FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Pedestrian Behaviour Modelling

João Nuno Fonseca Seixas

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: João Jacob

July 16, 2020

Pedestrian Behaviour Modelling

João Nuno Fonseca Seixas

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. Gilberto Bernardes

External Examiner: Prof. Óscar Ribeiro

Supervisor: Prof. João Jacob

July 16, 2020

Abstract

Virtual simulation environments typically seek to reduce costs, whether time, material or human. In particular, it is of interest to simulate pedestrian behaviour in these simulation environments, as it allows testing changes to the road network, traffic density, among other hypothetical situations, without endangering the health of individuals. It can also be useful to achieve full automation in Autonomous Vehicles. Based on the collection of real pedestrian behavioural data, it will be possible to create models that seek to simulate their behaviour and decision processes, allowing emerging behaviours to arise and be studied in the simulated environment. The goal was to identify information sources and metrics, possible crossing events, develop the pedestrian behaviour model on times of road crossing to analyse and classify possible crossing events and to create a taxonomy for these crossing events.

Using latitude and longitude coordinate points, it finds possible crossing events by detecting pedestrian crossings in a 15-meter radius. The information retrieved is used in automatic clustering algorithms that categorize types and sections of crossing events. Presumed types of crossing such as a diagonal approach—where the pedestrian starts and ends the crossing outside the pedestrian crossing bounds, passing through the middle of the pedestrian crossing—and a more orthogonal method—crossing in a near parallel way to the orientation of the pedestrian crossing—were recognized in the data.

With this work, it's proposed a data-agnostic methodology process to model the pedestrian behaviour in times of crossing. Implementing this process should result in a model capable of creating a taxonomy for crossing events and behaviours.

The ontology created asserts the existence of three phases during the crossing event. A pre-crossing moment preceding the moment where the pedestrian enters the street. A post-crossing one, when the pedestrian returns from the street to the pavement. And finally, the crossing moment, where the pedestrian is in the road.

This work is subject to some limitations. These limitations are from the data used, as it was recognized some inaccuracy in the GPS sensors that collected the data coordinates. Some implementation procedures also bring limitations, for example, the 15-meter radius to find nearby pedestrian crossing may have impacted the quantity of false-positive crossings found.

Keywords: Human Behaviour, Pedestrian, Simulated Environment, Behavioural Model, Modelling, Clustering, Pedestrian Crossing, Pedestrian Crossing Events

Resumo

Ambientes virtuais de simulação procuram tipicamente reduzir custos, sejam esses de tempo, materiais ou humanos. Em particular, é de interesse a simulação do comportamento de pedestres nestes ambientes de simulação, já que permite testar alterações à rede viária, densidade de tráfego, entre outras situações hipotéticas, sem colocar em risco a saúde de indivíduos. Também pode ser útil para obter automação completa em veículos autónomos. Com base na recolha de dados comportamentais de pedestres reais será possível criar modelos que procurem simular o seu comportamento e processos de decisão, permitindo que comportamentos emergentes surjam e sejam passíveis de serem estudados no ambiente simulado. O objetivo é identificar fontes e métricas de informação, possíveis eventos de travessia, desenvolver o modelo comportamental de pedestres em momentos de travessia de estrada e criar uma taxonomia para esses eventos de travessia.

Usando pontos de coordenadas latitude e longitude, são encontrados possíveis eventos de travessia através da deteção todas as passadeiras num raio de 15 metros. As informações retiradas são usadas em algoritmos automáticos de aglomeração que categorizam tipos e secções de eventos de travessia. Os tipos presumidos de travessia, como os de uma abordagem diagonal—onde o pedestre inicia e termina a travessia fora dos limites da passadeira—e o de um método mais ortogonal—atravessando quase paralelamente ao sentido de orientação da passadeira—foram reconhecidos nos dados.

Com este trabalho, propõe-se um processo de metodologia independente de dados para modelar o comportamento de pedestres em momentos de travessia. A implementação desse processo deve resultar num modelo capaz de criar uma taxonomia para eventos e comportamentos de travessia.

A ontologia criada afirma a existência de três fases durante o evento de travessia. Um momento de pré-travessia anterior ao momento em que o pedestre entra na rua. Um de pós-travessia, quando o pedestre retorna da rua para o passeio. E, finalmente, o momento da travessia, onde o pedestre está na estrada.

Este trabalho está sujeito a algumas limitações. Essas limitações advém dos dados utilizados, uma vez que foi reconhecido alguma imprecisão nos sensores GPS que recolheram dados de coordenadas. Alguns procedimentos de implementação também trazem limitações, por exemplo, o raio de 15 metros para encontrar passadeiras nas proximidades pode ter tido impacto na quantidade encontrada de travessias falso-positivas.

Keywords: Comportamento Humano, Pedestre, Ambiente Simulado, Modelo Comportamental, Modelação, Agrupamento, Passadeira, Eventos de Travessia de Estrada

Acknowledgements

To everyone who is reading this, thank you. It's a work of sweat and tears because it has been damn hot these days and I stub my toe. Nah, for real, thank you.

To everyone who I talked and vented my frustrations and problems during this time and this pandemic, thank you. Special thanks to everyone who didn't spoil me the season 3 of Dark.

I also want to thank my supervisor João Jacob and Thiago Rúbio for helping me overcome some obstacles and for their guidance in this dissertation.

Special thanks to my close friends. You know who you are, but I want to go even further and add an extra thanks to humus group and team coffee. Also want to praise Vicente's notebook for the memories of these years.

João Seixas

"But every now and then it's good to question those who question things."

Noah

Contents

List of Figures

List of Tables

Listings

Abbreviations

API	Application Programming Interface
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
GIS	Geographic Information System
GPS	Global Positioning System
LIACC	Artificial Intelligence and Computer Science Laboratory
OSM	OpenStreetMap
SIMUSAFE	Simulator of Behavioural Aspects for Safer Transport
UTM	Universal Transverse Mercator

Chapter 1

Introduction

The current chapter will introduce the context in which the problem is into and the drives to solve the problem. Moreover, it describes the objectives to solve the problem and the following document' structure.

1.1 Context

Currently, with population growth and technology advance, about 70% of the people inhabit towns and cities in the European Region. These urban areas are places regularly identified with heavy traffic[1]. Traffic also comes with traffic accidents, and in Europe, these crashes account for around 127000 deaths and 2.4 million injuries a year. On average, pedestrians and cyclists reckon approximately 30% of those involved in fatal crashes. Pedestrians have double the severity in consequences compared to people in cars[2].

A project that relies on LIACC as one of the research partners, SIMUSAFE - Simulator of Behavioural Aspects for Safer Transport - aims to apply the latest technology in simulation to develop authentic behavioural models in a traffic network that allows testing changes to the network and retrieving information that otherwise would be unobtainable. The goal is to know and prevent what affects crashes and infractions and, ergo, improve the safety of everyone involved[2].

The data to be used will be provided by SIMUSAFE and was collected using a mobile application where data from GPS and accelerometer sensors were recorded. Smart glasses were also used to record video of pedestrians in 1st person perspective and, through computer vision, the information was extracted. The data also contains socio-demographic information.

[1] `http://www.euro.who.int/en/health-topics/environment-and-health/urban-health/urban-health`, *World Health Organization*, World Health Organization, `www.who.int/` (last visited on 23/01/2020)

[2] `http://www.euro.who.int/en/health-topics/environment-and-health/Transport-and-health/data-and-statistics/injuries2`, *World Health Organization*, World Health Organization, `www.who.int/` (last visited on 23/01/2020)

1.2 Motivation

Virtual simulation environments typically seek to reduce costs, whether time, material or human. In particular, it is of interest to simulate pedestrian behaviour in these simulation environments, as it allows testing changes to the road network, traffic density, among other hypothetical situations, without endangering the health of individuals. It can also be useful to achieve full automation in Autonomous Vehicles.

Based on the collection of real pedestrian behavioural data, it will be possible to create models that seek to simulate their behaviour and decision processes, allowing emerging behaviours to arise and be studied in the simulated environment.

1.3 Objectives

To fulfil the problems mentioned above, some objectives are defined for the dissertation. Those are :

- Identify information sources and metrics.
- Identify possible crossing events.
- Develop the pedestrian behaviour model on times of road crossing to analyse and classify possible crossing events.
- Create a taxonomy for crossing events.

1.4 Document Structure

The document includes three more chapters, divided based on the content, which is the following:

- **Chapter 2** - introduces key concepts that will be used along the rest of the document, as well as related work on the area, and the current solutions on the market.
- **Chapter 3** - defines the proposed approach to the problem, and how the implementation should be structured.
- **Chapter 4** - explains decisions taken, used data, technologies and key aspects about the process of the work done.
- **Chapter 5** - discusses the datasets used, the results obtained, problems that surged and how to overcome them.
- **Chapter 6** - ends the document with future work and conclusions about the implementation of the work done.

Chapter 2

State of the art

The current chapter will introduce and explain some concepts needed for a better understanding of the dissertation. It will also describe the current state in the problem's related areas, as well as the current solutions on the market.

2.1 Concepts

2.1.1 Pedestrians

Roads, pavements and means of transportation fill urban areas, and the people that use them are the so-called road users divided into vehicle/car users, cyclists and pedestrians. When an individual walks or runs rather than travelling in a vehicle is called a pedestrian.

Traffic rules and laws started to appear when the motor vehicle started to be the favoured and main mean of transport[41]. Numerous rules were implemented to ease the mobility of traffic, however, following World War 2, a lot of rules have been created to promote traffic safety and, nowadays, only a scattered number of places of human activity aren't controlled by traffic rules[41]. "Road safety depends on three great factors: the road infrastructure, the quality and safety level of vehicles, and the conduct of drivers"[21]. Some drivers' personalities may lead to producing risk manoeuvres and to break traffic rules, which leads to road accidents[21, 25]. But the drivers aren't the only ones at fault. Based on [23], approximately 6% of the drivers performed at least one perceived infraction, while over 21% of the observed pedestrians performed a road-crossing infraction.

Younger pedestrians are more propitious to infringe traffic rules when compared to adults, and when we compare between genders, the men infringe more often than women[23, 24].

2.1.2 Behaviour

When organisms perform any sort of action to respond to stimuli, we can call it the organism's behaviour[30]. The behaviour can be intentional as well as involuntary and stimuli can be from two opposite types: internal (psychological motives such as genes, personality traits and others) and external (environment and weather among others)[29, 22, 30]. Humans, being a living organism, also have behaviours, and, such behaviours change and emerge along with a person's life[29, 22].

2.1.3 Data Modelling

Data is unprocessed information and, by itself, possesses no significance, being able to hold any form, regardless of its usability[19].

"A data model defines the structure and intended meaning of data. However it should also be noted that a data model is restrictive rather than permissive"[38]. The goal of data models is to increase the information's quality utilised to make decisions and it should follow some principles: match requirement's data; be explicit and clear to everyone; remain stable and resilient when data requirements and business practices alter; be reusable; be compatible with other models of the same scope and be capable of mitigating incongruence with different data models[38].

2.2 Machine Learning

Artificial intelligence has different sections, one of them being Machine Learning, which aims to learn patterns from data. Machine Learning divides between three major classes of algorithms, all learning from distinctive types of data. These classes are unsupervised learning, supervised learning and reinforcement learning.

2.2.1 Unsupervised Learning

Unsupervised learning uses unlabelled data to group or cluster similar data. It splits the data into a given number of groups based on the data's attributes[16].

2.2.2 Supervised Learning

Contrary to unsupervised learning, supervised learning algorithms use the labelled data to connect the data attributes of the data set to a specific outcome. The goal is to predict the label of future data[16].

2.2.3 Reinforcement Learning

Reinforcement Learning is an algorithm that sequentially produces actions based on its sensorial perception of the world and on its goal to know how to interact with the enveloping environment. Is often used to train robots for specific situations, e.g., moving objects from a place A to a place B. The objective of reinforcement learning is to connect perceptions to actions, which is called learning a policy. The perceptions are what can be retrieved from the world (i.e., the input from its sensors) and the action is the process of doing something (e.g., the movement in some direction)[16].

For the learning process, it uses a reward system, where a positive or negative score is given based on the quality of the completion of the objectives. In this example, the robot gets a substantial positive reward when it delivers the object in place B, but its penalised with a tiny negative reward for each second it takes to reach place B. The robot starts without knowing its objective of reaching B with the object, so it has to train[16].

Figure 2.1: Reinforcement learning cycle[16]

In complex environments, it may be challenging and time-demanding to learn the right policies, as the rewards may be scarce and scattered, not being available at each step[16]. To address this, **Imitation Learning** may be used to reduce the complexity of the learning search area. Rather than learning through scarce rewards or by manually implementing functions to try to help the learning process, imitation learning identifies and tries to replicate an expert's actions. The algorithm should generalize the demonstrations to apply in new and previously unseen situations[20, 17].

2.3 Pedestrian Behaviour Modelling

The unpredictability that influences human actions and its tight relation with external variables (e.g. meteorological conditions, location, interactions with other people) are some key points that make it difficult and complicated in modelling the human behaviour. One major goal of Human Behaviour Modelling is to have the possibility to predict and meet people's necessities, assist in the day-to-day basis and lessen their errors. Nowadays, Human Behaviour Modelling is applied to anticipate behaviours in evacuations, control driver's human reactions, recreate interactions between people, among other applications. Additionally, Human Behaviour Modelling is expected to eventually predict the eventual behaviour of a person[33].

Models for Pedestrians are categorized on their space representation (continuous, grid-based, network structure), the purpose (specific purpose or general-purpose) and level of detail (macroscopic, mesoscopic, microscopic)[37].

- **Macroscopic Models**

  "Macroscopic models focus on the aggregate representation of pedestrian movements in a crowd through flow, density and speed relationships"[37]. They operate the movement of pedestrians like a constant fluid and entrust the fluid's behaviour as an interactive system on a large scale. The use of this type of models can be befitted for very high density, large systems where the behaviour of groups of units is suitable. Still, by using the pedestrian as an "unconscious" member, the macroscopic models don't consider the diversity in the behaviour of individual pedestrians that are able to considerably change the crowd behaviour as a whole, particularly in emergency situations[37].

- **Microscopic Models**

  Microscopic models focus on mapping every pedestrian of a crowd as an individual agent holding a specific space on a moment in time. These models consider factors that relate to social interactions between pedestrians that push them to their destination, resulting in a more realistic portrayal of pedestrian behaviour, but at a computation effort and cost relative to the level of detail. Microscopic models can be separated into four groups[37]:

  - **Physical-based models**

    Physical models admit that individuals who react to the events around form a crowd. Physical models are mainly used to analyse indoor panic and emergency scenes to outline evacuation strategies. These models use multiple physical forces to find the optimal acceleration and the emerging models based on this are the Magnetic Force Model, the NOMAD (Normative Pedestrian Behaviour Theory) and the Social Force Model[37].

  - **Cellular-based models**

    The objective of cellular-based models is to have the area divided into cells, each of them only being able to contain one pedestrian, and representing free floor areas, obstacles, areas filled by pedestrians or groups of pedestrians, or even areas with other environmental attributes. Cellular based models are likewise known as a matrix-based system or Cellular Automata[37, 34].

    Even though researches proved the aptitude of these type of models to model pedestrian behaviour, there are difficulties in simulating crowd cross flows and concourses, and when displaying the model graphically, it seems unrealistic, as the movement is shown as hopping through the cells[37, 34].

  - **Queuing network models**

    Queuing models are a discrete event-based model using probabilities to designate pedestrian movement with priority rules dictating interactions between pedestrians.

These models follow the premises that every building is capable of being modelled as a walkway sections' network and pedestrian flow is capable of being modelled as a queuing network process that any pedestrian is handled as an individual flow object that interacts with other objects. These models were implemented in building's evacuations to estimate congestion and evacuation time, but there isn't evidence on the model's validation[37, 31].

– **Multi-agent models**

The objective in Multi-agent models is to have a set of agents behave on their own favourably to their strategies. An agent is an individual unit capable of being autonomous and goal-oriented and may be able to have other abilities like intelligence and adaptability. The agents interact with each other and these interactions can be designated by certain restrictions of behaviour corresponding to following or leading other agents or even stopping due to congestion. These models are considerably expressive in space-time dynamics as they enable interactions exploration of relationships between micro-level individual actions and emergent macro-level phenomena. Multi-agent models have been the favoured method for evacuation scenarios and complex systems in large scale[37, 34, 18].

Nonetheless, multi-agent simulations usually don't consider force effects, which are an important factor in modelling crowd behaviour, since these effects can prompt people to deviate from their sought trajectory to the objective, and accurate models must consider this[37, 28].

- **Mesoscopic Models**

Mesoscopic modelling focus on groups of elements in the same situation, instead of a single element. If a vehicle, for example, is going at an identical velocity as all the other vehicles in the same zone, they would be viewed as a group[37]. "Instead of modelling a single pedestrian, groups of pedestrians are used and every group has its own rules of behaviour"[37, 27].

According to Nicoletti et al., the major approaches to Modelling Human Behaviour were developed by scholars between the mid-'50s and the mid-'90s. These approaches are:

- **Knowledge-based**

Knowledge-based approaches are the implementation of IF-THEN algorithms, which boil down to act if certain conditions are met. Even though Knowledge-based Systems are easily extended and updated, it has major downsides. The disadvantages of these algorithms are that it isn't self-learning, requiring prior knowledge of the conditions to map them. Moreover, the depiction of human behaviour and decision making is essentially impossible through a static and simple algorithm, due to its complexity[33].

- **Agent-based modelling**

Agent-based modelling is based on the interaction and reaction between agents and the surrounding environment. The agents are autonomous, having an assigned behaviour and goals, having the possibility to adapt and learn from the interactions with other agents and the enveloping world, avoiding obstacles to reach their objectives. These models are one of the best to model human behaviour but still has a drawback when trying to model the irrationality of certain human behaviour[33].

- **Artificial Neural Networks**

  Artificial Neural Networks tries to replicate the nervous system of humans with neurons with weighted connections among themselves. These weights are what defines how the processed information is represented. As such, the Neural Networks resemble the human system, as they assess stimuli and produce an outcome action, although reproducing the Human neural process is a complicated process. The time and processing cost are the major drawbacks of this approach, but it can be used in situations where linear programs aren't well suited. The model is self-learning and is reliable as if some element suddenly fails the network still works[33].

- **Fuzzy Logic**

  Fuzzy Logic uses the notion of partial truth to model events with high uncertainty or unreliable information. The truth values vary from decimal numbers between 0 to 1, or as called, from completely true to completely false. Human Behaviour Modelling is involved with high uncertainty, as there's an uncountable number of probable actions, which makes it viable to use with Fuzzy Logic. Although it's made to work with imprecise knowledge, it's based on stochastic assumptions and, the resulting outputs don't have a defined interpretation, which leads to a difficult analysis[33, 35].

- **Genetic Algorithms**

  Genetic Algorithms are based on genetics and Natural Selection from Charles Darwin's theory of evolution, using the terminology chromosomes to represent the variables of a solution. These chromosomes can be altered and evolved from generation to generation to produce better solutions to the optimization problem. Genetic algorithms are good if a person needs to fulfil multiple objectives at the same time, which the above algorithms don't anticipate. They are also good with stochastic information. The need for a substantial population for better results, the dependency on the quality of the fitness function (the function that evaluates the solution) and, in multi-objective scenarios the impossibility to get to an optimal solution are the weaknesses of these algorithms[33].

- **Markov Chains**

  Markov Chains uses states to represent the chain of events/actions, making usage of probabilities to move between states. The model can only be in one state at a time and doesn't need to know past actions or states. These models are simple to infer from sequential data, can be highly reliable and the results are easily graphical represented. Despite this, this approach has some limitations since Markov Chains are stochastic models, do not incorporate self-learning and cannot model human interactions[33].

There are still some other approaches being developed. These are named Ambient Intelligence, Data-Driven, Dynamic Factors, Human-Centred Systems and Video-Analysis.

2.4 Related Work

Guo et al. used a Hazard Model based on the time from arriving till crossing the road to analyse the pedestrian crossing behaviour in signalized pedestrian crossing. The article reported a relationship between the waiting duration and a crossing violation, also influenced by a person's internal and external factors. About 10% of the pedestrians didn't even wait, and only a sparse people remained after waiting over 65 seconds, although 50% of pedestrians will still wait after 50 seconds, inferring that "the longer the time that has elapsed since the start of the waiting duration, the more likely pedestrians will end the wait soon"[26].

Using a Bayesian Network and making use of probabilistic dependencies among variables such as the season, speed of vehicles, weather, signal waiting time on the pedestrian crossing, and if the pedestrian respects or not the current crossing signal, Yi et al. modelled Human Crossing Behaviour. It concluded that the willingness of an individual to respect the law is the major influencing factor when crossing the street and the bigger part of the pedestrians follow the signs. The cycle timing of the signal is another impacting factor. The season and weather also play a key aspect, but not as important, as, with rain, the pedestrians grow impatient faster and tend to break the law when they see an opportunity to cross.

Shaaban and Abdel-Warith used an agent-based implementation to replicate the movement of pedestrians' illegal crossing in a multi-lane situation, trying to emulate the perception of the rolling gap. Uses a type of agent for controlling the vehicle, only changing the speed, and another type for the pedestrian, using the retrieved data. After testing with multiple distributions, it concluded the one that best fit the data was a normal distribution with a factor varying from 1.25 to 1.5. With this, it was also deducted "that the pedestrians add a factor to the vehicle speed before anticipating the gap, and then determine whether to cross or not"[36].

Using a layered behaviour modelling framework with an agent-based approach, Luo et al. develops a "generic behaviour modelling and simulation framework for crowd simulation, with focus on imitating real human's decision-making process". Every agent has dynamic attributes—physiological, emotional and social group—which can be updated based on the agents' situation awareness comprised of sense, reason and memory. Their case study reports that the behaviour model can produce realistic human behaviours in a crowd and that it can adjust to user-defined scenarios.

Zeng et al. used a social force model to portray the pedestrian psychological process in a signalized intersection. It contemplated the natural pedestrian behaviour such as individual and group evasive manoeuvres, reaction to the limits of the pedestrian crossing, and collision avoidance with vehicles. Zeng et al. further says their need to improve the model's reaction of pedestrians to vehicles, and the need to account the gender, age, partner relationship and other characteristics as determining aspects of crossing behaviour.

2.5 Solutions on the market

Although our goal is to develop the pedestrian behaviour model focused on pedestrian behaviour in times of interaction with other traffic actors and in times of road crossing, the solutions that currently exist are mostly specific to crowd simulation, either in evacuations or for new infrastructures planning to see the crowd flow. There are also models for visual effects in films to create and simulate crowds, for example.

- **MASSIVE**

  MASSIVE stands for Multiple Agent Simulation System in Virtual Environment and was developed by Stephen Regelous for use in Peter Jackson's *The Lord of the Rings* films. It's the leading software for crowd related visual effects and autonomous character animation, applied primarily in film and television by studios such as Pixar, Sony Pictures Imageworks, ImageMovers Digital, among others. *Game of Thrones*, *I, Robot*, *The Dark Knight*, *Ben Hur*, *World War Z* and *John Carter* are some examples of its usage. This software package is also used for Education, Architectural Visualization and Engineering Simulation, making usage of fuzzy logic. The software enables every agent to respond individually to its surroundings, including other agents, making usage of simulated natural senses of sight, hearing and touch[7].

Figure 2.2: Massive Software 7.0[8]

- **ALICE**

  ALICE stands for Artificial Life Crowd Engine and was developed originally for *Troy* in 2004. It's one of Moving Picture Company's flagship software product which allows crowd behaviour management, motion clip editing and blending and customised scripting for large groups of agents. The software is capable of producing up to several hundred thousand agents to simulate huge armies, flocks of birds, swarms of insects, zombie hordes and space battles, among other scenarios. Some of the films which used ALICE are *Exodus: Gods and Kings* and, like MASSIVE, *World War Z*. ALICE has some Group Behaviour modules for when a large number of characters are together, they need to interact among them, and as such, these modules have collision avoidance and flocking, where the group members will try to stay close to one another[1].

- **Golaem Crowd**

  Golaem Crowd is a plug-in developed by Golaem for Autodesk Maya that allows for the simulation of controllable digital characters crowds based on independent agents, from few to thousands. Golaem is used in commercials, episodic productions, feature films and games. Some examples of its applications are the season 5 of *Game of Thrones*, the game *Guitar Hero Live*, season 6 of *The Walking Dead*, the film *22 Jump Street* and the commercial from Orange *Be Prepared*. Its primary usages are to populate stadiums and concert halls, fill city streets or public spaces with passers-by, invoke armies, with battle formations, and create battles, and everything from horses to aliens and bikes[5].

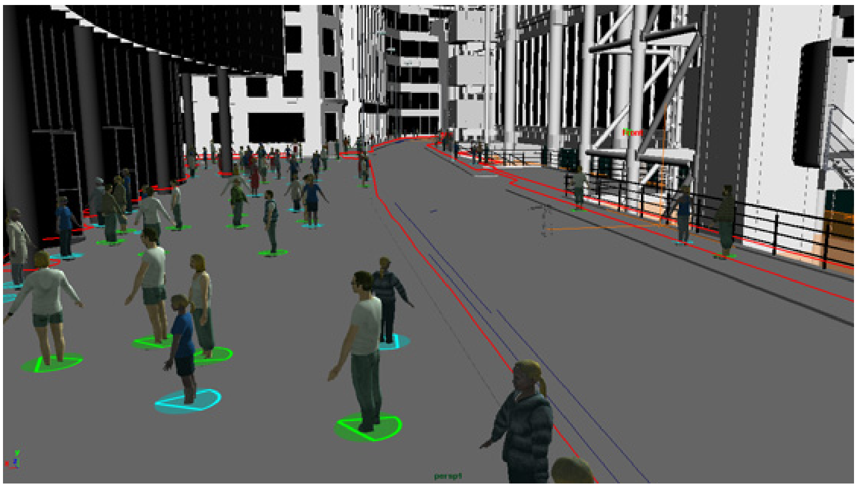


Figure 2.3: Golaem Crowd[6]

- **PTV Group**

  The PTV Group specialises in software solutions to improve mobility and transport, save time, improve road safety and reduce the impact on the environment. The most relevant traffic software for this work developed by the PTV Group are[11]:

  1. **PTV Visum** - a macroscopic model that's the world's leading software for traffic analyses, forecasts and GIS-based data management;
  2. **PTV Vissim** - world's leading software for microscopic traffic simulation;

3. **PTV Viswalk** - a microscopic software for pedestrian simulation, used in urban and construction planning, pedestrian safety planning and evacuation measures.

From the prior software packages, *PTV Viswalk* is tightly connected to this dissertation and implements, with some extra functionalities, the Social Force Model developed by Prof. Helbing[12].

Figure 2.4: PTV Viswalk crowd simulation[13]

- **Paramics**

Paramics is from Quadstone Paramics, a prominent microscopic pedestrian and traffic simulation software. It's used to plan infrastructures and urban spaces with estimations for traffic conditions, to manage situations from simple intersections to a city's traffic system and to evaluate interactions in traffic for accident analysis and safety prevention. Makes use of agent-based modelling to simulate the pedestrians, with every agent having physical attributes, goals, behavioural rules and being subject to the surrounding environment constraints[14].

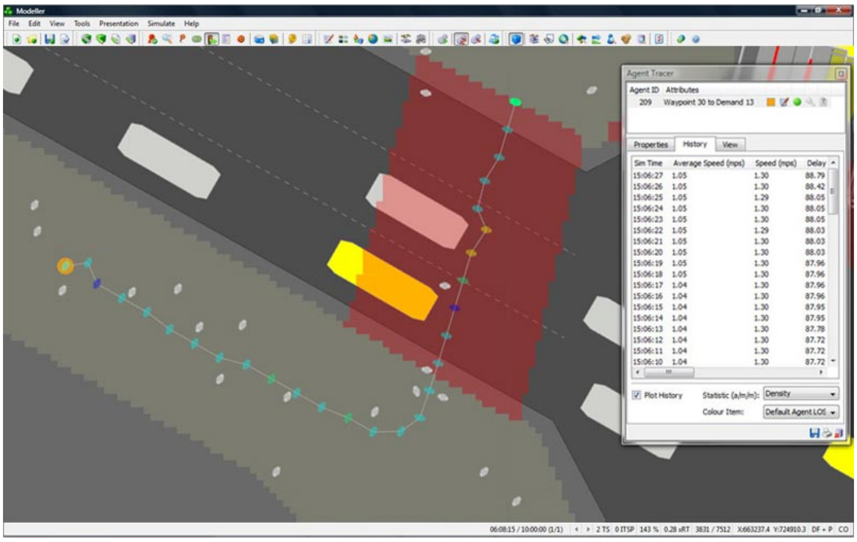


Figure 2.5: Pedestrian routing in Paramics[15]

Chapter 3

Methodology

In this chapter, it's proposed a way to approach the current problem of identifying and classifying different crossing events. In Figure 3.1, the suggested process scheme is presented, which passes through the collection of data from data sources, detection of crossing events and analysis and classification of these events.

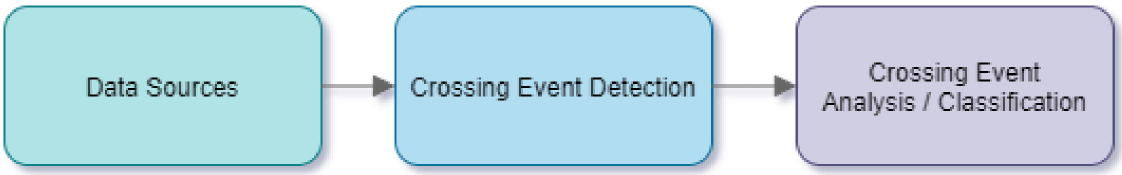


Figure 3.1: Proposed methodology process scheme.

3.1 Data Sources

The whole process depends on the data used. As such, to have an adequate source of data is required to apply the procedure. These sources should have reliable and relevant information such as the pedestrian's geographical location and the related context, i.e. what surrounds and affects the course of the pedestrian, also containing information about the current time, weather and season. Furthermore, it should incorporate the social-demographic information (e.g. age, gender), as well as information about the roads and the pedestrian crossings (e.g. if it has traffic signals, the number of lanes, its geographical location and orientation). For this required information, it's suggested that the data is obtained from recorded videos of on-site experiments, along with other pieces of information (e.g. GPS location) and manual annotations of these episodes, plus inquiries and reliable sources of geo-data (e.g. OpenStreetMaps, Google Maps). The information also should be stored in a format which can easily create and analyse tables, as well as freely manipulate and prepare the data to be used on each step of the process, e.g. a CSV file format.

It's in this section that the data should be described and firstly analysed. Additionally, it's where the data will be prepared to use in the rest of the development. It should be an iteration process, where the new requirements for the data, as well as new data (if it's the case) for the methods of detection and classification of crossing events, serve as input to a data processing section. There, the data is analysed and prepared according to the needs and outputs a ready-to-use data for the aforementioned methods, as it can be observed in Figure 3.2. Also, it may be necessary to process data during the detection and classification of crossing events.

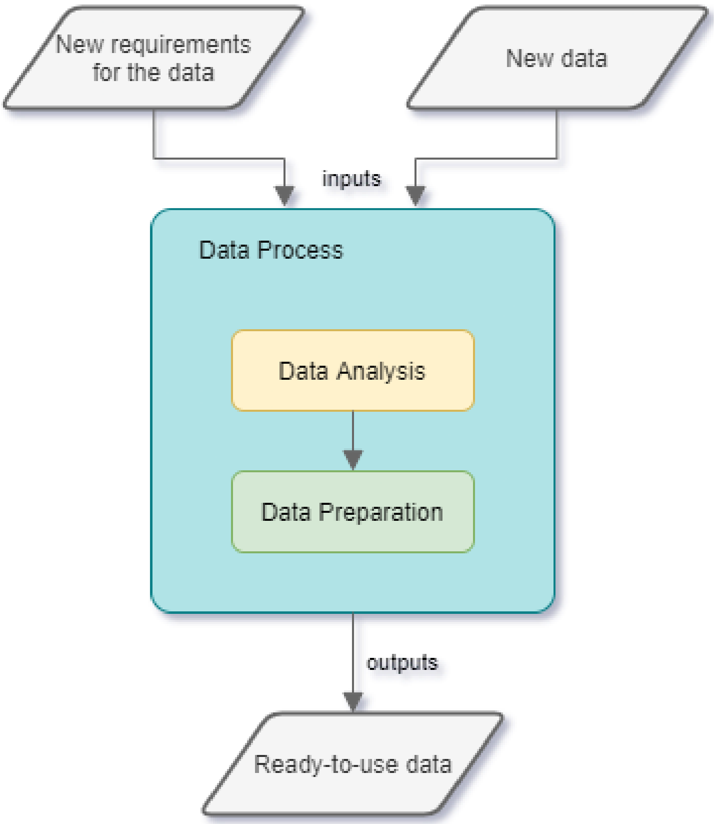


Figure 3.2: Proposed data iteration process.

3.2 Crossing Event Detection

Independently from the data available, it's required to identify the crossing events to complete the goal of modelling the pedestrian behaviour. It's considered a crossing event when a pedestrian does the sequence of events of leaving the pavement, entering the street, and returning to the pavement, either it being on the same side of the road that it had previously left (a cancelled crossing) or being on the other side of the road (an actual crossing). Additionally, these events are only recognized when there's a pedestrian crossing nearby, not considering cases of illegal crossings.

To detect these events, it's needed information about the movement of the pedestrian, as well as information related to its surroundings. This data can be obtained from videos, manual annotations, traffic cameras, inquiries or other sources, as previously mentioned in Section 3.1. The detection of these events serves as a way to verify the information in the data, if it already has information regarding the crossing events, or to find new evidence that leads to uncovering this information.

This part of the process requires cleaned and prepared data, which can be easily manipulated by the demanded implementation of the detection method. It should produce data regarding the found crossing events, as well as the ones that consider as noise. It's suggested implementing multiple different identification methods, as it serves to cross-validate the detections of the crossing events.

3.3 Crossing Event Analysis / Classification

This section is the last step of the process. It depends on the quality of the detection methods used in Section 3.2 as well as its data output, and the necessary data preparation.

The implementation of this part of the process should be able to cluster the data into different segments of a crossing event. The model used ought to classify the data into its respective division of the conceived sectors that surrounds a pedestrian crossing event, returning the labels from the classification. It's also expected a division of crossing events in 3 distinct moments: a pre-crossing phase, a post-crossing one and the crossing. In the last instance of this method, it should describe possible ways of crossing the road, as well as predicting crossing events, which will prove useful for, for example, autonomous vehicles. The outcome will give a perception of what types of typical crossings exist.

In Figure 3.3, it's proposed a flowchart for the whole methodology. The flow starts in the collection of data from the previously identified sources, proceeding to where it should be analysed and prepared according to Section 3.1 of Data Sources. It then iterates a loop around the Detection of Crossing Events from Section 3.2, executing specific Data Processes of Figure 3.2 for the requirements of its implementation. The next segment is the loop of the Classification Process until it produces a working method, as it's described in Section 3.3, also applying the same data processing procedure as the Crossing Event Detection.

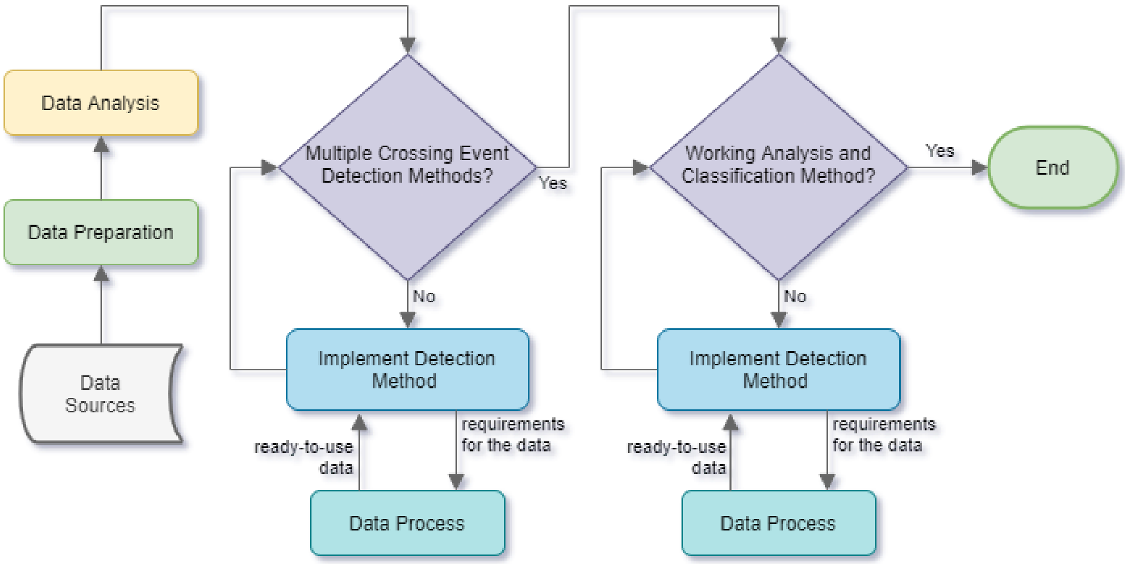


Figure 3.3: Steps in proposed methodology flowchart.

Chapter 4

Implementation

This chapter addresses the implementation of the processes mentioned in Chapter 3. It conveys decisions taken, scripting languages and packages used, as well as information about the data and the logic in creating the whole solution.

4.1 Analysis and preparation of data

In this section is introduced the dataset used in the implementation. Furthermore, it explains the steps that led to analysing this data as well as preparing the data to use in Sections 4.2 and 4.3.

4.1.1 Data Sources

Multiple data sources were needed and used to implement the processes accordingly to the methodology scheme from Chapter 3. Although the major volume of data came from SIMUSAFE project, it was also retrieved data from OpenStreetMaps using the Overpass API and the folium[1] package.

4.1.2 Data Description

The data used from the SIMUSAFE project contains GPS and accelerometer information retrieved from a mobile app using its mobile sensors to record it. The data also contained first-person videos with the pedestrians' perspective recorded by smart glasses. These videos were divided into episodes, based on the subject's recording sessions. Through computer vision, the videos' information was also made available in JSON files (Listing 4.1). The SIMUSAFE's data also contained social-demographic information related to the subjects, for example, gender, education level and attitudes related to road violations. Lastly, there were experts' annotations based on the videos, which contained information about the crossings, including timestamps, perceived risk, road and weather conditions, as well as inquiries done to the subject about the crossings.

[1] `https://python-visualization.github.io/folium/` (last visited on 01/07/2020)

```
{
    "frames": [{
        "frame": 1,
        "second": 0.03333333333333333,
        "timestamp": 1537465382.033,
        "context": {
            ...
        }
        "actions": {
            "left": "5.83",
            "right": "0.00",
            "up": "0.74",
            "down": "4.81"
        },
        "time": "2018-09-20 19:43:02",
        "sensors": {
            "event": "no_event",
            "gX": "0,0293982420",
            "gY": "-0,0681388229",
            "gZ": "0,0573818423",
            "latitude": "42.3316424",
            "longitude": "-3.7011271",
            "label": 0,
            "timestamp": 1537461782681,
            "speed": 0,
            "acc": 0.09380741785504064
        }
    }, {
        "frame": 2,
        "second": 0.06666666666666667,
        ...
    ...
    },
    "video_info": {
        "fps": 30.0,
        "file": "B07M3-ep-0-20180920_194302.mp4",
        "height": 1080,
        "width": 1920
    },
    "annotation_events": 0
}
```

Listing 4.1: Example JSON file.

The data from the Overpass API comes as a JSON response to a query. It has information related to the nodes or ways of the map from the queries' sent. The information contains ids, geographical location and descriptive tags.

4.1.3 SIMUSAFE's Data Preparation

The data available in JSON files, described in Subsection 4.1.2, was converted into CSV files to make tables and manually create charts more easily to find predicted cues in the data. A total of 14563 entries were deleted during the conversion of the files when there were missing values, resulting in 278911470 valid entries.

4.1.4 Analysis of SIMUSAFE's Data

With the ultimate goal to help the process of creating a model to classify the events of pedestrians crossing in a pedestrian crossing, the first step is to identify anticipated values in the data. These cues can be, for instance, a change in speed from the pre-crossing phase to the crossing one, and the same situation from the stage of crossing to the post-crossing.

The data available from SIMUSAFE was prepared as specified in Subsection 4.1.3 so it could be analysed. An example of the charts created is in Figure 4.1 from a part of one episode related to the subject B07M3. It represents the variations in speed (meters per second) along the time (in seconds) and, from it, two sudden changes in the speed at around the 325 seconds mark and the 475 seconds one can be identified. These two changes can be related to the subject reducing its velocity before a pedestrian crossing, however, they can also be linked to many other possible situations, such as interactions with other pedestrians.

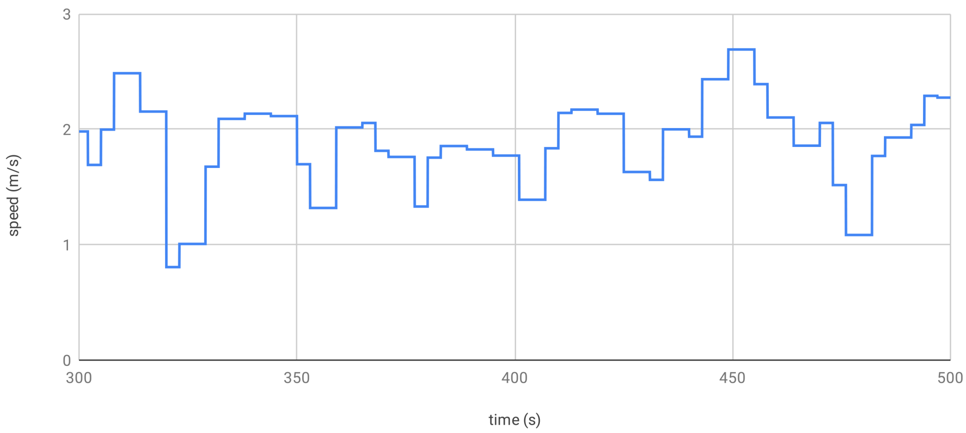


Figure 4.1: Speed variation along the time.

With the data available alone, it is not possible to know if a subject is near a pedestrian crossing or about to cross it, so it's necessary to obtain this information to identify and verify the possible cases of crossing events.

The data also served as input to RapidMiner, intended to help find cues in the data and correlation between its variables to future help in classifying the crossings. The prediction models failed to output a working version, and it was assumed to be because of missing context information about the crossings.

4.1.5 Data Preparation for Nearby Pedestrian Crossing

The data in the JSON files and in the converted CSV files had entries for each frame of the related video. It introduced redundant information not needed since values of latitude, longitude and speed did not change for a considerable amount of video frames, as it is possible to observe in Figure 4.1. The files were cleaned and appended to each other to simplify and remove the extra information, as it is possible to view in Figure 4.2.

frame	second	...	latitude	longitude	speed
1	0.03333333333		42.3349494	-3.7025798	0
2	0.06666666667	...	42.3349494	-3.7025798	0
3	0.1		42.3349494	-3.7025798	0
...					
150	5	...	42.3349494	-3.7025798	0
151	5.033333333		42.3349498	-3.7025801	0.008487376532
...					
18749	624.9666667	...	42.3428523	-3.7013696	1.860832512
18750	625		42.3428523	3.7013696	1.860832512

Table 4.1: Excerpt example from episode of subject B07M3.

file	starting_frame	number_of_frames	...	latitude	longitude	speed	seconds
0	1	120	...	42.3316424	-3.7011271	0	4
0	121	150		42.3317617	-3.7012002	3.005789628	5
...							
0	6721	90		42.3346531	-3.7025165	1.867143801	3
1	1	150	...	42.3349494	-3.7025798	0	5
1	151	90		42.3349498	-3.7025801	0.008487376532	3
...							
92	31	120	...	42.3406763	-3.7170591	0	4

Table 4.2: Excerpt example from cleaned CSV.

4.1.6 Data Preparation for Subject's Path

The experts' annotations contained manually detected pieces of information regarding crossing events from the videos. As such, they were treated as absolute truths. Using only the cases present in these data meant that it could be verified and wouldn't be wasting resources on other situations.

To the aforementioned data was added an extra range, when possible, to make sure there were enough points to create a spline since the implementation of the Catmull-Rom spline required at least four points.

4.1.7 Data Preparation for Heatmaps

Since the current data didn't have a continuous path of the subject, splines were created based on the GPS coordinates of the pedestrians to generate more samples of positions. This was intending to create heatmaps for the crossings, so ultimately they could be clustered and classified. To accomplish these heatmaps, all the splines needed to be normalized, including the orientation of the pedestrian crossing.

In order to normalize the splines, each value of latitude and longitude was subtracted by the corresponding values of its crossing node, making every crossing node point the centre point (0,0). With the end to rotate the orientation of the pedestrian crossing, the current orientation needed to be known. A new query to the Overpass API was produced to this objective, collecting the street way, where the crossing node was located, and the adjacent nodes to the crossing node that belonged to the way.

With all the aforementioned, it was obtained Figure 4.2, where it's possible to observe the differences from before (on the left) and after (on the right) normalizing the data points. The red points represent the subject's recorded pairs of coordinates, the blue line portrays the spline and the green point shows the node centre of the pedestrian crossing.

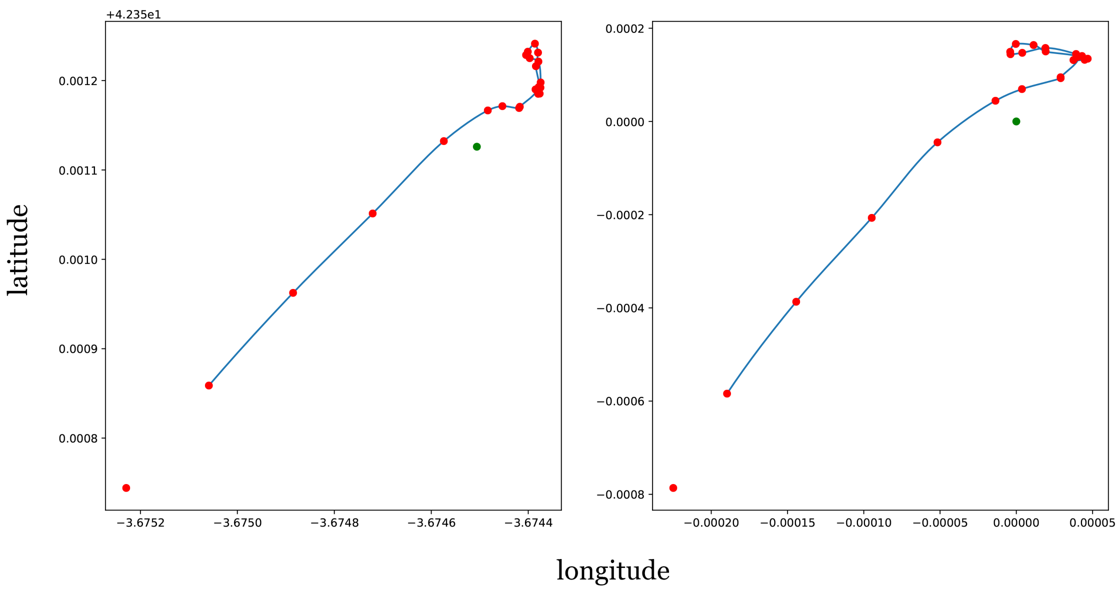


Figure 4.2: Representation of before normalizing and vertically orient the spline and after.

4.1.8 Data Preparation for Clustering Methods

To prepare the data, it was necessary to address some problems that arose in Subsection 4.3.3. For each event of crossing, every pedestrian crossing node was separated and counted as an individual crossing event.

One other problem in the data was the length of the event. To fix this, the data outside a 15-meter radius from each event's pedestrian crossing was ignored, as it was considered unnecessary information.

Using the splines revealed that it had too many points and therefore too much information to handle. Only using the data coordinates of the subject course reported missing information. The final approach was to combine the two options and start with the subject points. When the distance between coordinates was lower than 0.5 meters, the coordinate point was removed. When the distance was bigger than 1.25 meters, equally spaced points from the spline were sought to be added. All these still within the aforementioned radius (Figure 5.7). In some cases, it was still possible to add points before and after with values from the spline, and as such, these points were appended if within the data range of 15 meters (Figure 5.8).

4.2 Identification of Pedestrian Crossing Events

In this section is shown the process behind the various methods used to identify pedestrian crossing events. These approaches used distance to identify nearby crossing events (Subsection 4.2.1) as well as timestamps' intersection (Subsection 4.2.2).

4.2.1 Overpass API to Identify Nearby Pedestrian Crossing

Following a data cleaning as described in Subsection 4.1.5, the next step was to collect the geographical data needed. The chosen database was the OpenStreetMap one as it stated that was a "project that creates and distributes free geographic data for the world"[9]. Since it only needed a read-only API of the OSM map data, the python package overpass that serves as python wrapper of the Overpass API was chosen[10].

A python script was developed with the referenced python package overpass to identify situations where the subject is near a pedestrian crossing. The script queried the Overpass API in order to find and save every node of a pedestrian crossing in a 15-meter radius for each pair of coordinates in the cleaned data of Figure 4.2. The chosen limit is due to the inaccuracy in the data that come from the phone' sensors.

4.2.2 Timestamp Intersection with Experts' Annotations

The experts' annotations file, with manually annotated information about the videos with the multiple recorded episodes of the subjects, had information about each pedestrian crossing event, including the start and ending timestamp, along with other notes related to the event. These timestamps were intersected with the timestamps in the cleaned CSV file (Figure 4.2) to identify cases of pedestrian crossings.

The diagram in Figure 4.3 represents the logic behind the intersection. The blue time gap outlines the annotations timestamps, the green ones depict the values that intersect with the annotations timestamps, and the reds are those which don't intersect.

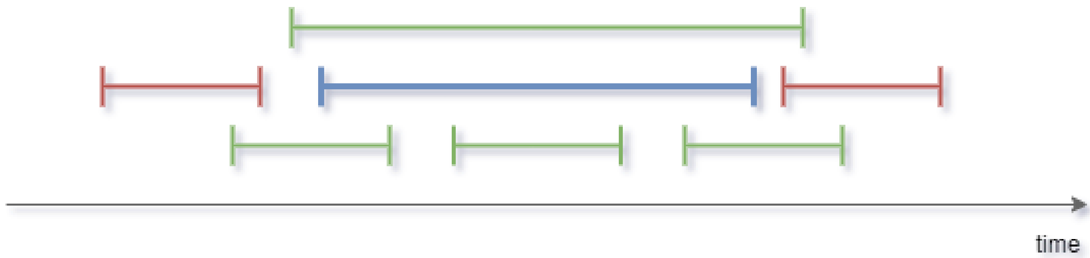


Figure 4.3: Diagram for timestamp intersection.

With the entries related to true events of crossing, it was implemented a way to identify the pedestrian crossing nodes linked to it. That would be necessary to display on the map in subsection 4.3.2 and for heatmaps on subsection 4.3.3.

4.3 Clustering and Classification of Crossing Events

In this section is presented the implementation of multiple approaches to try to cluster and classify different pedestrian crossing events. These methods used angles (Subsection 4.3.1), heatmaps (Subsection 4.3.3) and clustering algorithms (Subsection 4.3.4).

4.3.1 Simple Prediction Model Based on Angles

With the goal to create a model to predict the crossing events, a basic and naive model was developed, based on the angles between the data coordinates of the subject and on the angle between these coordinates and the nearby pedestrian crossing node. The angles that served the base to classify the possible crossing as a true crossing event were based on the presumed possible angles observed in Figure 4.4 and in Table 4.3.

As it was previously done in Section 4.1.4, the angles' information served as input to RapidMiner to help in classifying the crossing events.

The process in subsection 4.2.2 of the timestamp intersection with the experts' annotations was used to evaluate the model. The evaluation revealed a poor performance in predicting the outcome. The poor results were associated with the rudimentary angle approach, which did not work with the current data since real data isn't as linear as the presumed predefined points (verify in Figure 4.4) and can fluctuate along the pedestrian crossing and the pavement. Additionally, due to data inaccuracy, its position can have some offset.

To improve this model, a higher number of pedestrians' data points would have been needed, where a complete line of their path was available. With this, it could have been possible to select points that matched from the ones in red from the presumed angles (Figure 4.4).

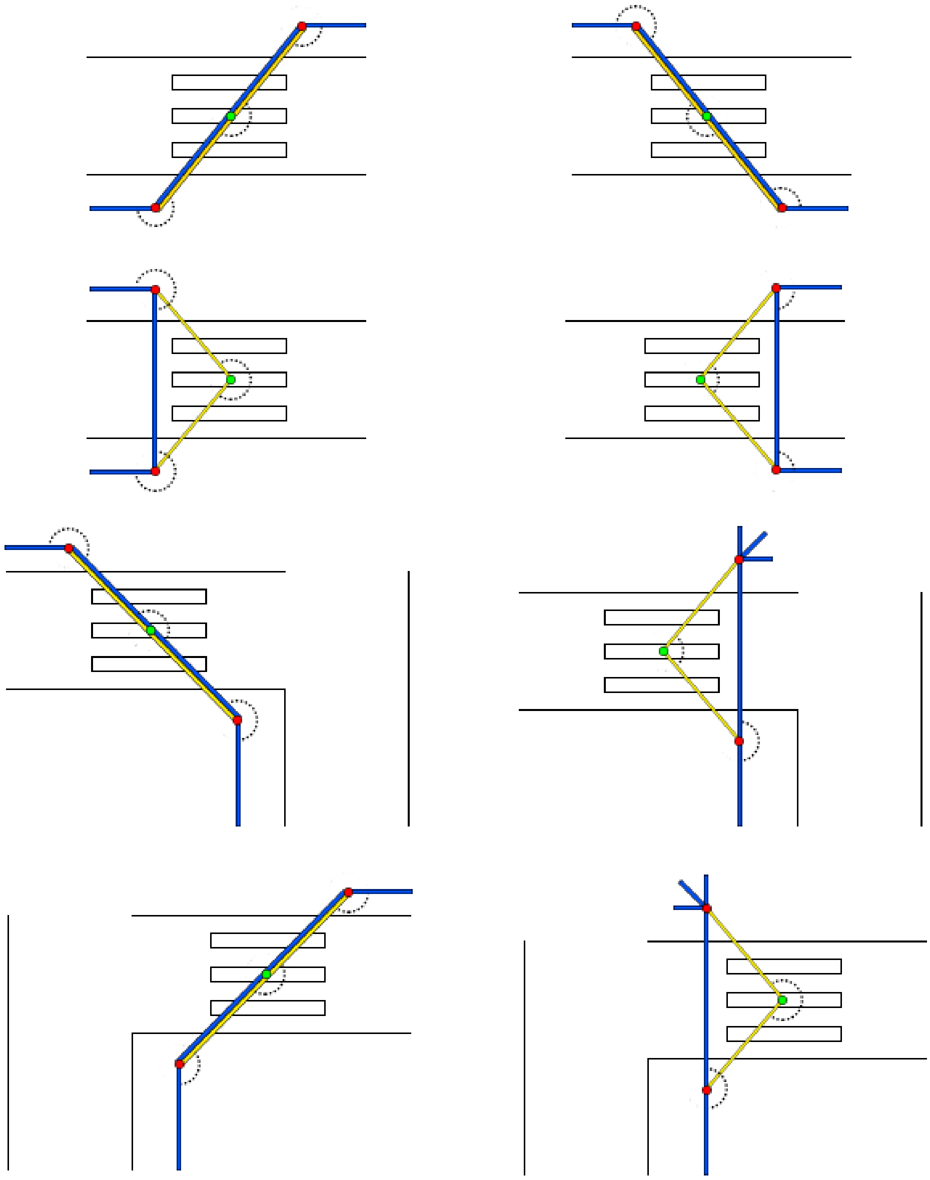


Figure 4.4: Presumed angles of crossing.

4.3.2 Subject's Approximated Path

The available data didn't have a continuous line of the subject's path, but instead, had spaced coordinate points. After arranging the data required in Subsection 4.1.6, it was implemented a Centripetal Catmull-Rom spline for each possible crossing event to create an approximated course taken by the subject, with the bounds provided. In essence, the Catmull-Rom spline is a geometric and parametric continuity line that ensures it contains the given points.

Point Angle 1	**Point Angle 2**	**Node Angle**	**Predicted Outcome**
160° to 200°	160° to 200°	30° to 120°	False
		remainder	Probably True
80° to 150°	80° to 150°	160° to 200°	True
		60° to 120°	True
	210° to 280°	160° to 200°	True
	remainder		Probably False
210° to 280°	80° to 150°	160° to 200°	True
		240° to 320°	True
	210° to 280°	160° to 200°	True
	remainder		Probably False
remainder			Probably False

Table 4.3: Presumed angles of crossing and the outcome of the model

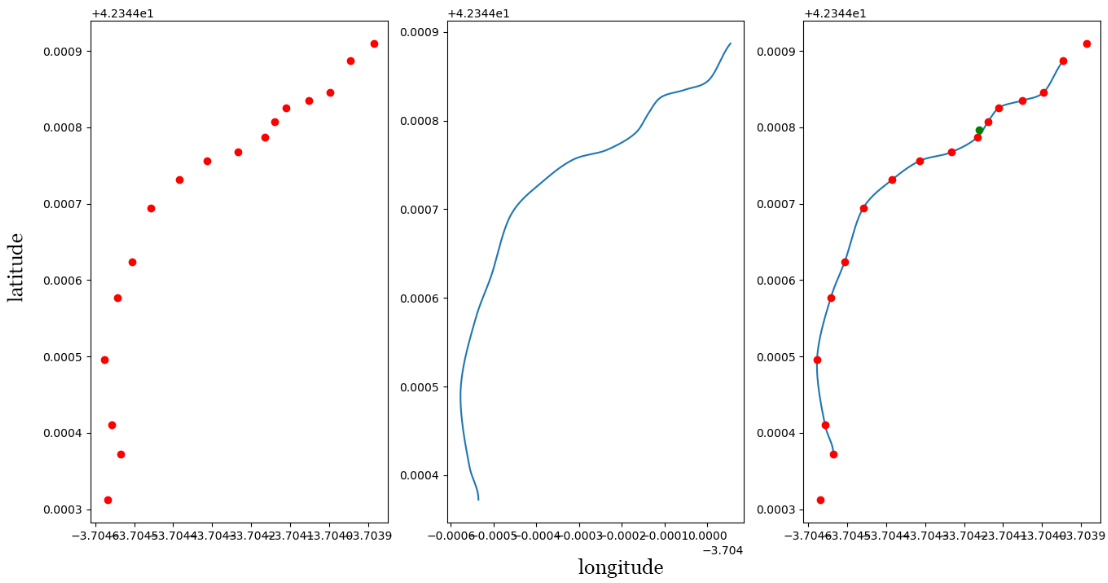


Figure 4.5: Plot of the Catmull-Rom spline with the control points.

To achieve the previously referred splines, the data coordinates were converted to the UTM coordinate system, using the utm[2] python package, calculated the spline points, and subsequently changed back to the geographic coordinate system of latitude and longitude. The conversion of coordinate systems minimized the inaccuracy and errors that could have been added to the splines as a result of the curvature of the Earth, despite the small distances that were worked with.

A python Jupyter Notebook[3] was developed with the folium python package, where a map

[2] `https://github.com/Turbo87/utm` (last visited on 01/07/2020)
[3] `https://jupyter.org/` (last visited on 01/07/2020)

is retrieved from OpenStreetMaps, and then displayed, along with the pedestrian crossing node point, the subject's stored data coordinates and the created spline for each crossing event, to see the obtained splines.

4.3.3 Heatmaps of Crossings to Map Different Types of Crossings

With the data normalized and with all pedestrian crossings oriented vertically in Section 4.1.7, the heatmaps could be created. To be able to classify the different crossings with the heatmaps, tags were retrieved from the OpenStreetMaps about the streets and pedestrian crossings, with new queries. Some tags were removed due to the reduced number of cases or for being unnecessary, such as the name of the street, that is irrelevant for heatmaps. Also, a problem that arose was the considerable amount of cases that didn't have values in some tags. That is a limitation from the use of this data source, which is inherent to the fact that OSM requires manual data entry.

With the available cases, pedestrian crossing's tags and street's tags, multiple heatmaps were produced using the seaborn[4] python package, with added regression lines of polynomial order 1 and another of up to order 5.

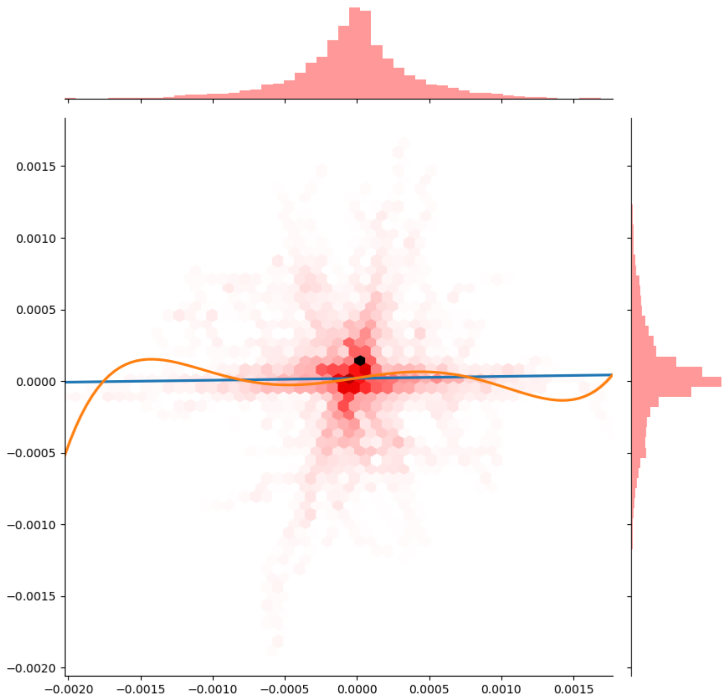


Figure 4.6: Heatmap of events when the pedestrian crossings has the value 'marked' for the tag 'crossing'.

As it is possible to assert from Figure 4.6, it was not possible to retrieve much perceptible information. Some possible problems could be the multiple pedestrian crossings nearby—although

[4]`https://seaborn.pydata.org/` (last visited on 01/07/2020)

just one of them was the correct one, there was no way to identify the correct one with the available data, so the data is repeated for each node—and the existence of multiple nodes for a single pedestrian crossing (e.g. when the road forks in a junction) for each crossing event. The length of the crossing event could also aggravate these situations.

4.3.4 Cluster Data in Different Crossing Events

The data used to cluster crossing events was previously prepared in Subsection 4.1.8, where the problems raised in Subsection 4.3.3 were fixed. With the data cleaned, a python Jupyter Notebook was then developed to implement clustering algorithms and visualize the results. The python package scikit-learn[5] made it possible to use the DBSCAN and K-Means clustering algorithms, by just making a function call to the respective package function with the data and some adjustable parameters of the algorithms. These parameters were changed until the results seemed reasonably similar to what they were expected to be.

It was also implemented another way to cluster the data. Considering a straight line between the first and last point of each crossing event, as a supposed direct path taken by the subject, it was possible to cluster, using the two algorithms mentioned above, the start and end points separately. Afterwards, the labels generated from the clustering were applied to the data prepared in Subsection 4.1.8. This way, it was possible to compare results between the two implementations and overcome the problem of DBSCAN when using the full extent of the data. This problem manifested because it clusters the data based on a distance parameter, which would either group everything with one label and noise or cluster it into too many labels[3, 4].

[5] `https://scikit-learn.org/stable/` (last visited on 01/07/2020)

Chapter 5

Results and Observations

The current chapter introduces the results and observations obtained throughout the implementation, as well as problems and questions that arose, and information acquired from these results.

5.1 Data Sample

The data used was from the SIMUSAFE project, as previously mentioned in Subsection 4.1.2. This sample contains data from a total of 10 subjects, distributed in 93 episodes. Through the data conversion to CSV files, some entries had missing values for components required for the next steps. Those entries were unaccounted and deleted. As it is possible to observe in Table 5.1, there were a total of 14563 deleted entries and the remaining entries amount to 278911470. The majority of the deleted entries came from the 3 episodes that were erased in its entirety.

Subject	**Episodes**	**Deleted Episodes**	**Final Episodes**	**Deleted Entries**	**Valid Entries**
B07M3	6	0	6	1	286229
B31M1	14	0	14	30	12839337
B061F1	5	0	5	0	3211220
P5M3	1	0	1	0	2518210
P005M3	4	0	4	30	6091520
P11M3	21	2	19	8862	40759830
P12M2	8	0	8	60	3904210
P15M4	13	0	13	0	43438170
P016M1	9	0	9	0	133531834
P140M5	12	1	11	5580	32330910
Total	93	3	90	14563	278911470

Table 5.1: Valid and deleted data entries of SIMUSAFE.

In the following bar chart (Figure 5.1), the disparity of the number of episodes per subject can be recognised. The variation in the average of valid entries per episode of each subject can also be perceived.

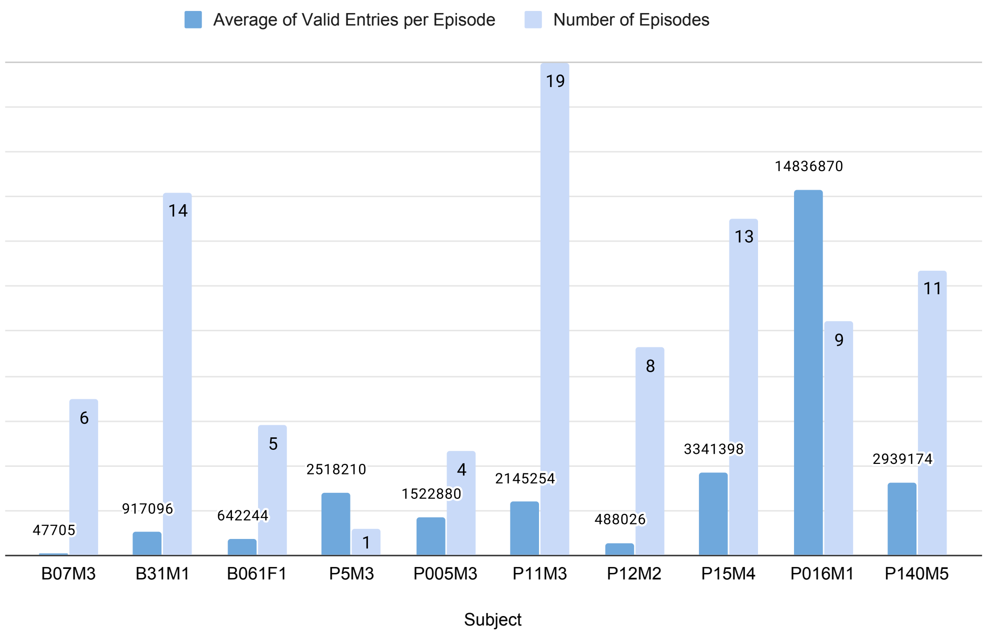


Figure 5.1: Bar chart of average valid entries per episode and number of episodes per subject.

5.2 Identification of Nearby Pedestrian Crossing

In this section, the data inside the converted CSVs had one entry for each frame of the video's episode. Since the data required for this part was only the latitude and longitude, which came from the phone' sensors, it had a considerable amount of redundant and repeated information for each unique pair of coordinates. Following the data cleansing in Subsection 4.1.5, a total of 278904332 rows of repeated information were removed, leaving 7138 entries of valid working data. It was also added some columns to the entries to compensate the deleted data. These columns regard the start and end timestamp, the referred file, the number of frames of that entry, as well as its starting frame. In Table 5.2, it's possible to see the average and median of the seconds and the number of frames that a pair of coordinates didn't change.

	average	**median**	**max value**	**min value**
seconds	5.39	5.57	92	0.03
number of frames	157.71	150	2760	1

Table 5.2: Variation of seconds and number of frames for a pair of coordinates

After the data preparation, it was the time to query the Overpass API as described in subsection 4.2.1, to find nearby pedestrian crossings. These queries' responses led to storing the information received about nearby pedestrian crossings in a file, which held the node id of the nearby pedestrian crossing, and its latitude and longitude. It stored 642 unique crossing nodes across all subjects and episodes.

This method is not the most suited to identify actual crossings made by the subject, considering that it finds all the pedestrian crossings in a designated radius around the subject. That means that the subject could have been only walking by the pavement and it recognises pedestrian crossings that the subject didn't cross. It also doesn't perceive cases where the pedestrian crosses outside of a designated pedestrian crossing, i.e. jaywalking. On the other hand, if in fact, the person crossed the street, the distance of the crossing to the crossing node, can be big enough that it requires a big radius, and therefore also find other pedestrian crossings when only walking by.

5.3 Timestamp intersection

From Subsection 4.2.2, the results on Table 5.3 were produced. It generated 79 rows, where the 'index_true_cross' is the index of the entry in the experts' annotations file, and the other indexes specify the entries in Table 4.2. The results are essentially a pivot table to inner join those two tables.

index_true_cross	starting_index	ending_index
0	1946	1968
1	1811	1827
2	1811	1831
4	1884	1899
...		
208	3554	3568

Table 5.3: Information resulting from timestamp intersection.

5.4 Simple Prediction Model Based on Angles

During the development of the prediction model in Subsection 4.3.1, it was at first to only to be used the angle of two coordinate data points with the crossing node (Figure 5.8). This approach led to not knowing if the subject crossed the road, e.g. if it was an acute angle of up to around 100, the person could have crossed and turned back after the crossing, going the same way it came, but on the other side of the road (going from Point 1 to Point 2). The subject could also just had passed by the pedestrian crossing, maintaining a straight path in the pavement (going from Point 2 to Point 3).

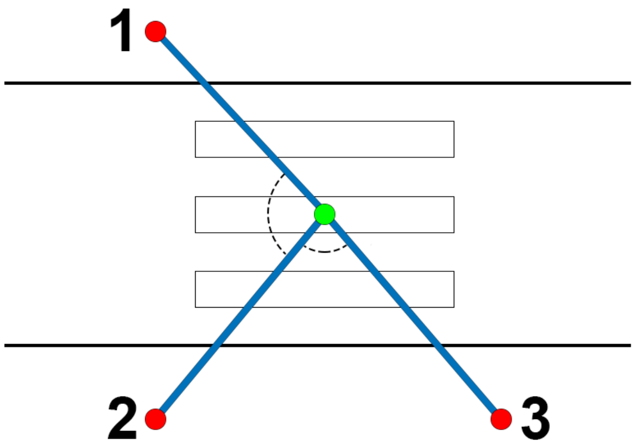


Figure 5.2: Representation of initial angle approach problem.

As a way to tackle this problem, it was added information about the angles of the subject's consecutive coordinate data points. With these angles it was created the model described in Subsection 4.3.1 and in Table 4.3. That also led to problems to classify the crossings. Lacking results identified some problems. The angle's approach was assuming a more fixed and linear position for the coordinate points, where the real data was more fluid and could fluctuate along the pedestrian crossing and the pavement, due to interactions with other road users or even the randomness associated with every action. Additionally, due to data inaccuracy, the subject's position could have some offset.

With the observations made about the angles' approach, it was assumed that more information would lead to more realistic models to classify the crossings, since adding more points and angles helped to create this naive and rudimentary method. Knowing that the full path taken by the subject could increase the information, it led to the next step of the implementation in Subsection 4.3.2.

5.5 Subject's Approximated Path

In Subsection 4.3.2, the approximated path of the subject was created, as well as it was implemented multiple ways to preview these results. In Figure 5.3 we can observe a plot of the control points, in red, used to generate the spline, in blue. These control points were the data coordinates recorded from the pedestrian. The green points are the pedestrian crossings nearby.

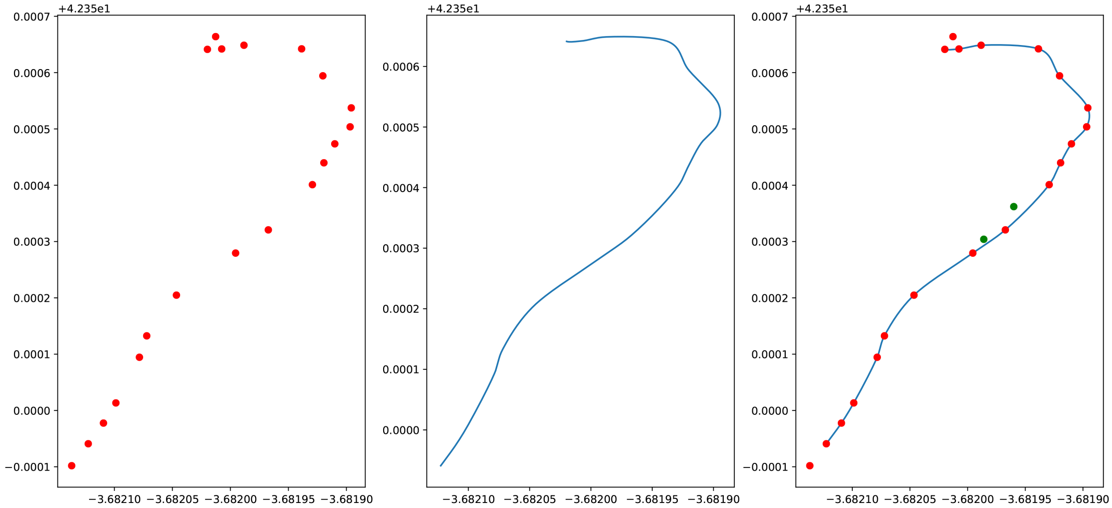


Figure 5.3: Plot of the spline with the control points.

To improve the readability of the plot and the understanding of the information in the spline, it was charted against a map. As it was possible to view in Figure 5.4, it was perceived more context about the path taken by the subject as well as the possible inaccuracy in the data. The brown circles are the radius of 15 meters around the pedestrian crossings. It also shows how it clearly becomes important to know the orientation of the pedestrian crossing in relation to the pedestrian's movement, to understand if the pedestrian is trying to cross or just walking beside them, which is difficult to recognise just by looking at the Figure 5.3.

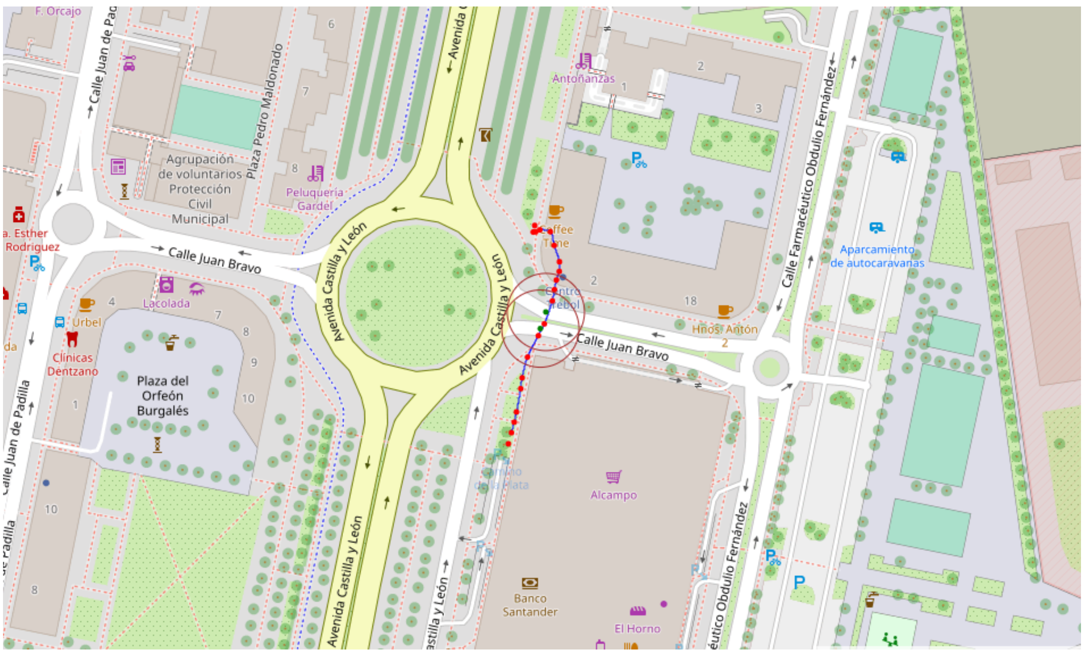


Figure 5.4: Plot of the spline in a map from OSM using folium.

5.6 Heatmaps of Crossings to Map Different Types of Crossings

After the creation of the approximated path of the pedestrian, it was combined all splines into one plot (see Figure 5.5). Although it may have seemed abstract, with no information to retrieve, it helped in finding the next step: to further improve it, into a heatmap. This way the information should have been able to be extracted.

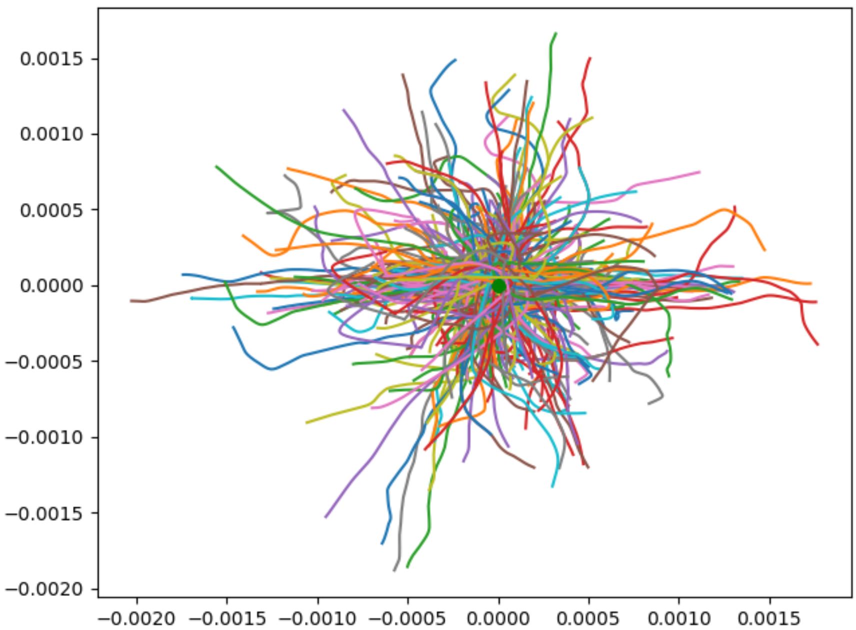


Figure 5.5: Plot of all splines together.

In Subsection 4.3.3 the code to generate the heatmaps was implemented, as well as its preview. The result is viewable in Figures 4.6 and 5.6. Despite the considerable improvement, it was still not possible to obtain much discernible information when comparing to Figure 5.5.

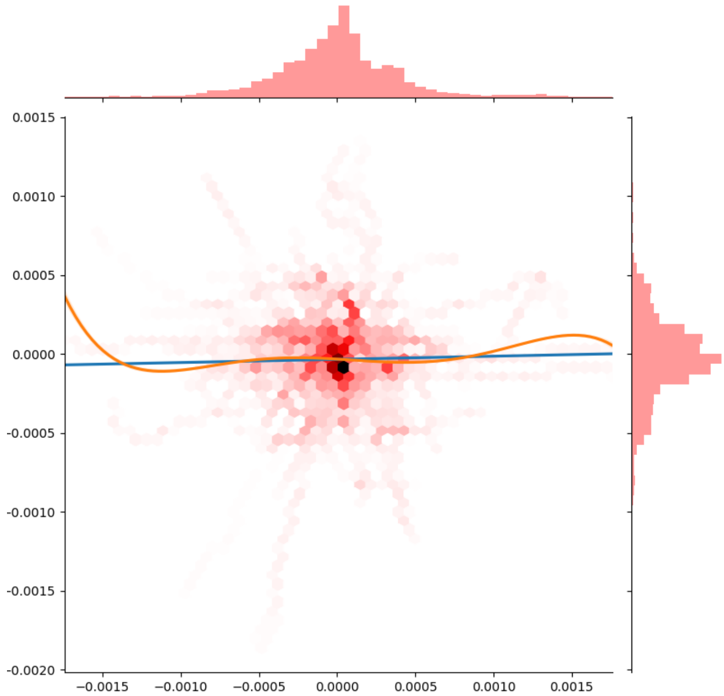


Figure 5.6: Heatmap of events when the pedestrian crossings has the value 'traffic_signals' for the tag 'crossing'.

Some problems for this overflow in the concentration of data points were: the multiple pedestrian crossings nearby when, in reality, only one of those was the actual one that was crossed; the presence of multi-node pedestrian crossing (e.g. when the road forks in a junction, or sometimes in pedestrian crossings with island); and the length of the data for the crossing event, sometimes even caught other events, displaying repeated information.

5.7 Cluster Data with different Clustering Algorithms

To cluster the data with different clustering algorithms in Subsection 4.3.4, it was necessary to start with the data preparation, which was specified in Subsection 4.1.8. The data produced can be observed firstly in Figure 5.7, where entries from the approximated path spline were combined in-between the pedestrian's recorded coordinates, and lastly in Figure 5.8, where points from the spline were added to the beginning and end of the results obtained in Figure 5.7 while maintaining the 15-meter radius.

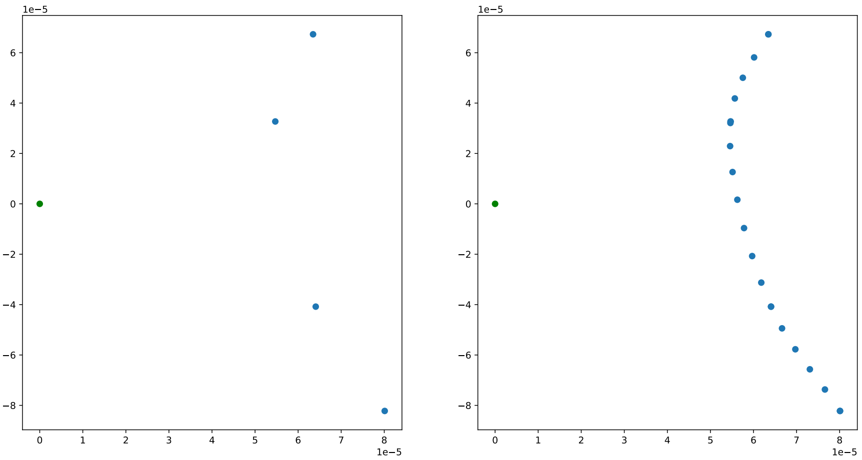


Figure 5.7: Before and after adding extra points from the spline to the subject's coordinates.

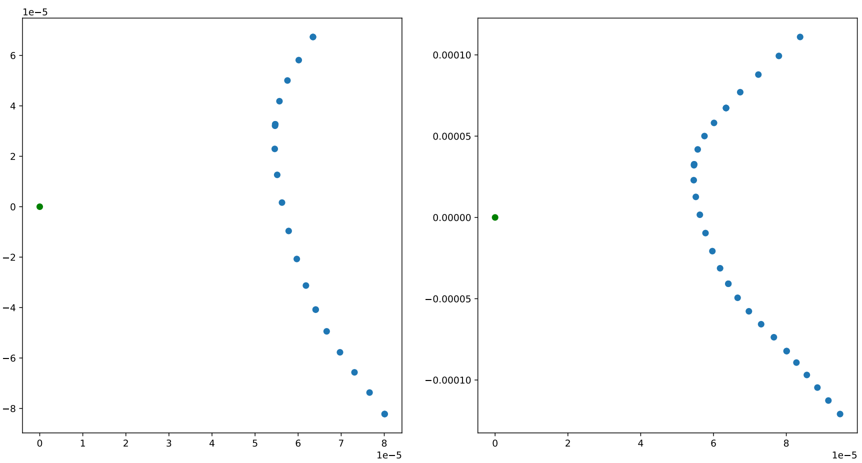


Figure 5.8: Before and after adding extra points from the spline to the beginning and ending of Figure 5.7.

With the generated data and with the implementation of DBSCAN it produced Figure 5.9. Even with multiple parameter tweaks, the DBSCAN approach was not appropriate for the data. Although it detects a heavy centre of density (in red), as well as some possible cases of entrance and exiting of the pedestrian crossing (in the greens, yellows and orange), it bases the clustering on a distance parameter in pair with a minimum number of surrounding points. That led to either agglomerate almost everything within the centre or separate it in an extensive number of labels.

With the K-Means algorithm, it generated Figure 5.10a. By overlaying a supposedly pedestrian crossing on top of Figure 5.10a, obtaining Figure 5.10b.

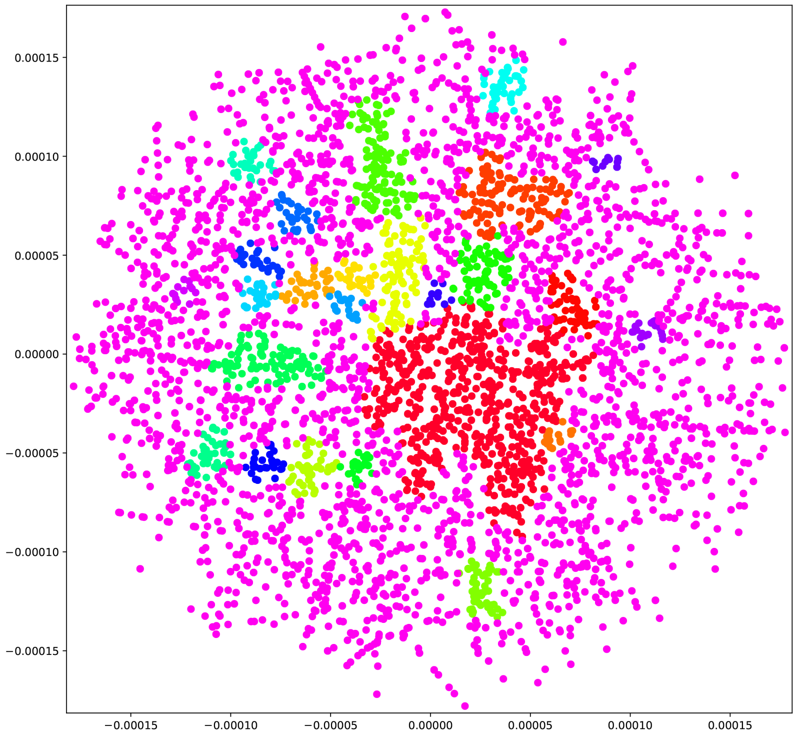
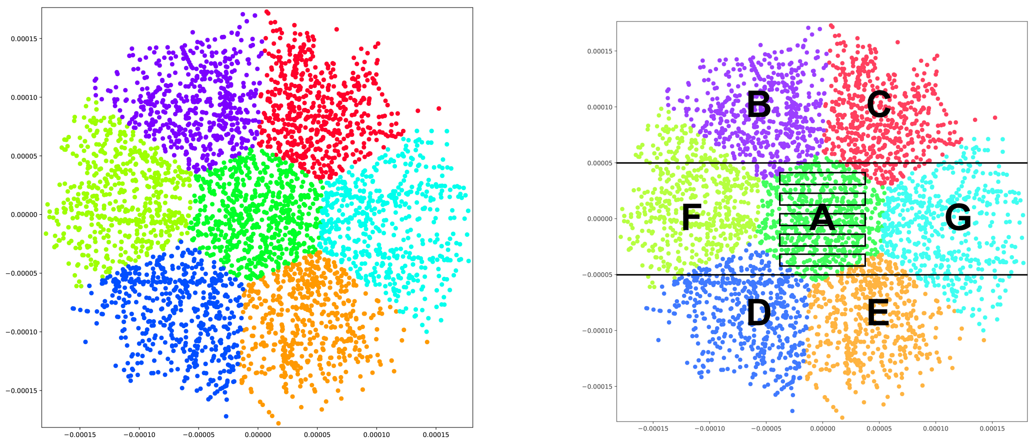


Figure 5.9: DBSCAN clustering. Pink is noise.



(a) Original

(b) With pedestrian crossing overlay

Figure 5.10: K-Means clusters.

With these results, it was possible to observe distinctive clusters that match some assumptions about crossing events. The cluster centre, in A, representing every crossing inside the pedestrian crossing, and where all paths cross. It was possible to detect the more normal and expected ways of crossing, from D to either B or C, from E to either B or C, or with the reverse direction, from B to D or E and from C to D or E. It was also possible to detect what was considered a more diagonal crossing, where a pedestrian starts before the markings of the pedestrian crossing and ends after them. Starting on F or G and ending on the other farthest points such as C, G and E, or on B, F and D respectively, show an even more extreme crossing which enters on illegal crossings. These cases were also interpreted as possible cases of misidentification of crossings and as to where the data used may be inaccurate and therefore were considered as probable noise.

The other approach implemented was to use only the start and end points from the spline, creating a direct line between them. In Figure 5.11 it's possible to observe the differences of its representation.

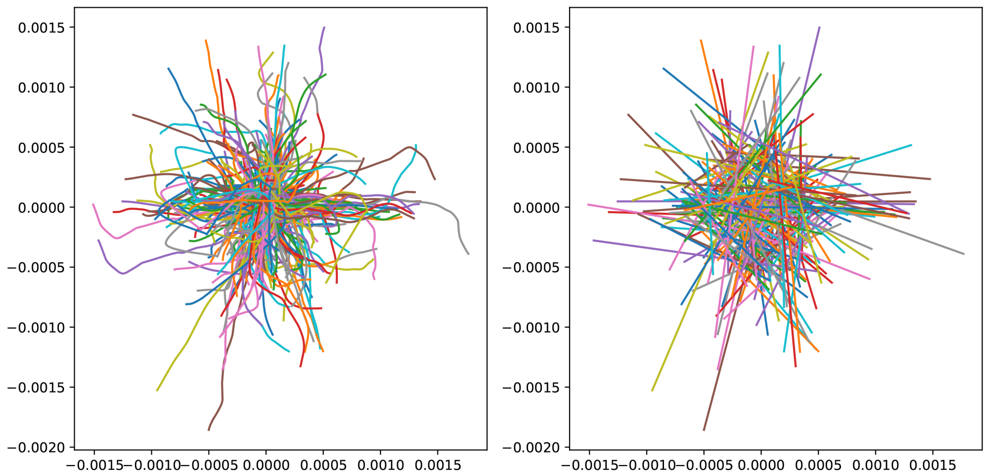
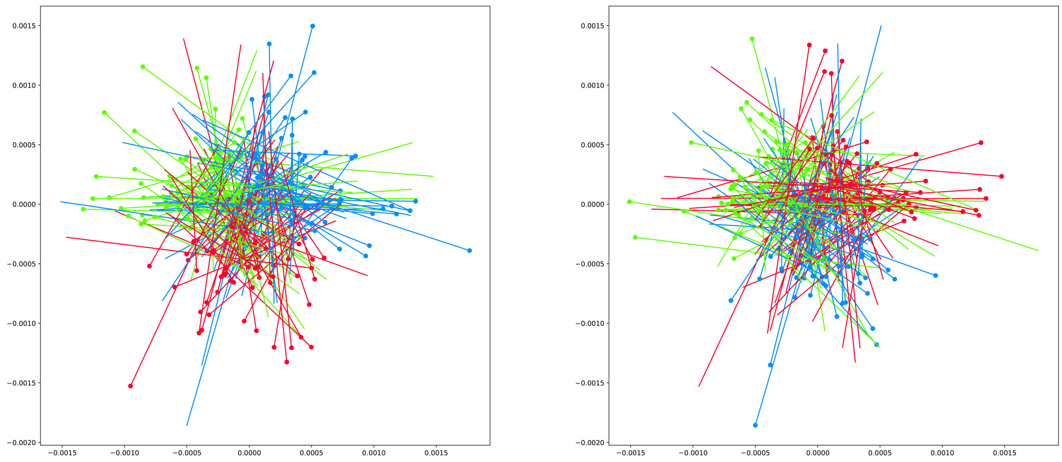


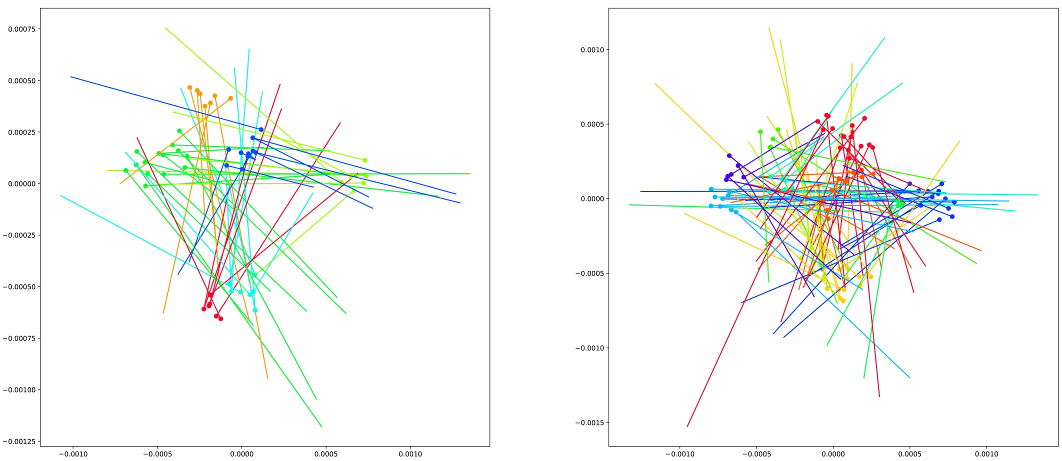
Figure 5.11: Representation of splines only using start and end points.

While with K-Means algorithm it was used a controlled number of clusters, resulting in 3 clusters for both starting points (Figure 5.12a) and ending points (Figure 5.12b), with DBSCAN, it was required to use an automatic recognition by the algorithm, resulting in 7 clusters for starting points (Figure 5.13a) and 11 for the ending points (Figure 5.13b).



(a) Starting points **(b)** Ending points

Figure 5.12: K-Means clustering with only starting and ending points.

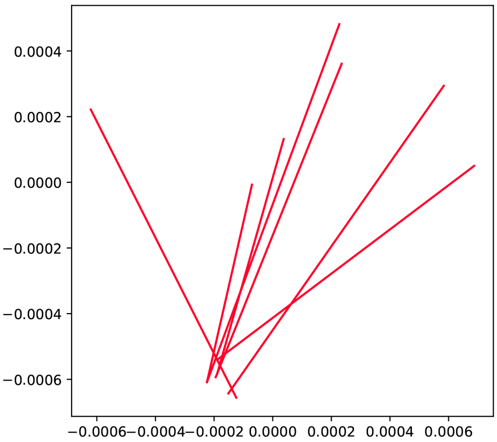


(a) Starting points **(b)** Ending points

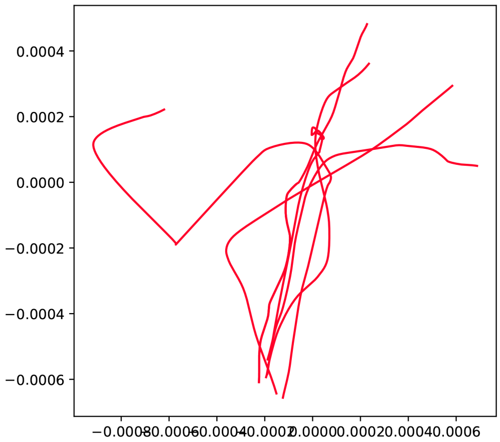
Figure 5.13: DBSCAN clustering with only starting and ending points.

The resulting labels were applied to the splines, obtaining some promising outcomes in some cases, as it is possible to observe in Figure 5.14, where seems to be noticeable more orthogonal crossings, that from Figure 5.10b fits in the D-A-C clusters.

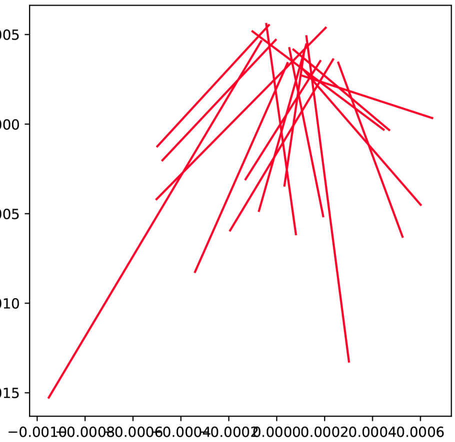
With these results, it's possible to cross-validate the outcomes obtained from clustering the points prepared in Subsection 4.3.4 with the K-Means algorithm (Figure 5.10) and propose a taxonomy to identify different types of crossings and the events in crossing. This could be future used to predict decisions of pedestrians, anticipating its route, in crossing related moments.



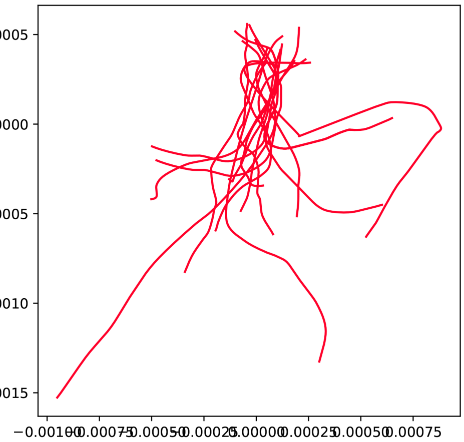
(a) Starting points



(b) Label applied on spline



(c) Ending points



(d) Label applied on spline

Figure 5.14: Example of results with label 0 of DBSCAN clustering with only starting and ending points.

5.8 Discussion

With the whole solution implemented and results obtained throughout the entire process leading to the outcomes of Section 5.7, it's possible to characterize what is a crossing event. A crossing event can be depicted as the act of a pedestrian crossing the street in a 3-step event. A pre-crossing moment preceding the moment where the pedestrian enters the street. A post-crossing one, when the pedestrian returns from the street to the pavement. And finally, the crossing moment, where the pedestrian is in the street. This is identified in Figure 5.10b where it's possible to assign the clusters into these 3 phases. The main part of F, A and G clusters coincide with the crossing phase, as they are in the road, especially A who overlaps the pedestrian crossing. The B, C, D and E clusters match with the pre and post-crossing phases. These are zones where the person approaches the pedestrian crossing with the intent to cross, as well as areas which the pedestrian leaves the pedestrian crossing. The paths took by the pedestrians in Figure 5.14 validate these assertions. It's possible to verify cases where the starting points are in the pre-crossing phase D, continue through the middle section A, the crossing part, and finish on the other side of the street on either B or C. The Figures 5.12 and 5.13 likewise prove the same as above, and when overlapped with the presumed types of crossing in Figure 4.4, these are easily identified. As such, a taxonomy for crossing events is created.

The aforementioned achieved the final objective of the dissertation, **Create a taxonomy for crossing events**. With this ontology is a step further in predicting human behaviour. This can bring further improvements and technology advances as it will influence simulations made in virtual environments. These simulation test changes to road networks where the pedestrians can be affected and prevent real-world problems beforehand without endangering lives. This can also be applied to autonomous driving, where the system may predict a possible collision with the pedestrian route based on the type of crossing and the phase of crossing event, and apply effective measures to avoid an accident to both road users. These are some possible applications of a vast sea of opportunities.

The other objectives were also completed along the way:

- **Identify information sources and metrics**. This was done in Section 3.1 where the process of what types of data to use, and where to find the sources is described. The implementation of this process further confirms this in Sections 4.1.1 and 4.1.2.

- **Identify possible crossing events**. When it was used the Overpass API to collect pedestrian crossing nodes and the timestamp intersection with the experts' annotations, in Subsections 4.2.1 and 4.2.2 respectively, this was achieved. Both outputted possible crossing events. The first was based on distance and nearby pedestrian crossings, while the second was with the addition of time intervals.

- **Develop the pedestrian behaviour model on times of road crossing to analyse and classify possible crossing events**. Multiple models were created, although the last from Subsection 4.3.4, achieved the end result so it was possible to analyse a classify the events.

Chapter 6

Conclusion

With this dissertation, it's proposed a new and different data-agnostic methodology process to model the pedestrian behaviour in times of crossing. This procedure, based on geographical data, identifies distinct pedestrian crossing events. It's conjectured the necessity in separating crossing events, as it may provide comprehension related to the pedestrian behaviour in these situations.

This work recognizes possible crossing events by detecting nearby pedestrian crossings in a specified range (in this case it was used a 15-meter radius) of a latitude and longitude dataset. Subsequently, the information retrieved is used in an automatic machine learning predictor to divide and label into different types and parts of events of crossings.

It's speculated that there are certain types of crossings, such as a more central crossing, where the pedestrian uses only the midway section of a pedestrian crossing, as well as a more diagonal approach, in which the crossing starts and ends in the road section, cutting through the centre of the crossing, attaining more risk. It's also presumed the existence of distinctive events of crossing.

The presumptions were validated and a taxonomy for crossing events was created in cases of crossing in or nearby a pedestrian crossing (Section 5.8). This taxonomy states the existence of three phases during the cross. A pre-crossing moment preceding the moment where the pedestrian enters the street. A post-crossing one, when the pedestrian returns from the street to the pavement. And finally, the crossing moment, where the pedestrian is in the street.

Whereas the work designed and implemented the aforementioned taxonomy, the data used from SIMUSAFE's project lacked to interpret and reason the entirety of the clusters obtained with the experiences (see Section 6.1 for more in-depth limitations).

This study launches new research opportunities to improve the results obtained and to enhance the assigned ontology (view in Section 6.2 future improvements).

6.1 Future Work

In section 5.7, it was created a way to analyse and model possible crossing events based on personal assumptions. Although the objectives of this work were completed, there is still room for improvements. The crucial next step is to develop a classifier which would outcome a prediction if the pedestrian was going to cross the road, and, if so, to where the subject was going to cross to, based on the data given. The model could also be improved and tested with other clustering algorithms, to contribute to evaluating and verifying the current model, as well as to enhance it with more specific and divisible events of the crossing.

It also could be tested with newly added information. If new context data was perceived from the videos, it could further improve the classification and prediction of a pedestrians path. For example, if it was retrieved contextual information of pedestrians nearby, that could explain certain deviations that the subject performed while crossing the road.

6.2 Limitations and Challenges

The results obtained derive from the way it was implemented and from the data used. Other implementations and other datasets may hold different results, which could have more suitable outcomes. Using the radius of 15 meters to find nearby pedestrian crossings may have impacted the number of false-positive crossings found but was also due to inaccuracy of the GPS data used. If it were used military-grade sensors to gather the data, a smaller radius could have been more appropriate in that case. Also, the sparsity of coordinate points recorded as well as its offset—derived from the accuracy and precision of the sensors used—left a slight necessity for better data. The clustering methods implemented also have their limitations, e.g. using DBSCAN with more split data and defined events should perform better. Of course, using other methods of clustering could bring better or worse results. These examples add some limitations of the methods used.

One thing that appeared as a challenge was the lack of relevant information about the context of the pedestrian. In the beginning, when first analysing the data, it was expected to have some context, especially about seen pedestrian crossings in the videos recorded, but that information was not yet been collected. This challenge was overcome by identifying pedestrian crossings with other means (Section 4.2), that could have been only used to validate the referred wanted information.

References

[1] MPC Alice. https://www.mpc-rnd.com/technology/alice/. Accessed: 2020-02-06.

[2] Krakow Simusafe Horizon 2020 Grant Information. https://www.aptiv.com/docs/default-source/european-grants/krakow-simusafe-horizon-2020-grant-information.pdf?sfvrsn=65e9683e_2. Accessed: 2020-02-06.

[3] Clustering – scikit-learn 0.23.1 documentation. https://scikit-learn.org/stable/modules/clustering.html#dbscan, . Accessed: 2020-07-01.

[4] Comparing Python Clustering Algorithms – hdbscan 0.8.1 documentation. https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html#dbscan, . Accessed: 2020-07-01.

[5] Crowd Simulation, Layout & Previz tools for Maya | Golaem. http://golaem.com/, . Accessed: 2020-02-06.

[6] City and Public Spaces | Golaem. http://golaem.com/content/usages/ambient, . Accessed: 2020-07-04.

[7] Massive Software – Simulating Life. http://www.massivesoftware.com/. Accessed: 2020-02-06.

[8] Massive Software 7.0. http://www.massivesoftware.com/massive7.0-press-release.html. Accessed: 2020-07-04.

[9] OpenStreetMap Wiki. https://wiki.openstreetmap.org/wiki/Main_Page. Accessed: 2020-06-30.

[10] Overpass API - OpenStreetMap Wiki. https://wiki.openstreetmap.org/wiki/Overpass_API. Accessed: 2020-06-30.

[11] PTV Solutions. http://vision-traffic.ptvgroup.com/en-us/products/, . Accessed: 2020-02-06.

[12] PTV Viswalk - Realistic Pedestrian Simulation Software | PTV Group. https://www.ptvgroup.com/en/solutions/products/ptv-viswalk/, . Accessed: 2020-02-06.

[13] Crowd Simulation for Events, Evacuations & Stations | PTV Group. https://www.ptvgroup.com/en/solutions/products/ptv-viswalk/crowd-simulation/, . Accessed: 2020-07-04.

[14] Quadstone Paramics - Traffic Microsimulation Software. http://www.paramics-online.com/, . Accessed: 2020-02-06.

[15] Pedestrains in Street Space | Quadstone Paramics. http://www.paramics-online.com/pedestrian-simulation.php, . Accessed: 2020-07-04.

[16] Background: Machine Learning - Unity-Technologies/ml-agents. https://github.com/Unity-Technologies/ml-agents/blob/release_3_docs/docs/Background-Machine-Learning.md. Accessed: 2020-07-04.

[17] SmartLab AI. A brief overview of Imitation Learning. https://medium.com/@SmartLabAI/a-brief-overview-of-imitation-learning-8a8a75c44a9c. Accessed: 2020-07-04.

[18] M Batty and B Jiang. New approaches to exploring space-time dynamics within gis. Technical report, CASA Working Paper 10 Multi-Agent Simulation, 1999.

[19] Gene Bellinger, Durval Castro, and Anthony Mills. Data, information, knowledge, and wisdom. 2004.

[20] Aude Billard and Daniel Grollman. *Imitation Learning in Robots*, pages 1494–1496. Springer US, Boston, MA, 2012. ISBN 978-1-4419-1428-6. doi: 10.1007/978-1-4419-1428-6_758. URL https://doi.org/10.1007/978-1-4419-1428-6_758.

[21] Josep Castellà and Jorge Pérez. Sensitivity to punishment and sensitivity to reward and traffic violations. *Accident Analysis & Prevention*, 36(6):947 – 952, 2004. ISSN 0001-4575. doi: https://doi.org/10.1016/j.aap.2003.10.003. URL http://www.sciencedirect.com/science/article/pii/S0001457503001532.

[22] P. Chance. *Introduction to Educational Leadership & Organizational Behavior*. Taylor & Francis, 2013. ISBN 9781317923176. URL https://books.google.pl/books?id=ZWXfAQAAQBAJ.

[23] Jonathan Cinnamon, Nadine Schuurman, and S. Morad Hameed. Pedestrian injury and human behaviour: observing road-rule violations at high-incident intersections. *PloS one*, 6(6):e21063–e21063, 2011. ISSN 1932-6203. doi: 10.1371/journal.pone.0021063. URL https://www.ncbi.nlm.nih.gov/pubmed/21698258. 21698258[pmid].

[24] Emilio Moyano Díaz. Theory of planned behavior and pedestrians' intentions to violate traffic regulations. *Transportation Research Part F: Traffic Psychology and Behaviour*, 5(3):169 – 175, 2002. ISSN 1369-8478. doi: https://doi.org/10.1016/S1369-8478(02)00015-3. URL http://www.sciencedirect.com/science/article/pii/S1369847802000153.

[25] James Elander, Robert West, and Davina French. Behavioral correlates of individual differences in road-traffic crash risk: An examination of methods and findings. *Psychological Bulletin*, 113(2):279–294, 1993. doi: 10.1037/0033-2909.113.2.279.

[26] H. Guo, Z. Gao, X. Yang, and X. Jiang. Modeling pedestrian violation behavior at signalized crosswalks in china: A hazards-based duration approach. *Traffic Injury Prevention*, 12(1): 96–103, 2011. doi: 10.1080/15389588.2010.518652. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-78951475823&doi=10.1080%2f15389588.2010.518652&partnerID=40&md5=e4ab6d93a44a9bd4ca66ca4b7430b86e.

[27] André Hanisch, Juri Tolujew, Klaus Richter, and Thomas Schulze. Modeling people flow: online simulation of pedestrian flow in public buildings. In *Proceedings of the 35th conference on Winter simulation: driving innovation*, pages 1635–1641, 2003.

[28] Colin M. Henein and Tony White. Agent-based modelling of forces in crowds. In Paul Davidsson, Brian Logan, and Keiki Takadama, editors, *Multi-Agent and Multi-Agent-Based Simulation*, pages 173–184. Springer Berlin Heidelberg, 2005.

[29] Richard M. Lerner, Jerome Kagan, and Marc H. Bornstein. Human behaviour. https://www.britannica.com/topic/human-behavior, Feb 2019. URL https://www.britannica.com/topic/human-behavior. (last visited on 25/01/2020).

[30] B. Lewis, R. Jurmain, and L. Kilgore. *Cengage Advantage Books: Understanding Humans: An Introduction to Physical Anthropology and Archaeology*. Cengage advantage books. Cengage Learning, 2008. ISBN 9780495604747. URL https://books.google.pl/books?id=Z6dg1FPViKEC.

[31] Gunnar G Løvås. Modeling and simulation of pedestrian traffic flow. *Transportation Research Part B: Methodological*, 28(6):429–443, 1994.

[32] Linbo Luo, Suiping Zhou, Wentong Cai, Malcolm Yoke Hean Low, Feng Tian, Yongwei Wang, Xian Xiao, and Dan Chen. Agent-based human behavior modeling for crowd simulation. *Computer Animation and Virtual Worlds*, 19(3-4):271–281, 2008.

[33] Letizia Nicoletti, Antonio Padovano, Francesco Pasquale Vartuli, and Marco Vetrano. Human behavior modeling: a state of the art. 2016.

[34] Xiaoshan Pan. *Computational modeling of human and social behaviors for emergency egress analysis*. PhD thesis, Stanford University Stanford, CA, USA, 2006.

[35] Natalia Díaz Rodríguez, Manuel P. Cuéllar, Johan Lilius, and Miguel Delgado Calvo-Flores. A fuzzy ontology for semantic modelling and recognition of human behaviour. *Knowledge-Based Systems*, 66:46 – 60, 2014. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2014.04.016. URL http://www.sciencedirect.com/science/article/pii/S0950705114001385.

[36] Khaled Shaaban and Karim Abdel-Warith. Agent-based modeling of pedestrian behavior at an unmarked midblock crossing. *Procedia Computer Science*, 109:26 – 33, 2017. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2017.05.291. URL http://www.sciencedirect.com/science/article/pii/S1877050917309481.

[37] Nirajan Shiwakoti, Majid Sarvi, and Geoff Rose. Modelling pedestrian behaviour under emergency conditions–state-of-the-art and future directions. In *31st Australasian Transport Research Forum (ATRF)*, pages 457–473, 2008.

[38] M. West. *Developing High Quality Data Models*. Elsevier Science, 2011. ISBN 9780123751072. URL https://books.google.pl/books?id=y_gSxWkCXtYC.

[39] S. Yi, H. Li, and X. Wang. Pedestrian behavior modeling from stationary crowds with applications to intelligent surveillance. *IEEE Transactions on Image Processing*, 25(9): 4354–4368, 2016. doi: 10.1109/TIP.2016.2590322. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-84984876309&doi=10.1109%2fTIP.2016.2590322&partnerID=40&md5=e6af111313742502e427466254369888.

[40] Weiliang Zeng, Peng Chen, Hideki Nakamura, and Miho Iryo-Asano. Application of social force model to pedestrian behavior analysis at signalized crosswalk. *Transportation Research Part C: Emerging Technologies*, 40:143–159, 2014.

[41] L Åberg. Traffic rules and traffic safety. *Safety Science*, 29(3):205 – 215, 1998. ISSN 0925-7535. doi: https://doi.org/10.1016/S0925-7535(98)00023-X. URL http://www.sciencedirect.com/science/article/pii/S092575359800023X.