This equilibrium state could be either stable or meta-stable. The evolution of the state is achieved via self-formation of homogeneous patters like dynamic lane formations.

In their work Hoogendoorn and Daamen [2] observed the formation of spatial patterns. The standard patterns observed are shown in Figure 2.1. The first pattern (Figure 2.1a) shows the case in which distances between pedestrian are approximately constant. The second and third pattern (Fig-

ure 2.1b and 2.1c) reflect cases where the lead gap for the pedestrians in one layer is relatively small, while the lag gap is relatively large (or vice-versa). The final pattern (Figure 2.1d) is a special one. It can be described as a missing pedestrian inside a lattice of other pedestrians. This vacancy position is caused by inefficient merging behavior at the bottleneck entry.

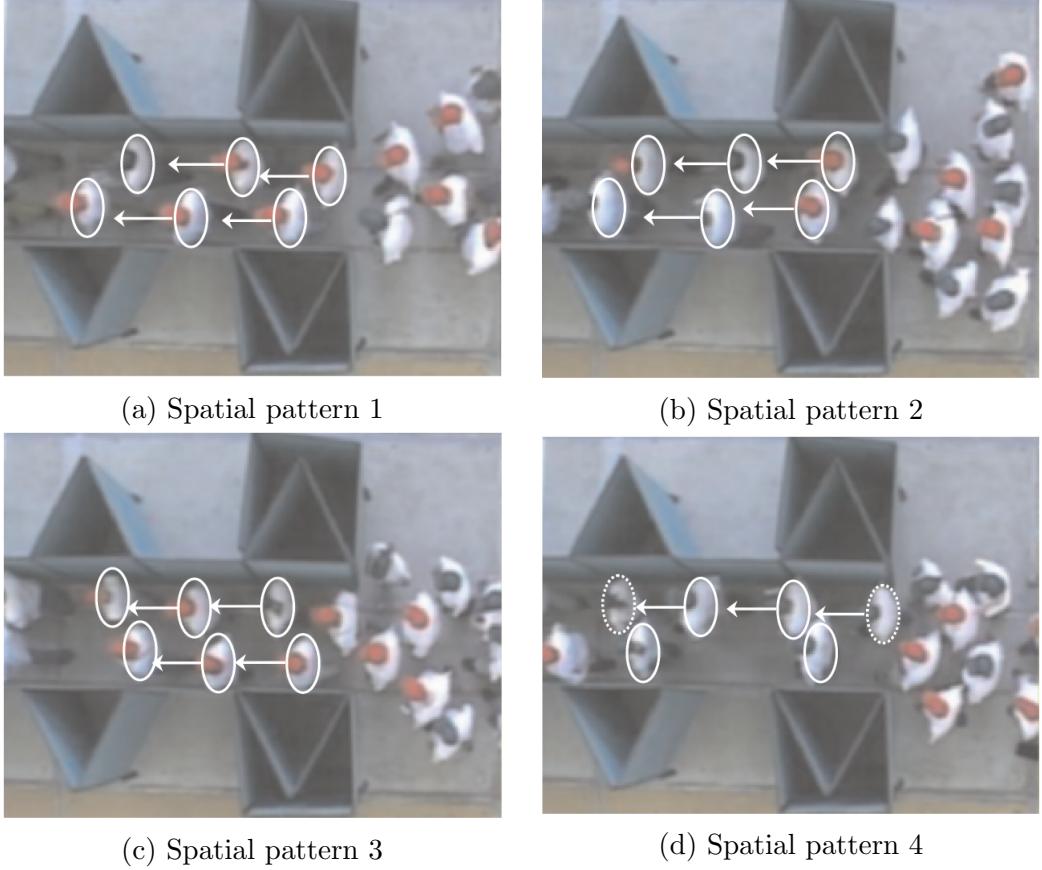


Figure 2.1: Spatial patterns according

Hoogendoorn et al. [2] also performed analysis of lane formation in bi-directional pedestrian flows. They observed that the patterns formed in uni-directional flow are formed in both walking direction in a bi-directional flow (Figure 2.2). In [5] was proposed that the number of formed lanes depends on the density of the walking area. In Section 2.3 we will describe their solution for the lane discovery problem.

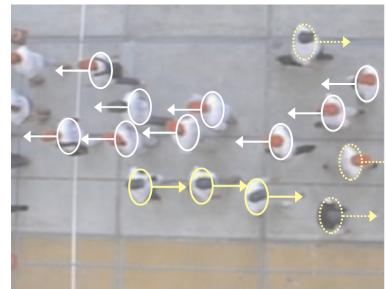


Figure 2.2: Example of standard patterns in bi-directional flow

## 2.2 Trajectories Clustering

Now we will take a look at the current state-of-the-art solution in trajectory clustering. First we will take a look at the commonly used distances and next we will present the clustering algorithms as described in the work of Yuan et al. [6]. Before we start, we need to introduce the notion of trajectory.

**Trajectory:** A trajectory  $TR$  is a time ordered sequence consisting of multi-dimensional points in space. Given  $N$  sampled points  $P$ , a trajectory is denoted by  $TR = \{P_1, P_2, \dots, P_N\}$  where  $P_j$  is a multi-dimensional point sampled at time  $j$ .

### 2.2.1 Distance measures

Distance measure is one of the most important part in a clustering algorithm: it allows us to distinguish between different objects and find similar ones. Different from traditional data, which are static, single and independent, trajectories are a serial of sampling points, and usually are not the same length. Therefore, the comparison between two trajectories should follow special strategies, which comprehensively compare their differences.

**Euclidean distance:** Is the simplest distance measure, but is also very powerful as it has a complexity of  $O(n)$  which means it can be used on large dataset. The main problem with euclidean distance is the low noise tolerance, moreover it also requires that the point are sampled at the same time. Let  $L_i$  and  $L_j$  be p-dimensional trajectories segment with length  $N$ . Their euclidean distance  $D_E$  is defined as follow:

$$D_E(L_i, L_j) = \frac{1}{N} \sum_{k=1}^N \sqrt{\sum_{m=1}^p (a_k^m - b_k^m)^2} \quad (2.1)$$

**Longest Common Sub-Sequence (LCSS):** Let  $L_i$  and  $L_j$  be two dimensional trajectories with length  $n$  and  $m$  respectively (Note that the restriction of two dimension is for notation simplicity, LCSS can be applied also to multi-dimensional trajectories). The longest common sub-sequence distance is recursively defined as:

$$D_L(L_i, L_j) = \begin{cases} 0 & n = m = 0 \\ 1 + LCSS_{\sigma, \epsilon}(Head(L_i), Head(L_j)) & |a_i^x - b_j^x| \leq \sigma \wedge |a_i^y - b_j^y| \leq \epsilon \\ \max(LCSS_{\sigma, \epsilon}(Head(L_i), L_j), LCSS_{\sigma, \epsilon}(L_i, Head(L_j))) & \text{other} \end{cases} \quad (2.2)$$

Where  $\sigma$  and  $\epsilon$  are the threshold of x-direction and y-direction respectively. When abscissa difference and ordinate difference between two trajectories is respectively less than  $\sigma$  and  $\epsilon$ , the pair of points are considered similar and the value of LCSS is increased by one.

LCSS allows certain deviation existing in sampling data. Therefore, the LCSS is effective and efficient in practical application. However, LCSS is over-reliance on two user parameters  $\sigma$  and  $\epsilon$ , so how to determine the optimal parameters is a difficult problem. The complexity of LCSS computation is  $O(m * n)$

Other measures include Dynamic Time Warping, a technique used to find the optimal alignment between two given (time-dependent) sequences under certain restrictions, and Fréchet distance which is a measure of similarity between curves that takes into account the location and ordering of the points along the curves.

### 2.2.2 Clustering Algorithms

Trajectory clustering aims at finding out trajectories that are of the same (or similar) pattern, or distinguishing some undesired behaviors (such as outliers). In 2001, Miller et al [7] classified clustering algorithms for handling static data into four categories: partition based method, hierarchy based method, density based method, grid based method, and model based method (this last one was introduced by Han et al. [8] in 2011). We will not introduce all methods but only present a few example for some of these categories. For more algorithms, see [7, 8].

**SStatistical INformation Grid (STING):** is a grid-based multiresolution data structure in which the spatial area is divided into cells. The layers are linked using a hierarchical structure of cells. We can see in Figure 2.3 an example of levels.

Cells contain pre-computed information such as count, mean, distribution, max, and min values. These measures are loaded from the bottom-most level. In order to perform clustering of this structure STING uses a top-down, grid-based method to find regions with a sufficient density (set by the user). The algorithm starts by computing a confidence interval for each cell in the layer. Using this value, the algorithm determine the relevant cells. The irrelevant cells are then refined to a finer resolution by repeating the procedure at the next level. This process is repeated until bottom is reached. If a relevant cell met the query specification it is returned.

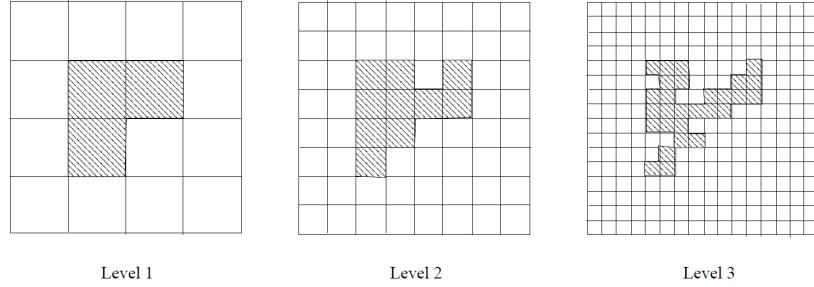


Figure 2.3: Example of consecutive layers

### Density-Based Spatial Clustering of Applications with Noise

**(DBSCAN):** is a density-based clustering algorithm developed by Ester et al. [9]. Given a set of points it groups together points that are closely packed together, and marking as outliers points in low density areas. We can see an example of how DBSCAN works in Figure 2.4. The algorithm requires two input parameters  $\epsilon$ , used to define the neighborhood of radius  $\epsilon$  of an object: the  $\epsilon - neighborhood$ . The other is *MinPoints*, an object with at least *MinPoints* of object in its  $\epsilon - neighborhood$  is called *core object*. In order to work correctly DBSCAN follows these rules:

- An object can only belong to a cluster if and only if it lies within the  $\epsilon - neighborhood$  of a core object in the cluster.
- A core object  $o$  within the  $\epsilon - neighborhood$  of another core object  $p$  must belong to the same cluster as  $p$ .
- A non-core object  $q$  within the  $\epsilon - neighborhood$  of some core object  $p_1, \dots, p_i$ ,  $i > 0$ , must belong to the same cluster to at least one of the core objects from  $p_1, \dots, p_i$ .
- A non-core object  $r$  which does not lie within the  $\epsilon - neighborhood$  of any core object is considered to be noise.

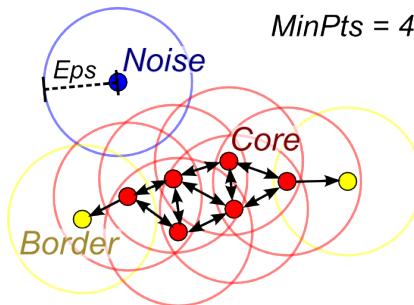


Figure 2.4: Example of DBSCAN algorithm

## 2.3 Lanes Discovery using Clustering

The study of pedestrian behavior and trajectories clustering bring us to the objective of this report: automatically finding lanes in pedestrian trajectories. There are few solution for this specific problem, we will present two of them. The first is the one proposed by Hoogendoorn et al. [2], the second is a variation of this solution using a modified version of DBSCAN and was proposed by Crociani et al.. Both of these solution perform clustering analysis on few frames, as we will see our solution takes all trajectory and then performs cluster analysis.

**Hoogendoorn et al.** They define a cluster using the location  $\vec{r}_i(t)$  and velocities  $\vec{v}_i(t)$  and the following criteria, we can consider this a simplified version of DBSCAN:

- The distance  $\|\vec{r}_i(t) - \vec{r}_j(t)\|$  between  $i$  and  $j$  is smaller than a value  $c_1$ .
- The velocity difference  $\|\vec{v}_i(t) - \vec{v}_j(t)\|$  is less than some value  $c_2$ .

This definition gives us a cluster, but not a lane, as a lane can be formed by many cluster, and cluster can't belong to more than a lane.

We can see some results of this solution in Figure 2.5, note that Hoogenendoorn et al. don't specify how the union of cluster was made in order to obtain a lane.

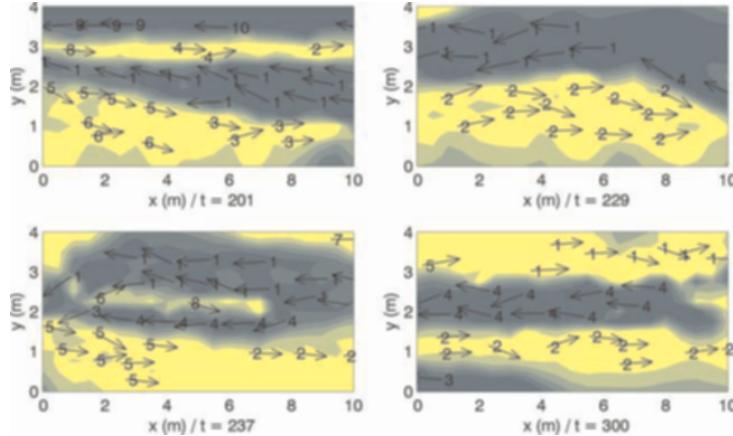


Figure 2.5: Example results of cluster analysis for four time slices; numbers indicate pedestrian clusters. The color indicates the average direction at a particular location

**Crociani et al.** Propose a solution that modifies DBSCAN to use the information of direction. The proposed solution perform two iteration of DBSCAN, the first time is executed using direction extracted from velocities of a few samples. The second run is performed using clustered information from the first iteration and it identifies lanes. The model requires four parameters:

- $\Theta_v$ : Angle threshold, used to define the direction of a pedestrian
- $\Theta_x$ : Defines the length of the semi-minor axis
- $\Xi_x$ : Multiplier for defining the length of the semi-major axis using the semi-minor one.
- *MinPoints*: Same as for DBSCAN

In the first step of clustering, the algorithm takes the velocities  $\vec{v}_i$  of pedestrians, and as parameters  $\Theta_v$ , and *MinPoints*. Using these values the distance measure between two velocity vectors  $\vec{v}_i, \vec{v}_j$  is performed using the cosine dissimilarity (Equation 2.3). This step returns a set of clusters  $\Phi = F_1, \dots, F_m$ .

$$a_{ij} = \arccos \left( \frac{\vec{v}_i * \vec{v}_j}{\|\vec{v}_i\| * \|\vec{v}_j\|} \right) \quad (2.3)$$

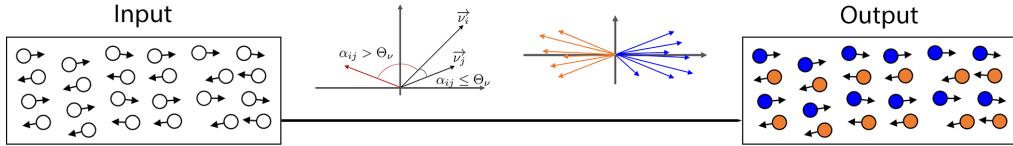


Figure 2.6: Step one. DBSCAN performed using direction

The second step, takes as input clusters from  $\Phi$  one at a time. The parameters for this step are  $Theta_x$ ,  $\Xi_x$ , and *MinPoints*. Lanes are identified using an elliptical distance. The ellipse is centered on the point and the angle is the mean of the neighbors, in order to describe more precisely the idea of pedestrian following other pedestrians. After the DBSCAN algorithm is executed, the results are cluster describing lanes.

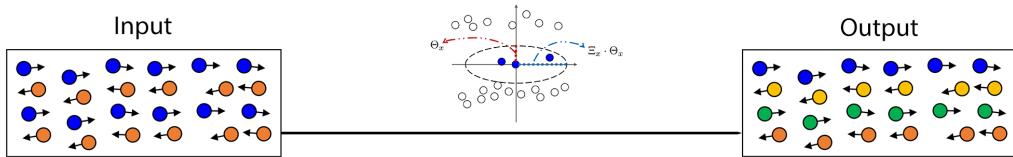


Figure 2.7: Step two. DBSCAN for lanes recognition

# Chapter 3

## Proposed Solutions

We introduce our solution. It is based on the work of Atav et al. [1] that performs spectral clustering analysis on vehicular trajectories.

We will first describe the distance measure, next we will present the clustering algorithm and finally in the next chapter we present the result.

### 3.1 Hausdorff Distance

Introduced by Felix Hausdorff [10] in 1914, is commonly used in computer vision. The Hausdorff distance is a metric between two subsets in a metric space. Given two non empty set  $P$  and  $Q$ , the Hausdorff distance  $D_H(P, Q)$  is given by:

$$D_H(P, Q) = \max \left\{ \sup_{p \in P} \inf_{q \in Q} d(p, q), \sup_{q \in Q} \inf_{p \in P} d(p, q) \right\} \quad (3.1)$$

The main problem with Hausdorff distance is that is very sensitive to outlier points. In addition it does not consider the order of the points, which is an important factor for trajectories. In order to address these two issues, Atev and his team modified the original Hausdorff distance like this:

$$h_{\alpha, N, C}(P, Q) = \operatorname{ord}_{p \in P}^{\alpha} \left\{ \min_{q \in N_Q(C_{P, Q}(p))} d(q, p) \right\} \quad (3.2)$$

Where:

- $C_{P, Q} : P \mapsto Q$  is a function that assigns to each point  $p \in P$  a corresponding point  $q \in Q$ ,
- $N_Q : Q \mapsto \mathbb{P}(Q)$  is a function that specifies a subset of  $Q$  as the neighborhood of the point  $q \in Q$ ,

- $\text{ord}_{s \in S}^{\alpha} f(s)$  denotes the value among the image  $f(S)$  of the set  $S$  that is greater than  $\alpha |f(S)|$  of all values. Intuitively this function can be viewed as  $\max_{s \in S} f(s)$  for  $\alpha = 1$ ,  $\min_{s \in S} f(s)$  for  $\alpha = 0$  and for  $\alpha = \frac{1}{2}$  as median $_{s \in S} f(s)$ . The parameter  $\alpha$  defines the sensitivity to noise, but it also brings to some issues with triangular inequality.

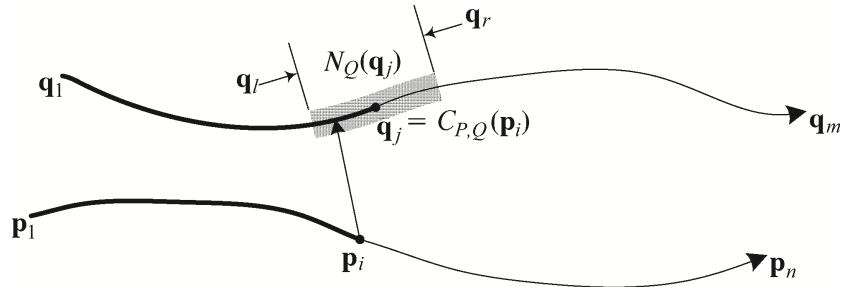


Figure 3.1: Atev's edited Hausdorff distance

In Figure 3.1 we can have a look at how each part act on the trajectory. For more info about these function see [1].

## 3.2 Spectral Clustering

Spectral clustering is a technique that uses dimensionality reduction methods based on eigenvalues of the affinity matrix to perform clustering in fewer dimensions.

The algorithm used by Atev is a variant of the one proposed by Ng et al. [11] with the scaling described in the work of Zelnik et al. [12].

The first step in order to perform the spectral clustering is the evaluation of the affinity matrix  $K$ .

$$k_{i,j} = \exp\left(-\frac{h_{a,N,C}(P_i, Q_i)h_{a,N,C}(Q_i, P_i)}{2\sigma(P_i)\sigma(Q_i)}\right) \quad (3.3)$$

We also need the diagonal matrix  $W$  defined as:

$$w_{i,i} = \sum_{1 < j < n} k_{i,j} \quad (3.4)$$

Finally we compute the normalized affinity matrix  $L$  as:

$$L = W^{\frac{1}{2}} K V^{-\frac{1}{2}} \quad (3.5)$$

We can now compute the eigenvalues and eigenvectors of  $L$ . In an ideal case counting the eigenvalues  $g$  that are  $= 1$  will give us the total numbers of clusters. Unfortunately as described in [12] this is not feasible, however we can find a range using the spectrum of  $L$ . Given his eigenvalues  $1 \geq \lambda_1 \geq \dots \geq \lambda_n \geq 0$  we can count all the values that are greater than 0.99 in order to find the minimum number of clusters  $g_{min}$ , and in order to find the maximum number of clusters  $g_{max}$  we count all values greater than 0.88. These values were chosen heuristically. The eigenvectors of  $L$  are sorted based on the eigenvalues and then using the latter ones we search the optimal number of clusters  $nc$  using K-Means iterated for all the possible values and then we perform clustering metrics in order to choose the best one.

Finally a K-Means clustering is performed on the trajectories using the first  $nc$  eigenvectors and the estimated value of  $k = nc$ .

### 3.3 Lane Discovery

Using the methods described before, we can now perform the lane discovery task. In our case we start from the trajectories data, all of them have the same start time. So we ignore the difference in time between trajectories that are sampled in different moments, this implies that our system is more suitable for static analysis rather than online methods like the solution proposed by Crociani et al. which uses only a few frames in order to assign a pedestrian to a lane. We use the trajectories as input for the spectral clustering algorithm that returns the trajectories with the corresponding id. We consider this id as the corresponding lane for the pedestrian. This means that a trajectory can only belong to one cluster, and a cluster represents only one lane.

# Chapter 4

## Experiments

In this chapter we take an overview of the experiments that we used to evaluate our system. We analyze the results, and present some problems of our approach.

### 4.1 Datasets

The system was evaluated using data from Gorrini et al. [13] and Zhang et al. [14] experiments. These datasets contain trajectories of walking pedestrian.

#### 4.1.1 Gorrini et al.

This data-set contains trajectories form an experiment of 2015 at the University of Tokyo. The experiment was designed as a corridor like scenario delimited with band partitions. The area of the experiment was divided in 3 parts:

- The center area with a size of 10 m x 3 m. The measurements were performed in this zone;
- Two buffer zones, of 2 m x 3 m, that allowed pedestrians to reach a stable speed;
- The two starting areas, with a size of 12 m x 3 m.

The sample of pedestrian was composed of 54 participants (male students from 18 to 25 years old). At the start signal, all participants were asked to walk to the opposite side of the corridor creating an bi-directional flow.

The dataset contains four different starting configuration based on lanes division (see figure 4.2). The first configuration is (3, 3), it consist in a separation in 3 lanes for each starting area. Next we have the (4, 2) configuration,



Figure 4.1: Frame extracted from an experiment of Gorrini et al. [13]

this time the separation is done in 4 lanes on one side and 2 lanes on the opposite side, and the same for the last configuration (5, 1). The data-set also contains an experiment of a single flow with configuration (6, 0). Each of these experiments were repeated 4 times asking the participants to change their starting position.

For our requirements we ignored configurations (5, 1) and (6, 0) because they will not give us much information.

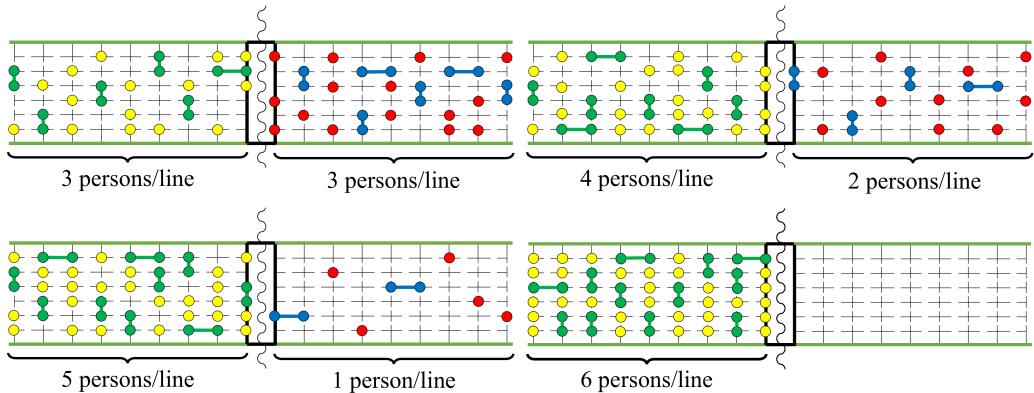


Figure 4.2: Lanes configurations. Colors represent directions and if the pedestrians are in group of two. Image from Gorrini et al. [13]

The trajectories of this dataset were annotated with an ID by a human, in order to evaluate the performance of our approach.

#### 4.1.2 Zhang et al.

The experiment performed by Zheng and his team [14] also consists in a corridor like scenario. In this case the participants were 350 German students with an average age of  $25 \pm 5.7$  years.

We can see in figure 4.3 the area used for the experiments. We can notice the two waiting area, one on each side. Each waiting area has an entrance

with a width that can be adjusted to regulate the flow ( $b_l$  for the left entrance and  $b_r$  for the right one). After the participants pass the entrance they have a buffer area of 4 m before entering the corridor which has a length of 8 m and a width of 3 m or 3.6 m depending on the experiment configuration.

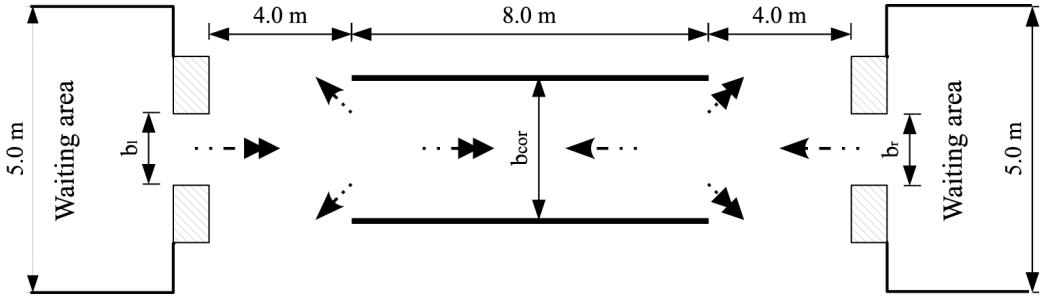


Figure 4.3: Experiment setup. Image form Zhang et al. [14]

In this experiments were analyzed different types of flow. To achieve this, at each run, the participants received instruction on the behavior they had to keep. The resulting flows described by Zheng et al. [14] are the following:

- Balanced Flow Ratio - Stable Separated Lanes flow (BFR-SSL): These trajectories are from the runs for  $b_l = b_r$  but without instructions. The opposing flows segregate and occupy proportional shares of the corridor. Stable lanes and interface formed autonomously. The gap between the opposing streams is larger for the situation with low crowd density in the corridor;
- Balanced Flow Ratio - Dynamical Multi-Lanes flow (BFR-DML): They are obtained from runs of experiment for  $b_l = b_r$  with instruction. With this initial condition again lane formation is observable, but the lanes are unstable and vary in time and space. This type of flow is comparable with two streams crossing at a small angle. In this type of flow, the location of different lanes seems more stable at high density conditions;
- Unbalanced Flow Ratio - Dynamical Multi-Lanes flow (UFR-DML): These trajectories are obtained from runs of experiment for  $b_l \neq b_r$  with instructions. Again lanes are unstable and vary in time and space. The accumulated trajectories indicate that the flow ratio of the opposing streams is unbalanced.

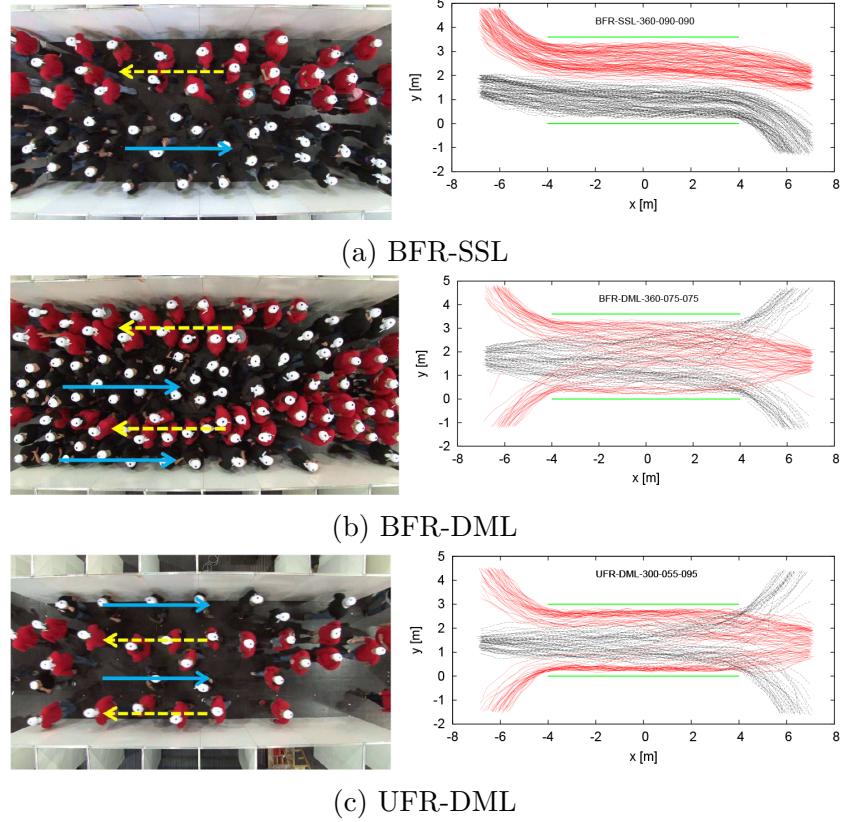


Figure 4.4: Different types of flows

For this data-set the trajectories were not labeled, we will only give a qualitative analysis of the results.

## 4.2 Performance Measures

For  $N$  trajectories, given a vector  $X = (x_1, \dots, x_N)$  of labels computed by the system and a vector  $Y = (y_1, \dots, y_N)$  of labels from the ground truth, by computing the confusion matrix we can evaluate the performance of our system.

- True Positive ( $tp_i$ ): the number of correctly recognized observations for trajectory  $T_i$
- False Positive ( $fp_i$ ): the number of correctly recognized observations that do not belong to the trajectory  $T_i$ .
- True Negative ( $tn_i$ ): the number of observations that were incorrectly assigned to the trajectory  $T_i$ .

- False Negative ( $fn_i$ ): the number of observations that were not recognized as belonging to the trajectory  $T_i$ .

Using these definitions we can calculate the micro performance scores of the system as defined by Sokolova et al. [15]:

- Micro Precision: represents the number of correctly labeled trajectories over the total number of trajectories. This value is defined in the interval  $[0, 1]$ ;

$$precision = \frac{\sum_{i=1}^N tp_i}{\sum_{i=1}^N (tp_i + fp_i)} \quad (4.1)$$

- Micro Recall: represents the number of correctly labeled trajectories over the total number of trajectories in the gold standard. It is defined in the interval  $[0, 1]$ ;

$$recall = \frac{\sum_{i=1}^N tp_i}{\sum_{i=1}^N (tp_i + fn_i)} \quad (4.2)$$

In our case, the Precision and the Recall are equal, as described in the scikit-learn library. On the other side, using the “macro-weighted” performance measures will bring poor performance to our system caused by the wrong classification of a single class with few elements.

We also used another score to evaluate our solution. The Rand index, defined by Willian Rand [16] in 1971, is a measure of partition correspondence used to evaluate clustering algorithms. It is similar to the accuracy measure.

Given a set of  $n$  elements  $S = \{o_1, \dots, o_n\}$  and two partitions of  $S$ ,  $X = \{X_1, \dots, X_r\}$  and  $Y = \{Y_1, \dots, Y_s\}$ , we define the following:

- $a = |\{(o_i, o_j) | o_i, o_j \in X_k, o_i, o_j \in Y_l\}|$  is the number of pairs of elements in  $S$  that are in the same subset in  $X$  and in same subset in  $Y$ .
- $b = |\{(o_i, o_j) | o_i \in X_{k_1}, o_j \in X_{k_2}, o_i \in Y_{l_1}, o_j \in Y_{l_2}\}|$  is the number of pairs of elements in  $S$  that are in the different subsets in  $X$  and in different subset in  $Y$ .
- $c = |\{(o_i, o_j) | o_i, o_j \in X_k, o_i \in Y_{l_1}, o_j \in Y_{l_2}\}|$  is the number of pairs of elements in  $S$  that are in the same subset in  $X$  and in different subset in  $Y$ .

- $d = |\{(o_i, o_j) | o_i \in X_{k_1}, o_j \in X_{k_2}, o_i, o_j \in Y_l\}|$  is the number of pairs of elements in  $S$  that are in the different subset in  $X$  and in different subset in  $Y$

for  $1 \leq i, j \leq n, i \neq j, 1 \leq k, k_1, k_2 \leq r, k_1 \neq k_2, 1 \leq l, l_1, l_2 \leq s, l_1 \neq l_2$ .

The Rand index, is defined as follow:

$$R = \frac{a + b}{a + b + c + d} \quad (4.3)$$

## 4.3 Results

In this section we present our results for the two datasets described in the previous section. First we will take a look to the results for the dataset of Gorrini et al. using both quantitative measures and qualitative analysis. Instead for the Zhang et al. dataset we will present only a qualitative analysis.

### 4.3.1 Gorrini et al.

We can notice from Table 4.1 that the performance of our system are quite good. The system overall obtains an average score of 97 for precision and recall, and 95 for the Rand Score.

| Experiment | Precision | Recall | Rand Score |
|------------|-----------|--------|------------|
| 3_3_A      | 1.0000    | 1.0000 | 1.0000     |
| 3_3_B      | 0.9814    | 0.9814 | 0.9678     |
| 3_3_C      | 0.9074    | 0.9074 | 0.8390     |
| 3_3_D      | 1.0000    | 1.0000 | 1.0000     |
| 4_2_A      | 0.9814    | 0.9814 | 0.9663     |
| 4_2_B      | 1.0000    | 1.0000 | 1.0000     |
| 4_2_C      | 0.9629    | 0.9629 | 0.9387     |
| 4_2_D      | 0.9629    | 0.9629 | 0.9591     |

Table 4.1: Gorrini et al. performance measures

Performing a qualitative analysis on a few interesting experiments, the configuration “3\_3\_C” on which our system performed poorly compared to other results, and the configurations “4\_2\_C” and “3\_3\_B” which presents some interesting issues that allow us to present some weaknesses of the system.

In the figures used to show the results, lines represent pedestrian trajectories, the arrows on the trajectories shows the direction of the pedestrian, and the color of the lane describe it's belonging to a specific cluster.

We can notice in Figure 4.5 the two pink trajectories. As we can see both of them are in a cluster that is quite distant overall, but they are near at the beginning and at the end, and so our system considers them as the same lane regarding that the blue lane is more suitable for the bottom pink trajectory. The other pink trajectory will be more suitable for its own lane. A good classification is achieved for the black and dark green lanes. In both cases the system correctly classified them as independent lanes.

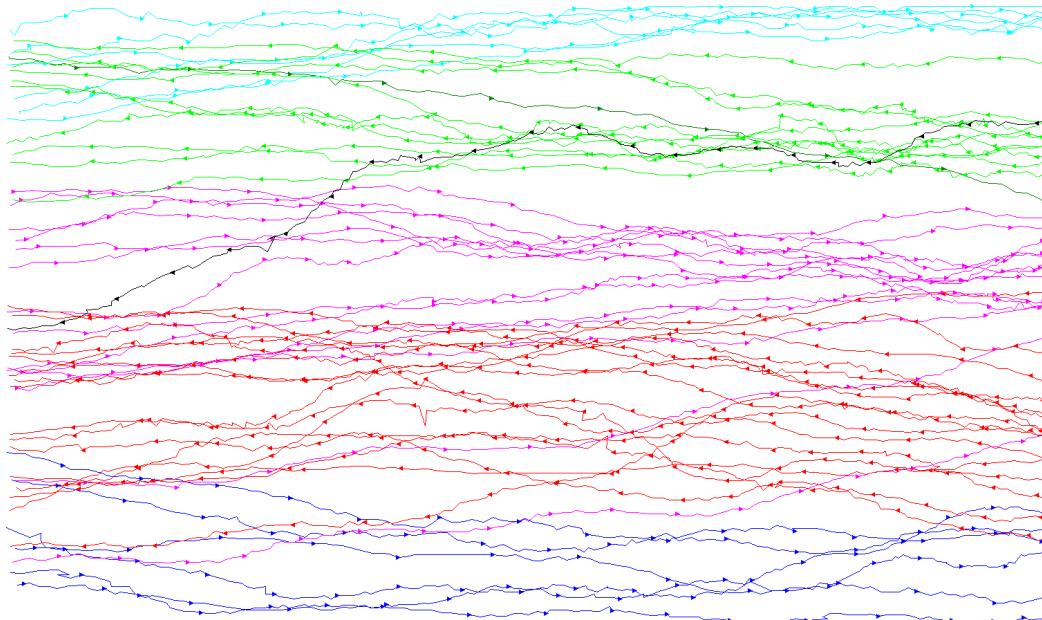


Figure 4.5: Results for the Gorrini et al. experiment “3\_3\_C”.

In configuration “4\_2\_C”, shown in Figure 4.6, similarly to the previous experiment, we can notice large distances between trajectories at the beginning and/or at the end. Our system keeps assigning these different trajectories to the same cluster, hence to the same lane. We can notice in the blue and in the red lanes that this issue is caused by the the low distance of adjacent trajectories that are near others that are slightly off the common trajectory, causing distant trajectories to be closer to trajectories that are correctly labeled.

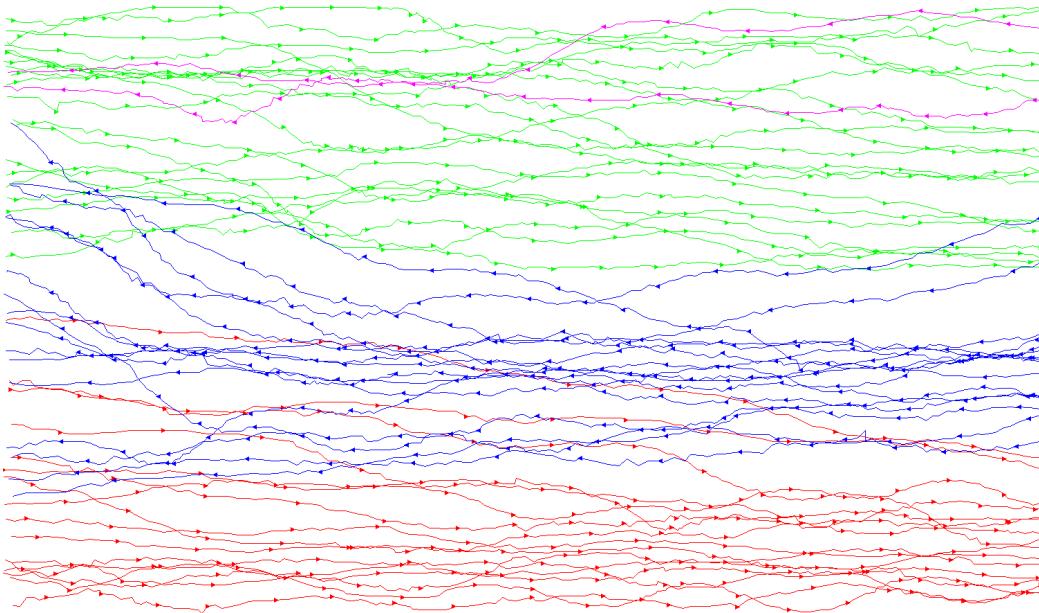


Figure 4.6: Results for the Gorrini et al. experiment “4\_2\_C”.

In the experiment “3\_3\_B” shown in Figure 4.7 beside the previously mentioned issues, we notice that the pink trajectory belongs to its own cluster even if it has the same direction, and behavior of its neighbors that belong to the green cluster. This can be a cause of the Hausdorff distance and the spectral analysis. We can notice the blank spot at the center, where only the pink trajectory appears, this increases the distance between the trajectories, also we notice that the pink trajectory has a snake-like walk, this also increases the distance from the near and more regular trajectories. This increased distance measure means that if the spectral analysis finds an extra cluster the assigned trajectory will be the one with the highest distance.

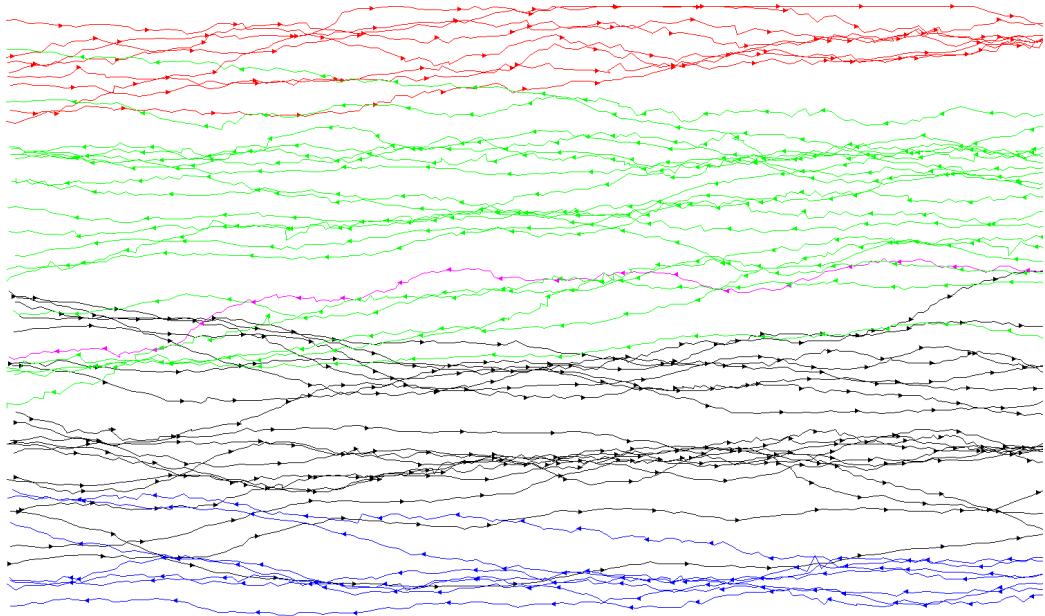


Figure 4.7: Results for the Gorrini et al. experiment “3\_3\_B”.

### 4.3.2 Zhang et al.

The first experiment we analyze from the experiments performed by Zhang et al. is very simple. The experiment shown in Figure 4.8 is a simple example of BFR-SSL. Our system correctly identifies the two lanes and an extra one of a pedestrian that crosses the corridor diagonally.

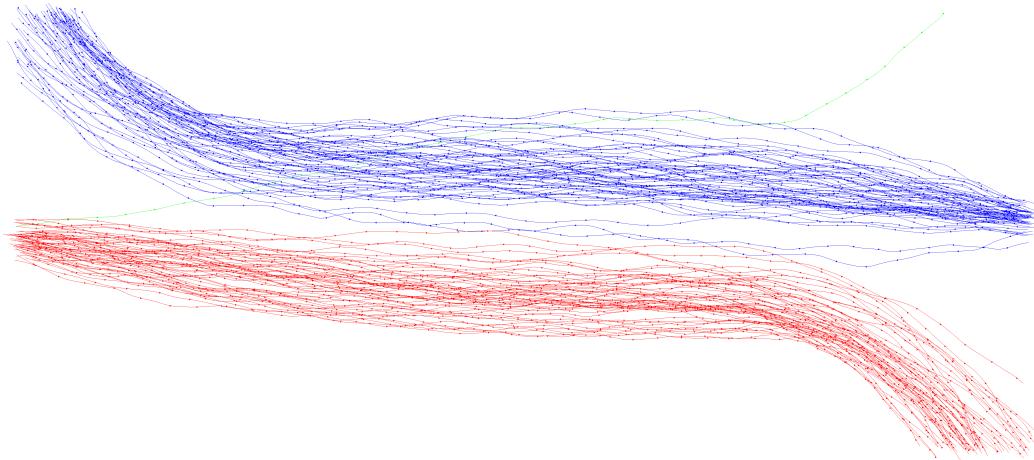


Figure 4.8: Results for the Zhang et al. experiment “bo\_360\_50\_50”.

In more complex scenarios, our system performs in a similar way to Gorini et al.’s experiments. Zhang et al.’s experiments presents more lanes due to the large number of pedestrians thus the results are difficult to analyze. This makes difficult to analyze the results. We will take a look at some of the experiments and we will analyze the strengths of our solution.

In Figure 4.9 we can notice four major lanes, with some noise representing solo pedestrian moving counter flow. In addition to the issues we described for the results of the experiment made by Gorrini et al. we can notice that in this case there is a formation of an UFR-DML flow.

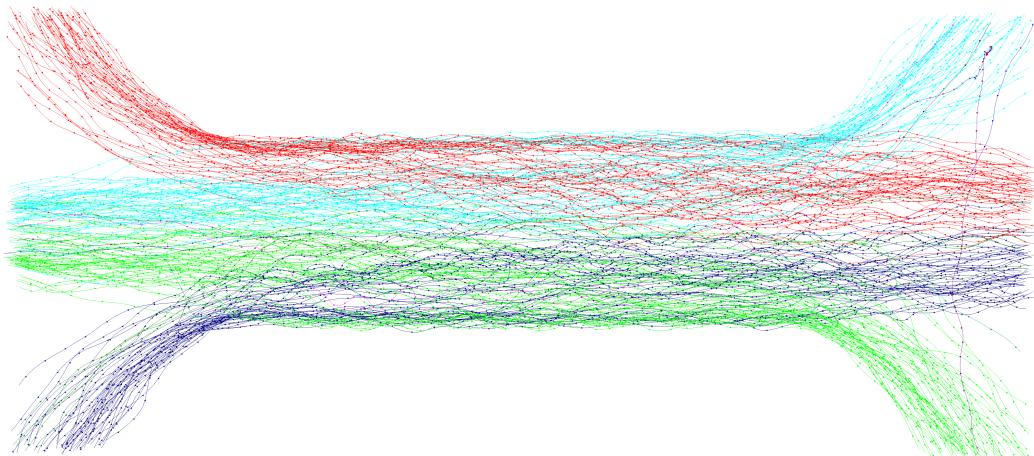


Figure 4.9: Results for the Zhang et al. experiment “boa\_300\_80\_120”.

In Figure 4.10 we notice a similar behavior to the previous experiment. This time the configuration of the gates created a more balanced flow.

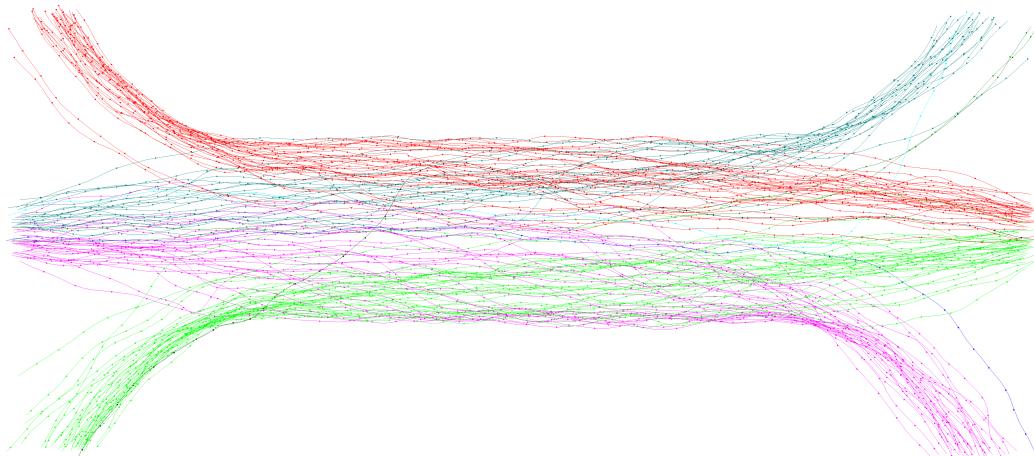


Figure 4.10: Results for the Zhang et al. experiment “boa\_300\_50\_50”.

# Chapter 5

## Conclusions

We presented a system for pedestrian lane recognition based on spectral clustering and Hausdorff distance. We performed analysis on two different dataset. The results, even if the quantitative measures were promising, showed us that a system based on whole trajectories does not give enough information on a lane. This type of analysis does not give us information on how much a lane stays alive, how many times a lane is reformed, and so on. Our type of analysis is more suitable for static analyses such as urban and architectural planning.

As future work, we aim to perform a time based clustering, and try different distance measures. Another improvement could be performed to the clustering algorithm, which actually does not make use of the direction and of the speed of a pedestrian.

# Bibliography

- [1] S. Atev, G. Miller, and N. P. Papanikolopoulos, “Clustering of vehicle trajectories,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 647–657, 2010.
- [2] S. Hoogendoorn and W. Daamen, “Self-organization in pedestrian flow,” in *Traffic and Granular Flow’03*, pp. 373–382, Springer, 2005.
- [3] S. P. Hoogendoorn and P. H. Bovy, “Normative pedestrian behaviour theory and modelling,” in *Transportation and Traffic Theory in the 21st Century: Proceedings of the 15th International Symposium on Transportation and Traffic Theory, Adelaide, Australia, 16-18 July 2002*, pp. 219–245, Emerald Group Publishing Limited, 2002.
- [4] A. Bressan and W. Shen, “Small bv solutions of hyperbolic noncooperative differential games,” *SIAM journal on control and optimization*, vol. 43, no. 1, pp. 194–215, 2004.
- [5] D. Helbing, “Traffic dynamics: New physical modeling concepts,” 1997.
- [6] G. Yuan, P. Sun, J. Zhao, D. Li, and C. Wang, “A review of moving object trajectory clustering algorithms,” *Artificial Intelligence Review*, vol. 47, no. 1, pp. 123–144, 2017.
- [7] H. Miller and J. Han, “Spatial clustering methods in data mining: a survey,” *Geographic data mining and knowledge discovery, Taylor and Francis*, 2001.
- [8] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.
- [9] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise.,” in *Kdd*, vol. 96, pp. 226–231, 1996.
- [10] F. Hausdorff, *Mengenlehre*. Walter de Gruyter Berlin, 1927.

- [11] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *Advances in neural information processing systems*, pp. 849–856, 2002.
- [12] L. Zelnik-Manor and P. Perona, “Self-tuning spectral clustering,” in *Advances in neural information processing systems*, pp. 1601–1608, 2005.
- [13] A. Gorrini, L. Crociani, C. Feliciani, P. Zhao, K. Nishinari, and S. Bagnardi, “Social groups and pedestrian crowds: Experiment on dyads in a counter flow scenario,” *arXiv preprint arXiv:1610.08325*, 2016.
- [14] J. Zhang, W. Klingsch, A. Schadschneider, and A. Seyfried, “Ordering in bidirectional pedestrian flows and its influence on the fundamental diagram,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2012, no. 02, p. P02002, 2012.
- [15] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [16] W. M. Rand, “Objective criteria for the evaluation of clustering methods,” *Journal of the American Statistical association*, vol. 66, no. 336, pp. 846–850, 1971.