Small or medium-scale focused research project (STREP)

ICT Call 8

FP7-ICT-2011-8

Cooperative Self-Organizing System for low Carbon Mobility at low Penetration Rates

COLOMBO: Deliverable 5.2
Traffic Simulation Extensions

Document Information	
Title	Deliverable 5.2 - Traffic Simulation Extensions
Dissemination Level	PU (Public)
Version	1.0
Date	07.05.2014
Status	Final version

Authors	Jakob Erdmann (DLR), Wolfgang Niebel (DLR), Daniel Krajzewicz (DLR), Jérôme Härri (EURECOM), Saleh Bensator (EURECOM)

Table of contents

1 Introduction

The goal of the COLOMBO project is to deliver a set of modern traffic management solutions, all based on information gained from wireless communication performed between traffic participants and between traffic participants and infrastructure (V2X). The applications developed in COLOMBO use information obtained from wireless communication to gain information about the state of traffic, to give traffic participants further information, to control them via traffic lights, and to advise them to perform certain actions for being more environment-friendly.

Nowadays, most of research on vehicular communications concentrates on motorized on-road vehicles. But road traffic, as investigated, is formed by participants of different kind. The inclusion of further modes has several motivations within COLOMBO. One of the first to name is the objective to deliver solutions that reduce traffic's ecological impact. Besides other measures, one attempt towards achieving this goal is the prioritization of "soft", unmotorized modes of transport, mainly pedestrian walks or using a bicycle, making them more attractive to use. The work on traffic light systems involves the development of such prioritization methods, located in COLOMBO's work package 2.

Within COLOMBO's work package 1, pedestrians are assumed to play an important role for traffic surveillance. While exploiting low equipment rates of V2X, COLOMBO incorporates mobile devices such as PDAs that are carried by pedestrians and which are capable to communicate via Bluetooth or WiFi-Direct. COLOMBO's work package 1 will evaluate whether these additional communication devices can be used as further sensors, communication relay nodes, and computational devices.

COLOMBO uses a simulation system that allows to evaluate the developed solutions' benefits. Given the tasks above, this simulation system must be capable to replicate the behaviour of pedestrians and bicycles in a given simulation scenario. The major part of the simulation system that is has to be extended is the used traffic simulation SUMO. With the begin of COLOMBO, SUMO already included a pedestrian model that allows to define inter-modal trip chains consisting of "walking", "driving", and "waiting", but no pedestrian walking dynamics were covered; when walking, a person was moved along its route with a static speed and "jumped" over intersections. The work performed in COLOMBO – described in the following chapters – extended this "walking" mode by opening SUMO to include pedestrian dynamics models which determine a pedestrian's behaviour in a two-dimensional plane using discrete time steps. Two pedestrian dynamics models were included. The first resembles the originally used static speeds with no lateral deviations, the second divides pedestrian lanes into stripes and models (inter-pedestrian) collision avoidance and speed adaptation. The implementation regards the influence of traffic lights as well as the interactions between motorized traffic, bicycles, and pedestrians at uncontrolled and controlled intersections. Additionally, a standalone application that realizes pedestrian dynamics simulation was implemented.

Including pedestrians and bicycles into SUMO will as well enlarge their applicability possibilities beyond the scope of COLOMBO. The inter-modal trip chains are now capable to replicate all modes of transport on a really microscopic base, allowing to benchmark and compare them. Being more vulnerable than vehicles, pedestrians and bicycles as well got into the focus of traffic safety. The currently available models for both vehicle and pedestrian traffic are not yet fine-grained enough to answer traffic safety research questions well, but the inclusion of pedestrians into a traffic simulation is assumed to be an important step towards obtaining new, better models.

At last, one should point out that the inclusion of pedestrians – and the infrastructure (crossings) they use – influences the performance of motorized traffic as well. Therefore, the extensions not only extend SUMO's capabilities, but also improve its quality.

1.1 Motivation

Taking into account pedestrians and bicycles promises different benefits, all of which are at least well-usable if not necessary for further work performed in COLOMBO. Pedestrians as well as the infrastructure they use influence the behaviour of motorized traffic by introducing new capacity constraints as well as by offering additional halting place at intersections for left-moving vehicles. Therefore, their inclusion improves the quality of the simulation of motorized traffic. Extending the used traffic simulation SUMO by these transport modes completes the representation of multi-modal person-based trip chains by microscopic dynamics, a feature hardly to be found in other traffic simulations of this type.

Regarding COLOMBO's major scope, the implementation of traffic management solutions based on vehicular communications, pedestrians play a significant role at different levels as well. Pedestrians wearing a mobile communication device (such as PDAs) are assumed to be usable as relay nodes that help in passing information between other participants and which may also be used as additional source of information. Incorporating pedestrians as further communication partners is therefore allocated within the development of V2X-based traffic surveillance systems in COLOMBO's work package 1. The traffic light systems developed in COLOMBO's WP2 are scheduled to take pedestrians and bicycles into account for making these mobility modes more attractive. Both work packages rely on a proper implementation of pedestrians and bicycles into the used simulation system.

1.2 Objectives

The major objective was to extend the used simulation system by models for pedestrian and bicycle traffic at intersections. This work affects mainly the used traffic simulation as it's the instance responsible for correctly replicating traffic. Other components of the simulation system, such as the communication simulation, get information from the traffic simulation and – albeit special cases may arise – treat the obtained information similar to vehicles, which from their point of view are communication nodes only.

1.3 Structure

The document is structured as following. In chapter 2 the current state of the art in modelling pedestrians and bicycles is outlined. Chapter 3 describes how the used traffic simulation SUMO was extended by these modes of transport and introduces the implemented standalone pedestrian dynamics simulator. Chapter 4 presents some initial evaluations of the implemented models and of the effects of pedestrians and bicycles on motorized traffic. The documents ends with a summary given in chapter 5.

2 State of the Art in Pedestrian and Bicycles Modelling

2.1 Models and Simulation

Modelling and simulating pedestrians and bicycles as participants of traffic generally applies many of the same concepts and paradigms already established for motorized road traffic. Differentiations between macro-, meso-, and microscopic granularity can be made, as well as continuous vs. discrete time-space-representation[1] respectively (cf. COLOMBO D1.1 sect. 2.1).

Pedestrian models in particular can be categorized into the following classes according to [Saberi and Mahmassani, 2014], [Teknomo et al., 2000]:

- Macroscopic (aggregated flows)
  - Gas-kinetic models
  - Fluid based models
  - Continuum macroscopic models
- Microscopic (individuals)
  - Cellular automata based models
  - Force-based models, diverted into social force and magnetic force
  - Agent-based models
  - Queuing (network) model

This plurality of different models evolved over time, beginning in the 1970's [Henderson, 1971], but first researches date back as early as the 1930's. Some models were founded on analogies to natural phenomena and were improved by superposed extensions over the time. Early models are limited, and fail to build up various effects or whole situations, like *lane formation*, *group adherence*, *freeze-by-heat*, and *faster-is-slower*. Also the necessity for simple or compromising models with respect to low computing resources fades with more demanding and detailed models being no hindrance anymore.

Most of the models comprise not only movements on or alongside public roads. The majority of work even concentrates onto situations outside roads, like evacuation of people from civil engineering structures such as buildings, stadiums, and traffic stations, as well as the optimization of passenger flows in big public transport hubs (including airports). In comparison to bikes and motor vehicles, pedestrian movements are not limited to a longitudinal direction anymore, but are freely possible in an isotropic manner.

With respect to the task of COLOMBO to reproduce the microscopic interactions between pedestrians and road traffic, the following sections concentrate mainly on those aspects which are necessary for this purpose. In section 2.2 a description of a pedestrian's basic unimpeded behaviour is given, followed by the introduction into the above mentioned microscopic models in section 2.3. These models try to replicate the behaviour when obstacles like immobile structures or moving traffic participants impede the free movement. A special case of such interactions is the multimodal one when pedestrians cross the motorized vehicles' stream. This is described in section 2.4. The pedestrians' flows as the traffic demand to be modelled within a network can be derived in different manners as described in section 2.5.

The microscopic behaviour of bicycles is less present in literature and software. Bicycles and powered two-wheelers often share the infrastructure together with the motor vehicles and similarly have an anisotropic nature. Although they differ in some behavioural and technical aspects they are covered by – extended – established models for either motor traffic or pedestrians. They are presented in sect. 2.6.

[1] N.B.: Continuous models may still be discretized for calculation within simulations.

To set up a simulation for a particular model, its calibration with parameters and validation (cf. section 2.7) are necessary steps following the general guidelines for good practise in traffic simulation as laid down in, e.g., [Brackstone et al., 2014]. Meaningful scenario definitions and the final computation of carefully chosen performance indicators (PI) to evaluate and compare the scenarios as described in [COLOMBO D1.1 sect. 3 and 4] are needed as well.

2.2 Single Pedestrian Behaviour

A pedestrian is a walking person and can be described with some cognitive and basic physical parameters. The cognitive parameter is connected with the individual's decision making within a hierarchical scheme of strategic, tactical and operational levels. The destination and mode choice (car, bike, walk, etc.) are made on the strategic level, as well as the coarse "macro[2] route choice" between the origin and the destination of the trip. The detailed "micro routing" including the choice of where to cross the road, which entry to use, or which lane to use on a walkway with multiple yet imaginary lanes is of tactical nature. Instantaneous decisions while walking and interacting with others (cf. section 2.3) are taken at the operational level [Daamen, 2004]. This can be the speed, direction changes, overtaking manoeuvers etc. These decisions depend, inter alia, on the knowledge about the pedestrians' environment outside their direct view field (behind humans or objects like corners). Depending on this knowledge they might chose different routes and speeds [Werberich et al., 2014].

The physical parameters can be the gender, age, body circumference, and attached objects like luggage or a pram. These parameters influence the maximum and desired speed, as well as the space which is occupied by the person. Investigations [Gorrini et al., 2014] show that the front zone of this space which the person tries to keep clear from obstacles grows with the walking speed. It is noteworthy in this context that cellular automata can have only one uniform speed and one uniform space for all different people due to the model constraints. In contrast to vehicular traffic the time needed for acceleration and deceleration is negligible.

Sources for crisp values which can be taken to parameterize the models, i.e., to calibrate them, can be found in section 2.6.

2.3 Multiple Pedestrians and Interactions

Pedestrians often walk in areas with constraints by immobile structures or moving traffic participants who impede the free movement. Interactions among multiple pedestrians lead to some self-organizing and collective effects that are: *bi-directional flow, lane formation, oscillations at bottlenecks, group adherence, freeze-by-heat*, and *faster-is-slower*. The following modern microscopic models try to replicate most of these phenomena, of which some are arising only under crowded circumstances.

2.3.1 Social Force Model

In the social forces model of [Helbing and Molnar, 1995] an individual is subject to long-range forces and his dynamics follow the equation of motion. Social force model describes pedestrian behaviour by so-called social force where interaction with environment and other people is explained by attractive and repulsive forces. This model uses Newton's equation to calculate the forces. The approach is based on the acceleration equation:

$$\frac{\mathrm{d}\,\vec{\mathrm{v}}}{\mathrm{dt}} = \overrightarrow{\mathrm{F(t)}} + \mathrm{fluctuations}$$

[2] Not to be mixed up with the category of macroscopic pedestrian models.

In this acceleration equation the fluctuation term takes into account random and unsystematic variations of behaviour and $\overrightarrow{F(t)}$ gathers the social forces influencing pedestrian.

In social force model, forces are firstly defined by physical concepts and then have been applied to pedestrian behaviours. It is more of a physical model than a behavioural one. On the contrary, behavioural models attempt to get mathematical representative formulas based on human characteristics. It was shown that algorithms for collision detection and avoidance can dominate the dynamics and mask the role of the repulsive force.

In [Mohcine, 2012] a broad insight into different issues of the force models is given. It has been proved that most of them describe pedestrian dynamics with the above mentioned collective phenomena qualitatively fairly well on the macroscopic scope, but unfortunately there are only poor quantitative descriptions of these phenomena. It remains unclear if they are able to describe individual walking behaviour accurately as well.

The first problem that is inherent in force-based models is the Newton's third law. According to this principle two particles interact by forces of equal magnitudes and opposite directions. For pedestrians this law is unrealistic since in general a pedestrian does not react to pedestrians behind, even if the angle of vision is taken into account. In this context the forces mutually exerted on each other are not of the same magnitude so that action and reaction principle does not hold in pedestrian dynamics.

Many other problems have been shown, among them:

- Superposition principle problem: The total force acting on a particle is given by the vector sum of individual forces. The principle is valid for all linear systems. However for force-based models it leads to undesired effects, especially in dense situations where unrealistic backwards movement or high velocities can occur.
- Overlapping and oscillations: This problem is related to the Newtonian equation of motion describing particles with inertia that lead to overlapping and oscillations of the modelled pedestrians. Depending on the strength of the repulsive forces the pedestrians might overlap and might violate the principle of volume exclusion.

In Communications for ITS and infrastructure planning reliable quantitative investigation is essential in order to define general patterns in terms of density variation, pedestrians distribution and inter-contact duration between different agents.

2.3.2 Queuing Model

While the social forces model is time-based, queuing models [Pajevic and Karlsson, 2013] are event-based in the sense that they compute the state of the system of each event in a predefined agenda.

These models are used for communication networking perspective, where important abstraction is made, e.g., bidirectional mobility is abstracted to one-dimension. In [Pajevic and Karlsson, 2013] a simple queuing model for pedestrians to estimate achievable content distribution in a confined area is suggested. They focused on specific urban environments where they assume relative zero mobility between nodes, e.g., offices, shops, bus and subway stations, inside buses and trains, stoplights at pedestrian crossings, which they represent as a single mobility block. Their main focus was to neglect internal mobility and investigate the impact of mobility and short transmission range on content distribution.

Queuing models are not appropriate to capture pedestrian dynamics independently to any other system. The concept of servers and the network organization do not correspond to a tangible reality.

2.3.3 Cellular automata

Cellular automata models are discrete in space, time and state variable. Cells of a particular length and width are either occupied or not by a traffic participant. The dynamics are defined by specific rules regarding a pedestrian's motion and probabilities to move to one of the neighbouring cells. An extension of this model to substitute individual intelligence is the floor field by which longer-ranged interactions can be introduced, similar to social forces [Burstedde et al., 2001]. The time discreteness with cellular automata means that all pedestrians move simultaneously at each time step.

2.3.4 Agent-based Models

Game theory

Game Theory model mimics the behaviour of players, adopting a strategy knowing the strategy adopted by other players. The main focus of game theory is to identify and analyse equilibrium situations and the outcome of a game is characterized by a payoff matrix.

[Lachapelle and Wolfram, 2006] came with a model based on the *Mean field Games Theory* (MFG) introduced by [Lasry and Lions, 2006]. They presented a new approach to model crowd motions. A particular feature is that the approximating macroscopic mean field model is derived from a mean field game as the number of players tends to infinity. In this model, the pedestrians are considered as smart individuals, having strategic interactions within the crowd. These individuals anticipate the future. This is mathematically expressed through a forward-backward structure, the forward dynamic describing the crowd dynamic whereas the backward one is needed to build the expectations. The limiting MFG are motivated by an N-player stochastic game. This approach can be used either for macro-simulation and micro-simulation.

The drawback of this model is that the application scope is limited to crowd simulations, which makes the model not suitable to our requirements for SUMO.

Velocity Obstacles

Velocity obstacles are a technique that is applied in robotics that predicts the potential area of space where collision can happen with other robots. In this model, each agent computes the collision-free spaces in order to make the next move.

Velocity obstacles exploit an intuitive principle. As two agents move through a shared space, each agent observes the position and velocity of other agents and predicts whether a conflict will happen in the future [Curtis and Manocha, 2012].

This family model shows many interesting patterns for crowd movements, safety planning and robotics field. Additional research must be conducted in order to show that velocity obstacles can also be applied for urban mobility.

In urban mobility, the scope is larger since the agents share different target destination, which makes the modelling problem at the operational level even more complex. Indeed, the mobility in urban area is completely heterogeneous while it is not the case in the crowd movement where the agents share the same goal and follow the flow of the other agents. If we apply this for safety planning, agents will share the same goal, which is to exit as fast as possible the dangerous area, in this context, velocity obstacles appear to be an appropriate model, since all pedestrian try to avoid collisions but at the same time to approach the exit point.

To summarize, velocity obstacles is based on a clear space definition of collision but on the other hand it does not explain other patterns that are proper to pedestrian behaviours such as "Leader-Follower".

[Antonini et al., 2006] focused more on pedestrian walking behaviour, naturally identified by operational level. By using discrete choice model concepts he went deeper into behavioural aspects of pedestrians' reactions contrary to models utilizing pure mathematical or physical concepts to estimate pedestrian behaviours. He succeeded to develop a model called "Next step model" that proposes a walking pattern where pedestrians choose their next step in a discrete choice framework. Although his model is not yet exploited by simulators, it is scientifically very useful because of its high capability to explore and analyse different pedestrian characteristics and their effects on the walking pattern. The "Next step model" shows different advantages, among them the microscopic calibration on real pedestrian trajectories data, which proves that it can reflect a realistic pedestrians walking behaviour. Although the model is currently used in video tracking, alone it presents a drawback when it is used without a tactical model definition. For example, the model encounters some problems modelling pedestrian which restart walking again after a short break, this is mainly true when it is used over real image sequences.

2.4 Multimodal Interactions at Crossings

2.4.1 Crossings at Traffic Lights

Literature about microscopic intermodal behaviour at signalised crossings can hardly be found. Such interactions arise when right-turning vehicles and crossing pedestrians have a green signal at the same time. As ruled by the traffic law vehicles have to yield to pedestrians. Only when an accepted gap between the end of the starting pedestrian group and later arriving pedestrians appears, vehicles can use their green time.

2.4.2 Mid-Link Crossings

Pedestrians may wish to cross road links anywhere between designated crossings like traffic lights or zebras. From a regulatory viewpoint they do not have the right of way at such undesignated points and must wait until an acceptable gap between following cars or voluntarily stopping cars offer them the possibility to cross. This also counts for median islands and in some countries like Poland even for zebras, where cars do not have to yield to waiting people. Beside the negative impact on traffic safety these crossing manoeuvres cause reductions in speed and capacity for road traffic

The willingness respectively the law obedience to give way to pedestrians was investigated in Poland and Germany [Gaca and Hogendorf, 2007] which is shown in Figure 2.1.

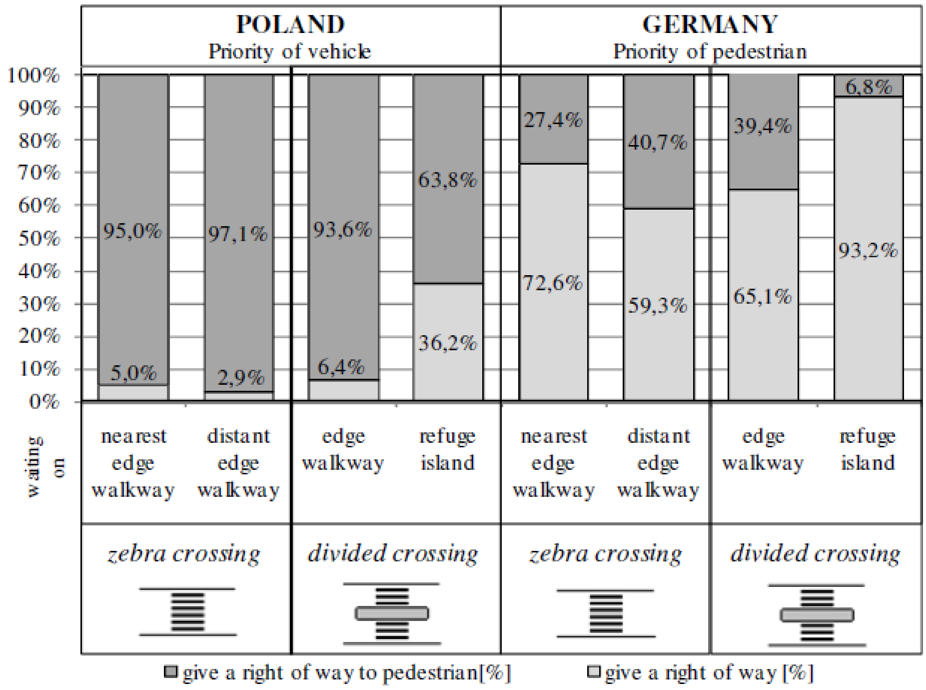

Figure 2.1: Distribution of drivers' behaviour at crossings with or without a median island, dependent on right of way and location of the waiting pedestrian [Gaca and Hogendorf, 2007]

2.5 Demand Modelling and Routing

The demand is the traffic flow between origins and destinations of a network within a particular period, generally denoted as [n/Δt]. Typically chosen periods are 24 hours (daily traffic), one hour (the peak hour), or even 15 minutes. Origins and destinations are typically nodes of the network. As traffic flows occur between multiple origins and destinations they are preferably noted within a so-called O-D-matrix. Such a matrix of pedestrian or bike flows can be derived from a classical macroscopic four-step-model after the trip generation, distribution, and mode choice have been calculated. The final fourth step – the assignment – is then carried out as routing. Such models normally cover areas of many square kilometres and exceed the scope of interest for microscopic simulation by far. Nonetheless smaller pieces can be cut out with traversing flows being preserved. This can be broken down to a single node, where origins and destinations and their respective demand must be defined for its different corners. Alternatively these numbers are often derived from project tailored counting.

2.6 Bicyclists Modelling

A recent paper by [Twaddle et al., 2014] gives an all-encompassing overview about bicyclists' behaviour and its modelling. Interesting behavioural aspects comprise the sudden changes between different types of infrastructure (road, bike lane, sidewalk), considerable percentages of law-breaking by red-light-violations and counter-flow rides, and intrinsic energy efficient driving styles.

Available models are derived from vehicle and pedestrian models and classified as longitudinally continuous models, cellular automata (CA), social force adaptions, and so-called logic models which are agent-based. The paper assesses whether these different models are able to produce viable results on the operational level. This is answered positively when less accurate results outside safety applications or very individual behaviour are sufficient. However, "*to model the interactions of two intersecting traffic streams, conflict points are often used. At a conflict point, road users in the minor stream must stop at a predefined point on their travel path and wait until a suitably large gap between vehicles presents itself. The bicycle specific characteristic of adapting the speed, lateral position or a combination of both is very difficult, or impossible, to depict in current* *simulation tools based on car following models. The use of social force models makes it possible to consider interactions with many different road users in both the longitudinal and lateral direction simultaneously. The modeling of interaction on the operative level in CA models is coarser, but requires much less computing power.*" [Twaddle et al., 2014].

The tactic level comprises the infrastructure selection (bicycle lane, car lane or sidewalk) and the disobedience of traffic laws based on the geometry and signalization of the intersection and presence of obstacles.

2.7 Model Calibration and Validation

The objects of models need to be parameterised within their different control parameter classes, which are mainly of spatial, temporal, behavioural, and technological nature.

The free (desired) speed and its standard deviation are the most common and important parameters. In [Bensator, Härri, Spyropoulos, 2014] 25 different studies were reviewed with regard to their stated values. They are replicated in Table 2.1.

[Marisamynathan and Vedagiri, 2014] found differences in speed depending on the gender and the age (adult vs. over 60).

A special case is the cellular automata model which can contain only one uniform speed for all pedestrians without any deviations.

But the free speed differs when pedestrians need to change their direction, for example when walking around corners or obstacles. The effects of resulting turning angles were computed in [Dias et al., 2014]. This speed decrease is shown for solo walkers and collectives (groups) in Figure 2.2.

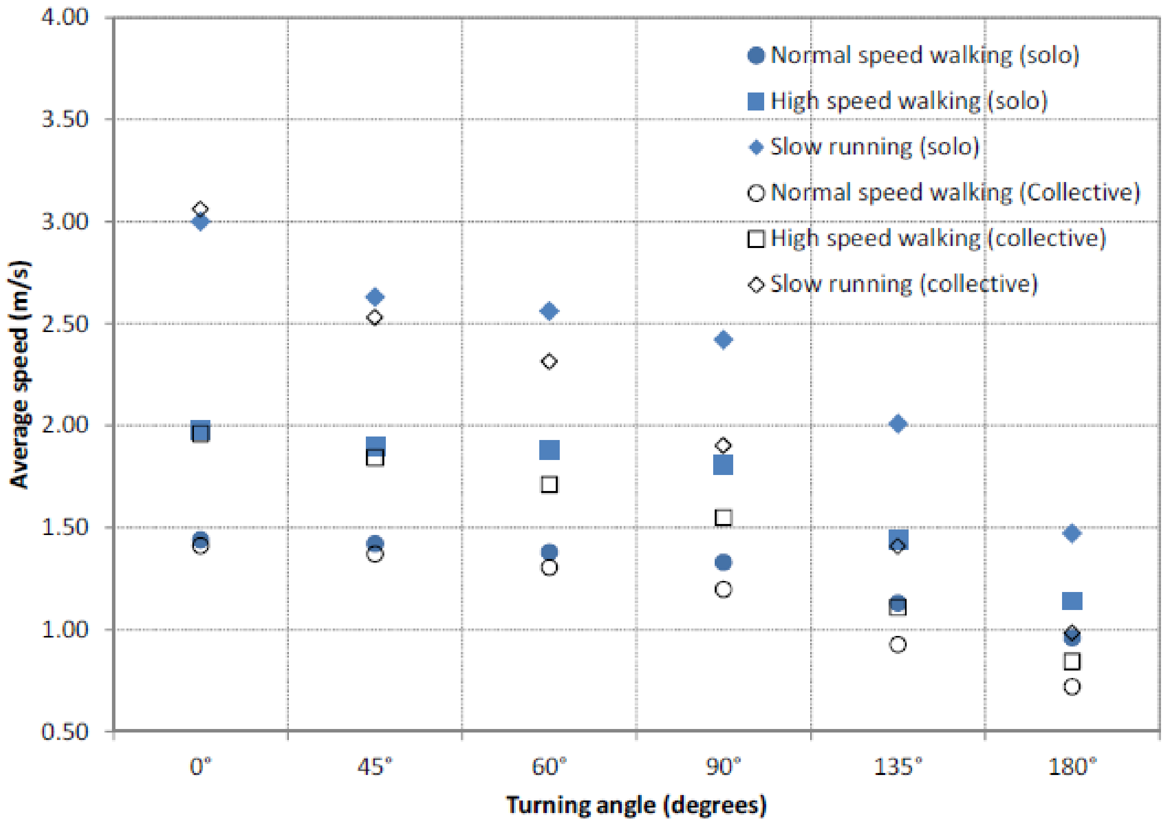

Figure 2.2: Comparison of curved path speeds for all cases with straight path walking speeds [Dias et al., 2014]

A spatial parameter is the circumference of pedestrians. In cellular automata models a cell extent of 40cm x 40cm is common. This measure is also taken as width in other models.

Table 2.1: Observed Values for Mean Speed and its Standard Deviation in Different studies

Source	Mean speed (m/s)	Standard deviation (m/s)	Location
Crow	1.4		Netherlands
Daamen	1.41	0.215	Netherlands
Daly et al.	1.47		United Kingdom
FHWA	1.2		United States
Fruin	1.4	0.15	United States
Hankin and Wright	1.6		United Kingdom
Henderson	1.44	0.23	Australia
Hoel	1.50	0.20	United States
Institue of Transport Engineers	1.2		United States
Knoflacher	1.45		Austria
Koushki	1.08		Saudi-Arabia
Lam et al.	1.19	0.26	Hong Kong
Morrall et al.	1.25 1.4		Sri Lanka Canada
Navin and Wheeler	1.32		United States
O'Flaherty and Parkinson	1.32	1.0	United Kingdom
Older	1.30	0.3	United Kingdom
Pauls	1.25		United States
Roddin	1.6		United States
Sarkar and Janardhan	1.46	0.63	India
Sleight	1.37		United States
Tanariboon et al.	1.23		Singapore
Tanariboon and Guyano	1.22		Thailand
Tregenza	1.31	0.30	United Kingdom
Virkler and Elayadath	1.22		United States
Young	1.38	0.27	United States
Estimated Overall average	**1.34**	**0.37**	

A technological parameter in the context of the COLOMBO project and cooperative ITS in general is the penetration rate of traffic participants with the respective communication units and software applications.

A behavioural parameter is the adherence rate to both obligatory traffic laws, and instructions or advices given by the aforementioned cooperative systems which might be followed only voluntarily.

For cycling models it is stated in [Twaddle et al., 2014], that there is a lack of empirical data that would allow model calibration on a tactical level.

Once the simulation is calibrated a base case scenario need to be run and validated with an independent data set. The quantitative validation flows, speeds, densities, etc. A qualitative validation regards the emerging patterns like dynamic lane formation, formation of diagonal strips in crossing flows.

3 Pedestrian and Bicycle Modelling in COLOMBO

One of the goals of the COLOMBO project is to evaluate traffic light algorithms using the traffic simulation SUMO. While the focus is on motorized road traffic, the interactions between different modes of traffic need to be simulated as well to receive an accurate assessment. To this end, the simulation suite SUMO was extended to allow the simulation of other modes and their interactions with motorized traffic at intersections. In addition, a standalone pedestrian simulation application was developed.

We first describe the requirements for modeling pedestrians and bicycles. In section 3.2 we give an overview of the architecture for modelling intermodal interactions. Rejected alternatives are presented in Appendix G. Most of the extensions that were added to implement the chosen architecture pertain to the modelling of pedestrians and are described in section 3.3. Some extensions were also added to handle the interaction with bicycles and are described in section 3.4. In section 3.5 we provide documentation on deploying the implemented SUMO extensions. Section 3.6 contains an overview of outstanding issues and possible future extensions. Usage documentation can be found in the Appendix B.

3.1 Model requirements

As stated, the ultimate goal of the extensions described in this document is the evaluation of traffic light algorithms under various conditions including intermodal traffic. To enable a qualitative as well as quantitative evaluation, the following functional requirements were deemed necessary:

- vehicles need to wait for pedestrians which are crossing the road in front of them;
- pedestrians need to cross the road in order to continue on the other side;
- right-of-way rules observed normally between the different modes at different types of intersections should be respected;
- pedestrians dynamics should be sufficiently detailed to model the time required for passing a pedestrian crossing including the following aspects:
  - width of the available walking space,
  - bidirectional movement,
  - positioning in front of the crossing while having to wait,
  - density of pedestrian traffic,
  - route choice when passing an intersection diagonally;
- simulation outputs should allow tracking of pedestrians.

Regarding the implementation of pedestrians within SUMO, additional requirements relate to the application chain used in scenario creation:

- The application for network building should be enabled to support the necessary networks structures for meeting the above functional requirements
  - by using explicit input specifications;
  - by using heuristics to generate the necessary structures from context.
- The tools for demand generation should be able to support the creation of multi modal demand
  - in regard to destination choice;
  - in regard to route choice.

Additionally, there were non-functional requirements related to the architecture of the simulation suite SUMO:

- the implemented models should be modular enough to make them replaceable,
- the applications should be backwards compatible with the input data formats of previous versions,

- input data formats should be changed as little as possible,
- the implementation should be fast enough to allow the simulation of city-sized scenarios (at least for some models),
- the visualization of pedestrians should be sufficiently detailed to allow diagnosing the simulation behavior.

In the following we describe the architecture and the models which were implemented to meet these requirements.

3.2 Architecture of Intermodal Interactions in SUMO

Every model is a compromise between accuracy and complexity. Due to our focus on evaluating traffic light algorithms, we have concentrated on modeling the interactions at intersections and simplified the interactions in other areas. In the current implementation road vehicles, bicycles and pedestrians move on separate network elements. Each mode only interacts with members of its own mode while traveling along a road and the interaction between modes happens at intersections only.

SUMO simulates movements along unidirectional roads (also called edges) consisting of one or more lanes which each lane allowing as many vehicles as it's longitudinal length permits but only ever allowing a single vehicle in the lateral direction. At intersections (also called nodes) vehicles regard traffic lights and right of way rules before passing. To allow for pedestrians and bicycles, additional lanes are added to existing edges which represent sidewalks and bicycle lanes. Furthermore, "blind" lanes which do not allow any traffic can be added to model green verges between these mode-specific lanes. To model the paths of pedestrians at intersections, specialized edges are added for modeling pedestrian *crossings* and for modeling sidewalk corners where crossings and sidewalks meet (later called *walkingarea*). Figure 3.1 shows a controlled 4-arm intersection with these features. As demonstrated in chapter 4 this architecture allows to model relevant intermodal dynamics.

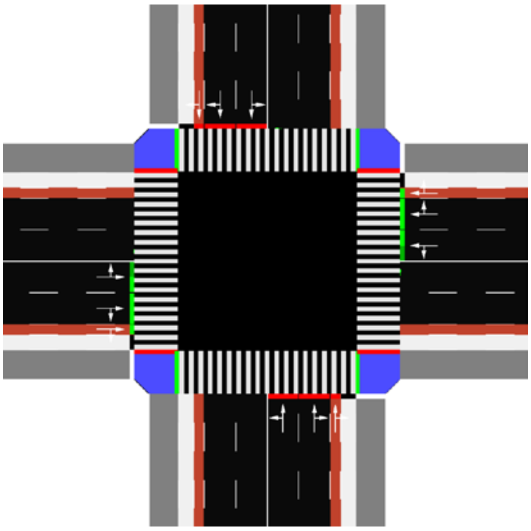

Figure 3.1: Visualization of an intermodal intersection colored by mode: motor vehicles (black), bicycles (brown), pedestrians (gray, blue, striped), forbidden (light gray).

Currently, bicycles dynamics are modelled by using the car-following model with modified parameters (i.e. speed and acceleration). Pedestrians on the other hand are handled by a customizable model which has full control of the use of the available space. This is of particular importance to allow bidirectional movement which is impossible for road vehicles simulated in SUMO. The parameterization of these models is described in chapter 4. It should be noted that model calibration is outside the scope of this document.

Some features of real word traffic cannot currently by modelled by our architecture. A list of possible extensions is given in section 3.6.

3.3 Extensions for modeling Pedestrians

SUMO has allowed for intermodal simulation since around 2007 but the only interaction was the propagation of delays along intermodal trip-chains. In the scope of these trip-chains a very basic walking model was added were pedestrians moved along the edges with constants speeds, “jumped” across intersections and never interacted among themselves or with other modes.

For the COLOMBO project, the simulation was extended in several areas to model interactions among pedestrians and between pedestrians and other modes. These are described in the following by contrasting them to “previous” version of 0.20.0.

3.3.1 Extension to the SUMO Road Network Model

In the previous version of SUMO, the network model differentiates between two types of edges. “Normal” edges are those which connect intersections while “internal” edges are spaces within an intersection which are used when passing that intersection. These types of lanes are show in Figure 3.2. Each edge consists of one or more lanes and the connectivity in the network is described at the level of these lanes. The existing network model allows restricting the use of each lane to a set of access classes (called `vClass`). Among the predefined classes are “passenger”, “bus”, “delivery” and “pedestrian”.

To these existing edge types two new types were added which are exclusive to modeling pedestrians (and only allow `vClass` “pedestrian”). Edges of type “crossing” lie within an intersection and model the path of pedestrians crossing a street. Edges of type “walkingarea” also lie within an intersection and model the space in front of a crossing where pedestrians wait. Furthermore they model the conflict space where pedestrian streams in different directions interact. Either of these new types always consists of a single lane. The addition of these new types was found to be a good compromise between modeling the topology as well as the geometry of pedestrian paths while minimizing changes to the network format. For rejected alternative architectures see Appendix G.

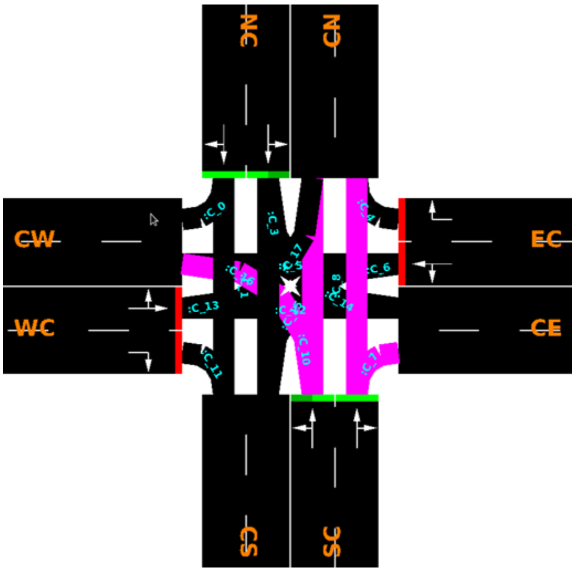

Figure 3.2: Previous network model with “normal” edges labelled in orange and “internal” edges labelled in cyan. The internal edges outgoing from edge “SC” are highlighted in magenta.

The network for pedestrian movements then consists of the following elements:

- sidewalks: typically, the rightmost lane of a "normal" edge. The `vClass` "pedestrian" must be allowed,
- walkingareas,
- crossings.

Possible connections between these elements are given in Table 3.1.

Table 3.1: Possible connections between different pedestrian infrastructure elements

	Sidewalk	Walkingarea	Crossing
Sidewalk	x	x	
Walkingarea	x		x
Crossing		x	

As defined in the network file (called `net.xml`), the lanes and the connections have a direction. This simply follows the logic of the previous network format. However, pedestrians can traverse the lanes and the connections in either direction.

Edges of the type "walkingarea" have the unique property of being connected to edges in multiple directions so as to make the question of their direction ambiguous. Resolving this ambiguity is left to the pedestrian model. When drawing a *walkingarea* in the GUI its "shape"-attribute is interpreted as the polygonal boarder around the space, rather than as a polygonal line in the direction of the edge.

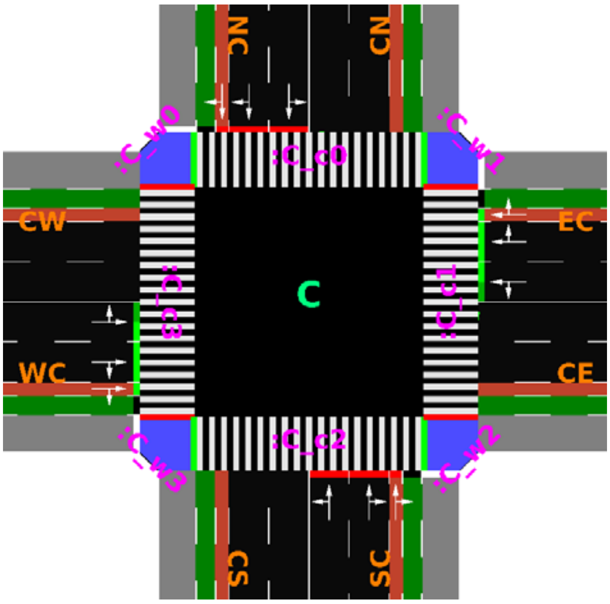

Figure 3.3: 4-arm intersection with bicycle lanes (brown), sidewalks (grey), walkingareas (blue), crossings (striped), green verges (green). IDs are shown for all edges which may be used by pedestrians.

Edges of type "crossing" may have priority which means pedestrians can assume that vehicles will yield. This is the case at controlled intersections and specially marked pedestrian crossings. Otherwise, pedestrians may only cross the street if there is a sufficient time gap to all vehicles in their path. The behavior in the latter case is discussed in section 3.3.3.

The distinct visualization of the new edge types can be seen in Figure 3.3. It shows a 4-arm junction with 4 *walkingareas* and 4 pedestrian *crossings*. The edges `:C_wi` and `:C_ci` with *i* in (0…3) along with the rightmost lanes of the edges NC,CN,EC,CE,SC,CS,WC,CW make up the network for pedestrian movement. The connections originating from normal road lanes and bicycle lanes are indicated by white arrows. The connections for pedestrians are not visualized. Each *walkingarea* is connected to all adjacent sidewalks and crossings. The connectivity at *walkingarea* `:C_w0` in the upper left corner of intersection `C` is given in Table 3.2.

Table 3.2: Connectivity at *walkingarea* :C_w0

fromEdge	toEdge	fromLane	toLane
NC	:C_w0	0	0
:C_w0	CW	0	0
:C_w0	:C_c3	0	0

The generation of these new network structures is discussed in the following section.

3.3.2 Extensions to NETCONVERT

The NETCONVERT application is part of the SUMO application suite. It is responsible for preparing the simulation network[3] (`net.xml`) from a wide range of input data formats such as OpenStreetMap (OSM), VISSIM or shape files[4]. Another important input format is a set of simple xml inputs (called plain XML) which describe the nodes, edges and optionally the connections of the road network[5]. NETCONVERT enriches its inputs by computing connectivity, right-of-way rules, and the geometry of intersections with configurable heuristic models.

To support intermodal simulations, NETCONVERT was extended to create sidewalks as well as the new edge types "walkingarea" and "crossing" described in the previous section. The crossings must be included in the generated right-of-way rules. Furthermore, heuristically generated traffic-light programs are adapted to include pedestrian signals. We describe these procedures in the sequence in which they are executed.

Generating Sidewalks

Sidewalks may be defined explicitly in plain XML input when describing edges (plain.edg.xml). This is done by defining an additional lane which only permits the `vClass` "pedestrian" and setting the appropriate width. When importing edges with defined types, it is also possible to declare that certain types should receive a sidewalk. This can be used to automatically generate sidewalks for residential streets while omitting them for motorways when importing OSM data. An example input file is shown in Appendix E. A third option which can be used if no edge types are available is a heuristic based on edge speed. This is activated by using the options given in Table 3.3.

Table 3.3: Options defining the sidewalks heuristics

Option	Type	Description
--sidewalks.guess	Bool	Activate the heuristic for generating sidewalks
--sidewalks.guess.min-speed	Float	Generate no sidewalks for edges with a lower speed (m/s)
--sidewalks.guess.max-speed	Float	Generate no sidewalks for edges with a higher speed (m/s)

[3] http://sumo-sim.org/userdoc/Networks/SUMO_Road_Networks.html
[4] http://sumo-sim.org/userdoc/Networks/Import.html
[5] http://sumo-sim.org/userdoc/Networks/Building_Networks_from_own_XML-descriptions.html

When adding sidewalks via type maps or heuristics, additional geometry computations are performed to ensure that the geometry of road lanes remains unchanged.

Generating Crossings

Crossings may be defined explicitly in plain XML input when describing connections (plain.con.xml). This is done using the new XML element `<crossings>` with the mandatory attributes `node="<node id>" edges="<list of edges to cross>"` and the optional attributes `width="<width in m>"` and `priority="true/false"`. This defines a crossing at the given node across the given list of edges. Crossings at TLS-controlled nodes are always prioritized. Since crossings are always associated with nodes, a node must be present if a crossing somewhere along an edge is to be modelled. This fits well into the existing simulation architecture which only recognizes conflicting traffic streams at nodes.

The second available method adding crossing information to a network is with the option `--crossings.guess`. This enables a heuristic which adds crossings wherever sidewalks with similar angle are separated by lanes which forbid pedestrians. If the edges to be crossed have sufficient distance between them or vary a by a sufficient angle, two crossings with an intermediate walking area are generated. Such split crossings can be seen in Figure 3.4.

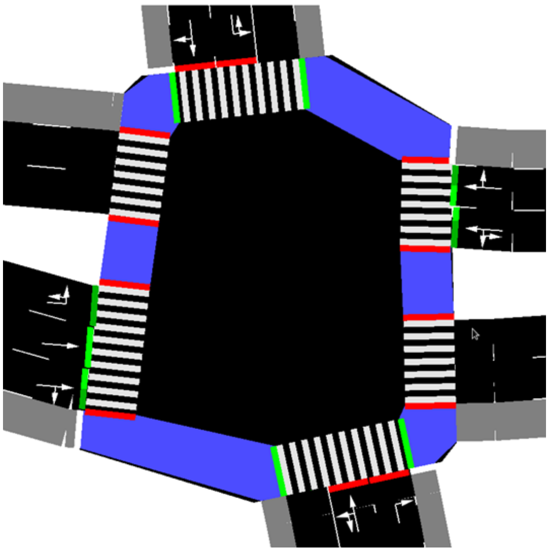

Figure 3.4: Intersections with split pedestrian crossings across horizontal roads.

Generating Right of Way Rules

In SUMO there are two concepts for modelling the influence of a conflicting traffic stream on a vehicle:

- Each vehicle registers its approach to an intersection along with an expected time slot for passing the intersection. A vehicle approaching the intersection must yield to vehicles with higher priorities which want to use the same time slot.
- Each vehicle must cross certain set of "foe" lanes which are used by conflicting streams. The vehicle must yield regardless of priority whenever such a "foe"-lane is occupied by another vehicle (and the vehicles are not geometrically past the conflict point).

These concepts are embodied in a right-of-way matrix and in a conflict matrix. The first matrix (called "response") defines for each connection, the set of foe connections to which it must yield in case of registered approaches. The second matrix (called "foes") describes for every connection, the set of foe lanes which may not be crossed in case of occupancy. These matrices are extended to incorporate crossings depending on whether they are prioritized or not. In the former case, all connections which have trajectories intersecting the crossing must yield to pedestrians occupying the crossing whereas the crossings themselves are only flagged in “foes” matrix which means pedestrians are free to walk. In the case of unprioritized crossings, the right of way varies depending on the properties of the road connection: Vehicles which perform a left or right turn must yield if there is a pedestrian crossing on their target edge. Otherwise the pedestrians must yield to all vehicles.

Generating Signal Plans for Crossings

A controlled intersection with pedestrian crossings needs to incorporate information about the signal states for pedestrians. In the previous versions of SUMO all connections entering an intersection are generally controlled by the traffic light. When adding pedestrian structures, this no longer holds true. Connections between sidewalks and *walkingareas* are never controlled. On the other hand connections from *walkingareas* to crossings are always controlled. Connections from crossing to *walkingarea* are uncontrolled as it is always possible to leave the crossing. When entering a crossing in the backward direction (relative to its natural direction), the traffic light state for the forward entering connection is substituted instead of using the (uncontrolled) connection from the crossing to the walking area in reverse.

The additional controlled connections are indexed in clockwise directions starting in the north following the connections from normal edges. Thus, signal plans for such intersections can be given explicitly by defining phase states of the appropriate size.

When signal plans are generated heuristically, the signal state for pedestrian crossings is set to “red” whenever any intersecting straight connections are set to “green major” (being able to drive with absolute priority). Otherwise the crossing is set to the “green major” state itself. This ensures that pedestrians are only allowed to walk when they do not disturb straight-going traffic. TLS signals for vehicles with a green state are set to “green minor” if the destination edge of that connection intersects a pedestrian crossing which also has a green state. This models the fact that vehicles turning right or left at an intersection need to yield to pedestrian crossings when leaving an intersection. The “green minor” state ensures a slow approach which allows vehicles to brake for pedestrians in time.

Generating Walkingareas

Whenever at least two sidewalks are adjacent at an intersection or a sidewalk is adjacent to a crossing, a *walkingarea* which connects these structures is generated. The geometry is computed as the clockwise path around the connected pedestrian lanes. Unidirectional connections following the existing schema for regular road connection are generated according to the following rules:

- sidewalks of edges incoming to the current node have a connection to the walkingarea
- walkingareas have a connection to sidewalks of outgoing edges
- Connections between walkingareas and crossings are generated in a counter-clockwise fashion around the node.

Currently *walkingareas* are only generated if the network is built with the option `--crossings.guess` or at least one crossing is specified in the input files. This was done to still allow the generation of networks without pedestrian structures but could be made configurable in the future.

3.3.3 Pedestrian Dynamics in the Simulation

In this section we describe the extensions to core simulation application SUMO for modelling pedestrians. We first describe the interface between the pedestrian models and the rest of the simulation. Next, we give a brief description of the dynamics of the previous version and then describe the extensions in detail. Some of these extensions are generic which means they concern the architecture and the interfaces of the simulation while others are model specific. In this case we describe the dynamics of a specific model which was implemented to showcase the new features.

Interface of Pedestrian Models

The pedestrian model to use can be selected by using the new simulation option `--pedestrian.model`. The interface between this model and the rest of the simulation was designed with the aim of having a high degree of freedom when implementing new models. Thus it only concerns the most basic areas of interaction. Any pedestrian model must support the following functionality:

- Return whether a given lane is currently blocked by any pedestrians from being passed at a certain location (function `blockedAtDist`)
- Add a new pedestrian and return a PedestrianState object which must be able to report on the position and angle and speed of that pedestrian

When adding a pedestrian to be controlled by the Pedestrian model, the following information must be supplied:

- A sequence of (normal) edges to define the “skeleton” of the walking route
- The starting position relative to the first edge
- The destination position relative to the last edge
- The maximum speed

These attributes are all contained in the definition of a `<walk>` which is part of a person’s plan, just as in the previous version of SUMO. It is the responsibility of the pedestrian model to select the sequence of walkingareas and crossings which are needed to connect the given normal edges when passing an intersection.

Previous Dynamics (nonInteracting)

In the previous versions of SUMO pedestrians followed a very basic walking model. They always moved forward along normal edges (regardless of any access restrictions) and “jumped” across intersections. They could either be configured to complete a walking trip in a fixed amount of time or to move along the edges with a fixed speed. No interaction between pedestrians and vehicles or other pedestrians took place.

This very simple model is made available in the new version of SUMO using the option `--pedestrian.model nonInteracting`. It may be useful if the pedestrian dynamics are not important and high execution speed is desired. One enhancement that was made is that now pedestrians may use edges in either direction which is computed based on the topology of the edge sequence.

New Dynamics (striping)

To showcase the new features which are enabled by the architecture, a new pedestrian model was developed. This model is selected using the option `--pedestrian.model striping` (but it is also the new default model for pedestrians). This new model has three main functionalities:

1. interactions of pedestrians with each other
2. interactions between pedestrians and other modes
3. routing within an intersection

The model assigns 2D-coordinates within a lane (of type `sidewalk`, `walkingarea` or `crossing`) to each pedestrian. These coordinates which are defined relative to the leftmost side of the start of the lane are updated in every simulation step. This is in contrast to the coordinates of vehicles, which (generally) only have 1D-coordinates within their respective lane. Pedestrians advance along a lane towards the next node which may either correspond to the natural direction of the lane (forward movement) or it may opposite to the natural direction (backward movement). Thus, the x coordinate monotonically increase or decreases while on a lane. Once the end of a lane has been reached, the pedestrian is placed on the next lane (which may either be unique or determined dynamically with a routing algorithm). Details of the coordinate update rules are described in the following sections.

Interactions between Pedestrians

The most important feature of pedestrian interactions is collision avoidance. To achieve this, the “striping”-model divides the lateral width of a lane into discrete stripes of fixed width. This width is user configurable using the option `--pedestrian.striping.stripe-width` and defaults to 0.65 m. These stripes are similar to lanes of a multi-lane road are used by vehicles. Collision avoidance is thus reduced to maintaining sufficient distance within the same lane. Whenever a pedestrian comes too close to another pedestrian within the same stripe it moves in the y-direction (laterally) as well as in the x-direction to change to a different stripe. The y-coordinate changes continuously which leads to situations in which a pedestrian temporarily occupies two stripes and thus needs to ensure sufficient distances in both. The algorithm for selecting the preferred stripe is based on the direction of movement (preferring evasion to the right for oncoming pedestrians) and the expected distance the pedestrian will be able to walk in that stripe without a collision.

During every simulation step, each pedestrian advances as fast as possible while still avoiding collisions. The updates happen in a single pass for each walking direction with the pedestrian in the front being updated first and then its followers sorted by their x-coordinate. The speed in the x-direction may be reduced by a random amount with the maximum amount defined as a fraction of the maximum speed, using the global option `--pedestrian.striping.dawdling` (defaulting to 0.2).

As a consequence of the above movement rules, pedestrians tend to walk side by side on sidewalks of sufficient width. They wait in front of crossings in a wide queue and they form a jam if the inflow into a lane is larger than its outflow.

Interactions between Pedestrians and other Modes

In SUMO there are two concepts for modelling the influence of a conflicting traffic stream on a vehicle:

a) Each vehicle registers its approach to an intersection along with an expected time slot for passing the intersection. A vehicle approaching the intersection must yield to any vehicle with higher priority which wants to use the same time slot.
b) Each vehicle must cross certain set of “foe” lanes which are used by conflicting streams. The vehicle must yield regardless of priority whenever such a “foe”-lane is occupied by another vehicle (and they vehicles are not geometrically past the conflict point).

Concept a) is used for modelling uncontrolled crossings. A pedestrian wishing to cross the street at an uncontrolled intersection can only do so if its expected time slot for using the intersection does not interfere with that of an approaching vehicle. It should be noted that the dynamics at unprioritized crossings are conservative in estimating the time required gap. In the simulation, pedestrians will only use such a crossing if the whole length of the crossing is free of vehicles for the whole time needed to cross. In reality, it can be observed that pedestrians start to cross while vehicles are still occupying the far side of the crossing. Furthermore, they will often choose to cross even if the near side of the crossing will be occupied by vehicles by the time the pedestrians are past that point. This kind of “weaving” between traffic is especially notable in dense traffic if pedestrians would otherwise have to wait very long.

Concept b) is used for preventing vehicles from driving across a pedestrian crossing which is occupied by pedestrians. Pedestrians themselves never register for a time slot. While they have not moved onto the crossings, vehicles are free to drive. The influence on vehicles is implemented via the interface method `blockedAtDist` which is called to request whether a “foe”-lane in the vehicles path is blocked at specified distance due to the presence of pedestrians. The given distance value corresponds to the geometric intersection between the crossing and the vehicles trajectory measured as distance from the start of the crossing. The “striping”-model computes its results by iterating over all pedestrians on the lane and returns “blocked” status if a pedestrian is found which is not yet past the intersection point but within a threshold distance to that point (currently hardcoded as 10m). For “foe”-lanes other than crossings the check always returns false since pedestrians do not walk there.

Concept b) could also be used to prevent pedestrians from walking into vehicles which occupy the crossing but this is currently not implemented.

The “striping”-Model can be seen as a compromise between space-discrete and space-continuous pedestrian models due to combination of continuous positions and discrete stripes. The model captures qualitative dynamics when there are two main directions of movement such as is found on sidewalks and crossings but is not well suited to describe the dynamics in other cases. As an advantage over other more detailed models it allows for a computation time which is linear in the number of simulated pedestrians. More specifically the running time for executing a single simulation step is in the order of $O(n \times k)$ with n being the number of pedestrians and k being the maximum number of parallel stripes for all lanes. This is achieved by using only a very limited set of surrounding pedestrians to compute pedestrian interactions.

3.3.4 Pedestrian Routing

The route of a pedestrian may either be specified as a sequence of (normal) edge ids in the `edges` attribute of a person’s `<walk>` element or it may be given as a pair of start and destination edges using the `from` and `to` attributes. In the former case local routing must be performed to find a sequence of *walkingareas* and crossings to connect two subsequent normal edges. In the latter case global routing is required to find a sequence of pedestrian lanes (*normal*, *walkingarea* and *crossing*) to connect the start- and destination edge.

The routing task that must be solved differs from vehicular routing insofar as the edges and connections may be used in both directions. To solve this task a routing algorithm was implemented which transforms the bidirectional graph into a unidirectional graph which can then be given to the existing vehicular routing algorithms. The returned route is then transformed back into a sequence of edge ids on the original graph.

Another area where pedestrian routing differs from vehicular routing is the dynamic treatment of TLS-controlled crossings. In reality pedestrians which need to cross the streets twice to reach the diagonal corner of an intersection will usually select the crossing which first shows a green light, using the knowledge that the second crossing will be green soon after they reach it. To achieve this type of behavior, the travel times which are returned by each edge in the routing graph take into account whether access to an edge is regulated by a traffic light which is currently in its red phase.

The travel time for passing an edge behind a red light is computed using the following formula:

$$traveltime = length / speed + max(0, 20 - (t - t_D))$$

where $(t - t_D)$ is time offset to reach that edge from the current moment. Thus, red lights in close proximity are avoided while far away red lights are not.

The DUAROUTER application was extended to perform pedestrian routing whenever a `<walk>` element contains attributes `from` and `to`. The route output then contains a `<walk>` element with all normal edges of the computed route. The SUMO application was likewise extended to handle `<walk>` element with attributes `from` and `to`.

The pedestrian model "striping" employs the pedestrian routing algorithm whenever a pedestrian reaches a *walkingarea* and needs to decide which crossing to use. In contrast, the pedestrian model "nonInteracting" does not perform dynamic routing. Pedestrians "jump" across intersections and strictly follow the sequence of normal edges.

3.3.5 Simulation Outputs

In the previous version of SUMO only a single type of outputs for persons was available: The time of completion for each completed element of a person's plan. For `<ride>` elements, the time of departure was also given. This output is triggered by supplying the option `--tripinfo-output`.

An additional facility for obtaining information on persons, the "netstate" output (option `--netstate-dump`[6]) was extended to include information about the position of each person during every simulation step. The output takes the following form:

```
<timestep time="0.00">
    <edge id="CN">
        <person id="p0" pos="20.00" angle="0.00" stage="walking"/>
        <person id="p1" pos="20.00" angle="180.00" stage="walking"/>
    </edge>
</timestep>
```

As can be see, multiple pedestrians may occupy the same length-wise position if the sidewalk on the corresponding edge is wide enough.

The fcd-output (option `--fcd-output`[7]) was likewise extended to include information on pedestrians:

```
<timestep time="0.00">
    <person id="p0" x="23.00" y="2.30" angle="90.00" speed="0.82" pos"20.00"
       edge="CN" slope="0.00"/>
</timestep>
```

3.3.6 Pedestrian Demand generation (TOOLS)

Pedestrian demand may be specified in XML input files as in the previous version of SUMO by declaring persons with `<walk>`-elements:

```
<routes>
    <person id="p0" depart="0">
        <walk edges="NC CN"/>
    </person>
</routes>
```

The existing script `randomTrips.py` was extended to allow the generation of random pedestrian demand by adding the following options:

[6] http://sumo-sim.org/wiki/Simulation/Output/RawDump
[7] http://sumo-sim.org/wiki/Simulation/Output/FCDOutput

- Option `--pedestrians`. This serves to generate persons with a single `<walk>` element instead of vehicles.
- Option `--max-distance` <FLOAT>. This serves to limit the (air)-distance of generated trips and is useful to avoid walks of excessive lengths.

Furthermore, the script was extended to recognize access restrictions when generating source and destination edges. This serves to avoid walks starting on motorways or other edges without sidewalk.

3.3.7 Extension to SUMO-GUI

"SUMO-GUI", the application for visualizing a simulation was extended in the following areas:

- Lanes of type crossing are shown with a zebra pattern which is white for prioritized crossings (as often found in reality) and dark grey for unprioritized crossings (which is unlike reality but allows easy recognition of these lanes).
- *Walkingareas* are drawn as a filled polygon to visualized parts of the junction which are reserved for pedestrians (and correspond to a special part of the sidewalk in reality).
- The colored bars which signify the state of TLS-controlled connections are drawn at both sides of a crossing since these are used by pedestrians in both directions.
- Pedestrians are now drawn according to their configured width and length. Their shape for visualization was modified to always show the walking direction and they may be colored to visualize different properties.
- Pedestrians are drawn at 2-dimensional coordinate on their network element according to the used pedestrian model (whereas cars and bicycles only have 1-dimensional coordinates within their lanes)
- Pedestrian attributes can now be accessed from a context menu.

At the time being, a kind of a "standard" or at least "default" coloring scheme that allows to easily distinguish between different modes of transport is assumed to be additionally needed. This will be targeted in the future.

3.4 Extension for Modeling Bicycles

Only minor extensions were implemented to model bicycle traffic. Bicycles are modeled as special vehicles by using appropriate parameters for the existing car-following models (i.e. lower maximum speed and acceleration). Bicycles may either drive on the road or on designated bicycle lanes. The latter may be added by specifying them in plain edg.xml input as described for sidewalks in section 3.3.2.

The NETCONVERT application was extended to recognize the following inter-modal conflicts:

- between right-turning motor vehicles and straight going bicycles (connections must still be defined explicitly)
- between right-turning bicycles and straight going pedestrians

Experimental results for a traffic simulation involving bicycles along with the used model parameters are description in section 4.2. Section 3.6 contains a list of desirable future extensions for supporting bicycle simulation.

3.5 Deployment

Availability

The extensions described in this document will be part of the upcoming SUMO distribution 0.21.0.

System requirements

The extensions described in this document incur no additional library dependencies over those required by the regular SUMO suite. Therefore, they can be deployed on all the originally supported platforms (MS Windows, Linux, Apple's iOS).

User Documentation

The documentation on using the described extensions was added to the SUMO documentation. Adaptations include:

- inclusion of option description into the manual pages of the extended applications[8],
- a description how to include pedestrians in a simulation, listed in this document as Appendix H
- extension of the "native" XML format description[9]
- references to the aforementioned documentation in the persons' specification[10]

The kind reader may note that this document includes other descriptions about setting up and using the implemented pedestrian models in the appendices "Appendix B", "Appendix C", "Appendix D", and "Appendix E". They will be included in SUMO's user documentation at a later time.

Tests

To validate that the implemented extensions satisfy the requirements stated in section 4.1, an number of automated regression tests were set up which are run daily to ensure that existing functionality is not broken in successive modifications. Currently 51 simulation tests (using application SUMO) and 32 network building tests (using application NETCONVERT) as well as 5 tests for application DUAROUTER exist to test the extended functionality.

Among the tested functionalities are:

- vehicles being blocked by crossing pedestrians,
- routing across an intersection,
- pedestrian interactions,
- building pedestrian networks with fully specified XML inputs,
- building pedestrian networks based on heuristics,
- pedestrian routing.

The full list of tests is given in Appendix F.

[8] The extensions will be not presented in this deliverable, as these documentation pages are relatively long and COLOMBO's contribution, which was described in prior chapters, is relatively small in comparison. The changes include the manual for the applications SUMO and NETCOVERT and their current versions as well as the history can be investigated at http://sumo-sim.org/wiki/SUMO and http://sumo-sim.org/wiki/NETCONVERT

[9] http://sumo-sim.org/wiki/Networks/Building_Networks_from_own_XML-descriptions; not replicated here for same reasons

[10] http://sumo-sim.org/wiki/Specification/Persons; not replicated here for same reasons

3.6 Possible future Extensions

Besides the work performed as given in sections 3.2 to 3.4, several further simulation extensions would be desirable to increase the usefulness of the intermodal simulation as well as its accuracy:

- Implementation of additional pedestrian models (i.e. social force) which better model the interactions of pedestrians on walking areas where streams in different directions interact and to improve the dynamics on sidewalks and crossings as well.
- Allow street crossings by pedestrians anywhere on an edge (without disturbing traffic).
- Heterogeneous lane use:
  - allow road vehicles to pass a bicycle (or other two-wheeler) on a single wide lane,
  - allow multiple narrow vehicles to drive side by side on a single lane.
- Using the opposite-direction lane for passing and overtaking manoeuvres (especially when overtaking bicycles on narrow roads)
- Custom bicycle model to allow for bidirectional lane use.

Likewise, NETCONVERT the application for generating simulation networks could be extended in several ways to better support intermodal simulations:

- automated generation of bicycle lanes and their connectivity with NETCONVERT,
- import of bicycle lanes from OSM,
- import of sidewalk information (explicit pedestrian permissions and prohibitions) from OSM. This information is not yet widely available as shown in Figure 3.5.
- Generation of spacing (green verges) between road and bicycle lane, road and sidewalk, bicycle lane and sidewalk
- Input options for defining the shapes of intersections and *walkingareas*

Support for generating multi-modal demand could be improved by extending the ACTIVITYGEN application to generate pedestrian and bicycle demand.

Strengthening tool support for public transport will also be an important area of improvement. Not only do busses and trams interact with road traffic, the usage of public transport infrastructure also generates pedestrian traffic to, between and from stations.

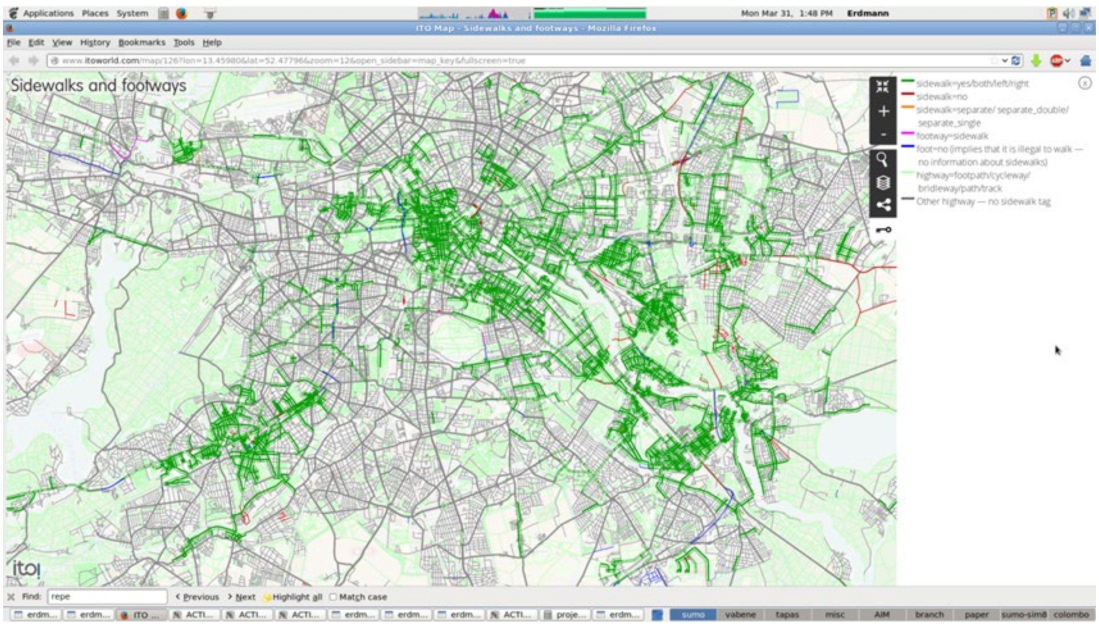

Figure 3.5: Heterogeneous availability of sidewalk information in OSM. No information is provided for streets marked in grey.

3.7 Standalone Behavioral Model Implementation

Besides extending the used traffic simulation SUMO, a further model has been implemented, realized as a standalone simulation application. This application is described in the following. The scope of use within COLOMBO is discussed at the end of this section.

3.7.1 Introduction

Similarly to vehicular mobility, pedestrian also have a strong correlation in their movements. They tend to follow a direct lane at a desired speed, but might have to alter their direction or speed according to potential other pedestrians or cars. The motion correlation between pedestrian is therefore critical to model in order to be able to reproduce close to reality motion patterns (cluster, speed, etc.). After reviewing multiple existing models, we chose to focus on behavioral models and the related discretization of mobility options, both for the modularity as well as implementation scalability.

We chose to implement the behavior model from Antonini et al [Antonini, Berlaire, Schneider, Robin, 2009], which addresses the walking process as a sequence of choices over time. Pedestrian are assumed to be rational decision makers and base the process of choosing their next position in the surrounding space as a function of their kinematic characteristics and reacting to the presence of other individuals. Although strictly not a flow model, this model should also be implementable into SUMO.

3.7.2 The Behavioral Model

The walking process is considered as a sequence of choices over time, where individuals choose where they will be at the next time step, in the space around their current position. This decision making process is approached through the rational behaviour paradigm, using **Discrete Choice Models (DCM)**, with a **random Utility (RU)** representation. Following this paradigm, individuals (decision makers) choose between different options, maximizing a certain utility function. Such a function is represented by the utility that each individual perceives from each of the available alternatives. In this context, the term rational is used to describe decision makers having consistent and transitive preferences. It means that repeated choices are made under identical circumstances, and the preferences over different alternatives satisfy the transitivity property.

The choice for DCM and RU models is dictated by two reasons. First, these kinds of models are based on utility maximization, so they are consistent with the rational behaviour assumption. Second, DCMs are disaggregate behavioural models, designed to forecast the behaviour of individuals in choice situations, where the set of alternatives is finite. As a consequence, they well adapt to a microscopic approach to pedestrian modelling, and they are coherent with an agent-based methodology.

Modeling travel behavior is a key aspect of demand analysis, where aggregates demand is the accumulation of individual's decisions. We focus on "short-term" travel decisions. The analysis of travel behavior is typically disaggregated, meaning that the models represent the choice of individual travelers.

We define the DCM general framework is described as follows:

1. **Decision maker**: defining the decision-making entity and its characteristics;
2. **Alternatives**: determining the options available to the decision-maker;
3. **Attributes**: measuring the benefits and costs of an alternative to the decision-maker according to a *utility function*.
4. **Decision rule**: Describing the process used by the decision-maker to choose the best alternative.

The utility function is modelled, as a Random variable in order to reflect the uncertainty related to unknowns parameters, such as unobserved alternative attributes, unobserved individual characteristics, or measurement errors.

More specifically, the utility that individual *n* associates with alternative *i* in the choice set C_n is given by:

$$\boldsymbol{U_{in} = V_{in} + \varepsilon_{in}}$$

Where V_{in} is the deterministic part of the utility, and ε_{in} is the random term, capturing the uncertainty. The alternative with the highest utility is chosen. Therefore, the probability that alternative *i* is chosen by decision-maker n from choice set C_n is

$$\mathbf{P}(\mathbf{i}|\boldsymbol{C_n}) = \mathbf{P}[\boldsymbol{U_{in}} \geq \boldsymbol{U_{jn}} \forall \boldsymbol{j} \in \boldsymbol{C_n}] = \mathbf{P}[\boldsymbol{U_{in}} = \mathbf{max}(\boldsymbol{U_{jn}})]$$

We briefly described next the deterministic part, as well as the random term.

The **deterministic term** V_{in} of each alternative is a function of the attributes of the alternative itself and the characteristics of the decision-maker.

Considering z_{in} as a vector of attributes as perceived by individual n for alternative i, and S_n as the vector of characteristics of individual n, we can formulate an aggregated vector of attributes from both z_{in} and S_n, as:

$$\boldsymbol{x_{in} = h(z_{in}, S_n)}$$

The choice of h is very general, and several forms may be tested to identify the best representation in a specific application. It is usually assumed to be continuous and monotonic in z_{in}. For a linear in the parameters utility specification, h must be fully determined function (meaning that it does not contain unknown parameters). Then we have

$$\boldsymbol{V_{in} = V(x_{in}) = \sum_k \beta_k x_{ink}}$$

The deterministic term of the utility is therefore fully specified by the vector of parameters β $(Taste\ of\ variation)$.

The random term of the utility is based on *Nested Logit Model* [Ben-Akiva, Bierlaire, 1999] is designed to capture correlations among alternatives. It is based on the partitioning of the choice set C_n (potential mobility alternatives) into M nests C_{mn} such that

$$\boldsymbol{C_n = \bigcup_{m=1}^{M} C_{mn}}$$

and $\boldsymbol{C_{mn} \cap C_{m'n} = \emptyset \ \forall\, m \neq m'}$

The utility function of each alternative is composed of a term specific to the alternative $(\boldsymbol{V'_{in} + \varepsilon'_{in}})$ and a term associated with the nest. If *i* is an alternative from nest $\boldsymbol{C_{mn}}$ $(\boldsymbol{V'_{C_{mn}} + \varepsilon'_{C_{mn}}})$, we have

$$\boldsymbol{U_{in} = V'_{in} + \varepsilon'_{in} + V'_{C_{mn}} + \varepsilon'_{C_{mn}}}$$

The error terms ε'_{in} and $\varepsilon'_{C_{mn}}$ are measurement errors and supposed to be independent.

Considering mobility of pedestrians, and correlations between nests, we used a *cross-Nested Logit Model*, which is a direct extension of the Nested Logit Model, where each alternative may belong to more than one nest.

Similar to the *Nested Logit Model*, the choice set C_n is portioned into M nests C_{mn}. Moreover, for each alternative i and each nest m, parameters $0 \leq \alpha_{im} \leq 1$ representing the degree of 'membership' of alternative *i* in nest m.

The utility of alternative i is given by

$$U_{in} = V'_{in} + \varepsilon'_{in} + V'_{C_{mn}} + \varepsilon'_{C_{mn}} + \ln \alpha_{im}$$

$$\mathrm{P(i}|C_n) = \sum_{m=1}^{M} P(C_{mn}|C_n)P(i|C_{mn})$$

Where

$$\mathrm{P(}C_{mn}|C_n) = \frac{e^{\mu V_{C_{mn}}}}{\sum_{l=1\ to\ M} e^{\mu V_{C_{ln}}}}$$

$$\mathrm{P(i}|C_{mn}) = \frac{\alpha_{im} e^{V'_{in}}}{\sum_{j \epsilon C_{mn}} \alpha_{jm} e^{V'_{jn}}}$$

And $V_{C_{mn}} = V'_{C_{mn}} + \ln \sum_{j \epsilon C_{mn}} \alpha_{jm} e^{V'_{jn}}$

$$P(i|C) = \sum_{m=1}^{M} \frac{\left(\sum_{j \epsilon C} \alpha_{jm}^{\mu m/\mu} y_j{}^{\mu m}\right)^{\frac{\mu}{\mu m}}}{\sum_{n=1}^{M} \left(\sum_{j \epsilon C} \alpha_{jn}^{\mu n/\mu} y_j^{\mu n}\right)} \frac{\alpha_{im}^{\mu m/\mu} y_i^{\mu m}}{\sum_{j \epsilon C} \alpha_{jm}^{\mu m/\mu} y_j^{\mu m}}$$

for each j and m, $\boldsymbol{\alpha_{jm}}$ **,** represents the degree of membership of alternative j to nest m.

3.7.3 Adaptive Discretization

Whereas the behavioural DCM model provides generic description of pedestrian modelling, one key aspect of the model relates to the adaptive discretization of the choices.

The basic elements that are used to define the spatial structure of a pedestrian n are, the current position $\boldsymbol{pos_n} = (\boldsymbol{x_{dn}}, \boldsymbol{y_{dn}})$, the current speed $\boldsymbol{v_{dn}} \boldsymbol{\epsilon} \mathbb{R}$**,** the current direction is $\overrightarrow{d_n} : d_n \in \mathbb{R}^2$ (normalized, so that $\|d_n\|$=1) and the maximum angular 'vision' θ_n. These elements generate the region of interest ($R \subset P$) within the walking plane P, which is situated in front of the pedestrian, where the length of this region represent the maximum progress according to the speed and distance to destination, and the width represents the widest lateral vision of a pedestrian.

Considering speed, we assume that the decision maker (pedestrian) has three different speed regimes that are available to adjust its speed intensity value v_n: **accelerated, constant speed,** and **decelerated** that correspond, respectively, to $v_{acc} = (1+\gamma)v_n$, v_n and $v_{dec} = (1-\gamma)v_n$.

Considering direction, a pedestrian is assumed to have a maximum visual angle $\theta_n = 170°$. Such area is then segmented into 11 radial cones, where one cone capturing the decision not to change the direction (assumed to have an angle of 10°), and 10 cones capturing the decision to change direction, 5 at left of the central cone, and 5 symmetrically defined at the right. The apertures of those cones are not equal. Cones far from the central one have larger angles. Each cone is characterized in the model by its bisecting direction.

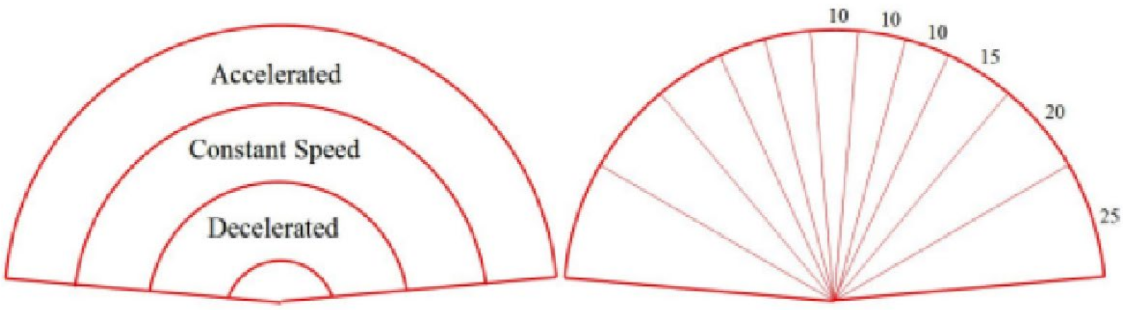

Figure 3.6: Adaptive Space Discretization for Pedestrian Mobility – speed regime (left) and direction regime (right); source: [Antonini, Berlaire, Schneider, Robin, 2009].

The central cone is characterized by the current direction d_n. Each alternative with speed v and direction d is characterized by the physical center of the cell in the space discretization

$$\boldsymbol{c_{vd}} = \boldsymbol{pos_n} + \boldsymbol{v} \cdot \boldsymbol{t} \cdot \boldsymbol{d}.$$

Accordingly, 33 mobility alternatives will be considered for the selection of potential mobility steps, each of the associated the current position and speed of the decision maker.

For each pedestrian, some cells can be declared unavailable because of physical obstacles blocking the cell itself or the access to it. Also, a maximum speed can be assigned to each individual. If a pedestrian is already walking at maximum speed, the cells corresponding to acceleration are declared not available.

3.7.4 Model Implementation

As described in the previous section, the *mobility alternative* of a pedestrian *n*, $\boldsymbol{C_{vdn}}$ corresponds to a **speed regime** $\boldsymbol{v}\epsilon[v_{const}, v_{acc}, v_{dec}]$ and a **direction *dir***. The model will update these microscopic parameters at every time step.

The global pedestrian model is based on five underlying models: **Destination**, **Keep Direction, Free Flow, Follow-the-Leader,** and **Collision Avoidance.** While the last three integrate interactions between pedestrians, the first two consider individual decisions only.

The **Destination** model represents macroscopic selections of destinations, according to pedestrian large-scale schedule, or to entry/exit points of a sidewalk. Within SUMO, entry/exit points could be nested toward a larger scale pedestrian route planning.

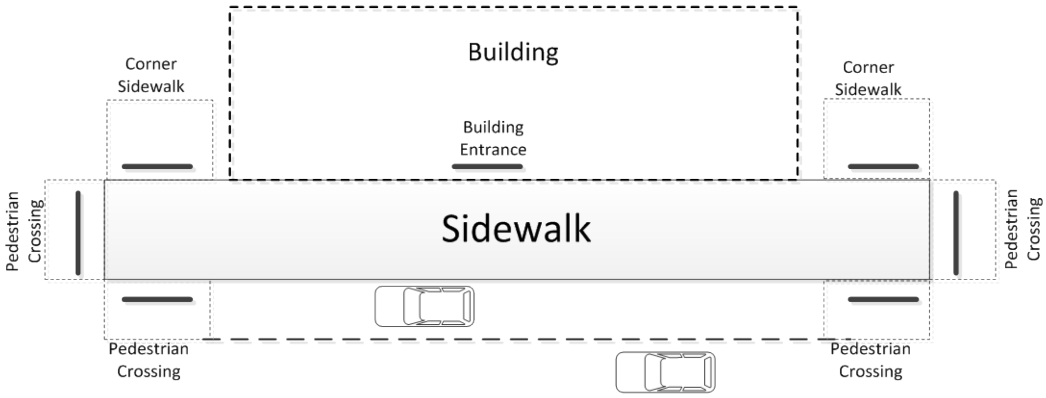

Figure 3.7: Macroscopic Entry/Exit points corresponding to potential destination – two corner sidewalks, 4 pedestrian crossing (two per sidewalk side) and one building access.

The **Keep Direction** model aims at limiting the choice of potential cells to maintain the general direction toward a destination. This model will increase the likelihood to select cells around the direction to destination.

The **Free Flow** acceleration/ deceleration captures the attractiveness of acceleration, respectively deceleration. It sets the speed evolution alternatives considering mobility correlations and ignoring other potential pedestrians Someone who is already walking fast has less incentive to accelerate than someone who is walking slowly.

A **Follow-the-Leader** model captures the attractive interactions among pedestrians. As Car-Following-Models for vehicles, pedestrians would follow another pedestrian's mobility, potentially producing strings of followers. It is described by a sensitivity and stimulus framework. For a given leader, the sensitivity to follow a leader is described by:

$$sensitivity = f(D_L) = \alpha_g^L D_L^{\rho_g^L}$$

D_L Represents the distance between the decision maker and the leader. G={acc, dec] indicating when the leader is accelerating or decelerating.

The decision maker reacts to stimuli coming from the chosen leader. The stimulus is modelled as a function of the leader's relative speed Δv_L and the leader's relative direction $\Delta\theta_L$ as follows:

$$stimulus = f(\Delta v_L, \quad \Delta\theta_L) = {\Delta\theta_L}^{\delta^L} {\Delta v_L}^{\gamma^L}$$

where $\Delta v_L = |v_L - v_n|$ where v_L and v_n are the leader's speed module and the decision maker's speed module, respectively. The variable $\Delta\theta_L = \theta_L - \theta_n$, where θ_L represents the leader's direction and θ_d is the angle characterizing direction d.

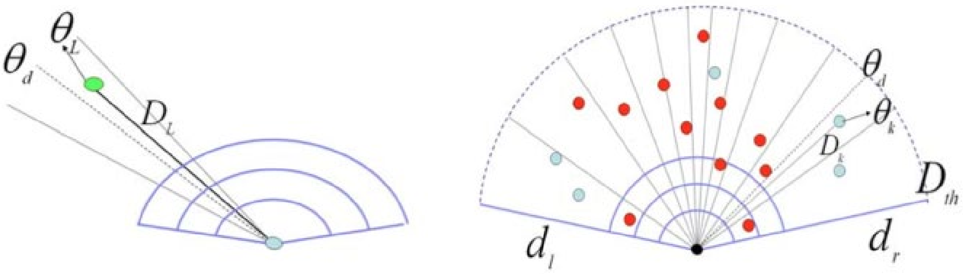

Figure 3.8: Follow-the-Leader model - direction and speed correlation (left), and selection of leaders (red) in each direction (right) (source: [Antonini, Berlaire, Schneider, Robin, 2009]).

When considering a leader, several potential leaders for each direction are considered. An individual k is defined as a potential leader based on the following indicator function I_g^k:

$$I_g^k = \begin{cases} 1, & \text{if } d_l \leq d_k \leq d_r, \\ & \text{and } 0 < D_k \leq D_{th}, \\ & \text{and } 0 < |\Delta\theta_k| \leq \Delta\theta_{th} \\ 0, & \text{otherwise.} \end{cases}$$

Where d_l and d_r represent the bounding left and right directions of the choice set defining the region of interest while d_k is the direction identifying the pedestrian k position. D_k is the distance between pedestrian k and the decision maker, $\Delta\theta_k = \theta_k - \theta_d$ is the difference between the movement direction of pedestrian k and the angle characterizing direction d, the direction identifying the radial cone where individual k lies. The two thresholds D_{th}, and $\Delta\theta_{th}$ are fixed at the values $D_{th} = 5\ D_{max}$ where D_{max} is the radius of the choice set, and $\Delta\theta_{th} = 10°$. A potential leader is assumed to be at a minimum distance $D_L = \min(D_k)$. The indicator function discriminate between accelerated and decelerated alternatives.

A **Collision Avoidance** model captures the repulsive interactions among pedestrians. The scenario is similar to the **Follow the Leader model**, by following a similar sensitivity/stimulus framework where the sensitivity is defined as:

$$sensitivity = f(D_C) = \ \alpha_c e^{-\rho_C \delta_C}$$

where the parameters α_c and ρ_C have to be estimated and D_C is the distance between the collider position and the center of the alternative. The decision maker reacts to stimuli coming from the collider. The stimulus is modelled as a function of two variables:

$$stimulus = f(\Delta v_C, \Delta\theta_C) = {\Delta\theta_C}^{\delta_C} {\Delta v_C}^{\gamma_C}$$

$\Delta\theta_C = \theta_C - \theta_{d_n}$, where θ_C is the collider movement direction and θ_{d_n} is the decision maker movement direction, and $\Delta v_C = v_C + v_n$, where v_C is the collider's speed module and v_n, is the decision maker's speed module. Individuals walking against the decision maker at higher speeds and in more frontal directions (higher $\Delta\theta_C$) generate stronger reactions, weighted by the sensitivity function.

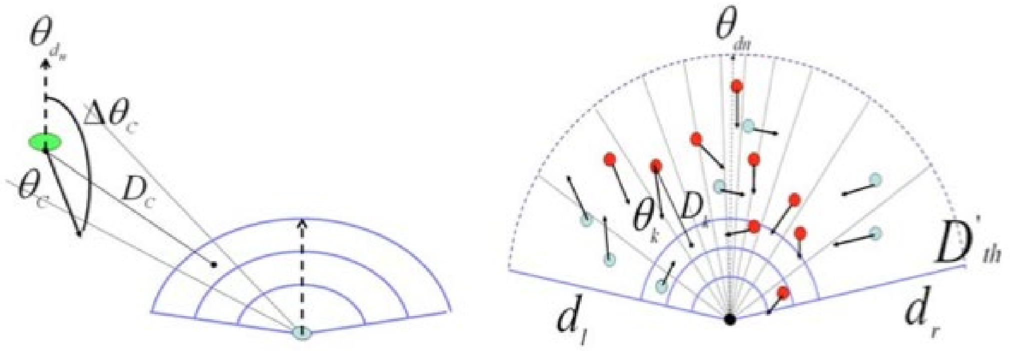

Figure 3.9: Representation of potential collisions from the stimulus function (source: [Antonini, Berlaire, Schneider, Robin, 2009]).

As for a leader, a collider for each direction is chosen considering several potential colliders. An individual k is defined as a potential collider based on the following indicator function:

$$I_C^k = \begin{cases} 1, & \text{if } d_l \leq d_k \leq d_r, \\ & \text{and } 0 < D_k \leq D'_{th}, \\ & \text{and } \frac{\pi}{2} \leq |\Delta\theta_k| \leq \pi \\ 0, & \text{otherwise.} \end{cases}$$

The value of the distance threshold is now fixed to $D'_{th} = 10D_{max}$. The threshold is larger than the one used in the **Follow-the-Leader** model, assuming the collision avoidance behaviour being a longer-range interaction, happening also at a lower density level. We assume an implicit collider choice process, executed by the decision maker herself. Among the set of potential colliders for direction d, the collider is chosen as that individual having $\Delta\theta_C = \max|\Delta\theta_k|$. The related indicator function is Ic. Finally, the collision avoidance term is included in the utility functions of all the alternatives, with the exception of the central ones. So, the indicator function $I_{d,dn}$ is equal to 1 for those alternatives that are not in the current direction and 0 otherwise.

Summarizing, the systematic utility part of the random utility associated with this alternative is defined as [Antonini, Berlaire, Schneider, Robin, 2009]:

$$\begin{aligned}
V_{vdn} = \beta_{\text{dir}}\text{dir}_{dn} \quad &+ \} && \textit{keep direction} \\
\beta_{\text{ddist}}\text{ddist}_{vdn} \quad &+ \\
\beta_{\text{ddir}}\text{ddir}_{dn} \quad &+ && \textit{toward destination} \\
\beta_{\text{acc}} I_{\text{v,acc}} (v_n / v_{\max})^{\lambda_{acc}} \quad &+ \\
\beta_{\text{dec}} I_{\text{v,dec}} (v_n / v_{\max})^{\lambda_{dec}} \quad &+ && \textit{free flow acceleration} \\
I_{\text{v,acc}} I^L_{acc} \alpha^L_{acc} D_L^{\rho^L_{acc}} \Delta v_L^{\gamma^L_{acc}} \Delta\theta_L^{\delta^L_{acc}} \quad &+ \\
I_{\text{v,dec}} I^L_{dec} \alpha^L_{dec} D_L^{\rho^L_{dec}} \Delta v_L^{\gamma^L_{dec}} \Delta\theta_L^{\delta^L_{dec}} \quad &+ && \textit{leader follower} \\
I_{\text{d},d_n} I_C \alpha_C e^{-\rho_C D_C} \Delta v_C^{\gamma_C} \Delta\theta_C^{\delta_C} \quad & \} && \textit{collision avoidance}
\end{aligned}$$

Where all β as well as $\lambda_{acc}, \lambda_{dec}, \alpha^L_{acc}, \rho^L_{acc}, \gamma^L_{acc}, \delta^L_{acc}, \alpha^L_{dec}, \rho^L_{dec}, \gamma^L_{dec}, \delta^L_{dec}, \alpha_c, \rho_c, \delta_c, \gamma_c$ are unknown and must be estimated. We assume here the values estimated by [Antonini, Berlaire, Schneider, Robin, 2009].

The estimated results are measured over 10783 samples from real pedestrian mobility and given in Table 3.4.

Table 3.4: Calibrated Parameters for the Pedestrian Model (from [Antonini, Berlaire, Schneider, Robin, 2009])

Variable	Coefficient estimate	t test 0	t test 1
$\boldsymbol{\beta_{ddir}}$	-0.075	-11.81	
$\boldsymbol{\beta_{ddist}}$	-0.661	-4.06	
$\boldsymbol{\beta_{dir}}$	-0.044	-5.61	
$\boldsymbol{\beta_{acc}}$	-4.06	-14.86	
$\boldsymbol{\beta_{dec}}$	-2.9	-18.30	
$\boldsymbol{\lambda_{acc}}$	0.746	18	
$\boldsymbol{\alpha^L_{acc}}$	4.91	3.27	
$\boldsymbol{\rho^L_{acc}}$	-0.890	-3.78	
$\boldsymbol{\gamma^L_{acc}}$	0.824	9.18	
$\boldsymbol{\alpha^L_{dec}}$	3.96	6.53	
$\boldsymbol{\rho^L_{dec}}$	-0.767	-7.18	
$\boldsymbol{\gamma^L_{dec}}$	0.431	8.25	
$\boldsymbol{\delta^L_{dec}}$	-0.0843	-1.31	
$\boldsymbol{\alpha_c}$	-0.0059	-3.86	
$\boldsymbol{\rho_c}$	-0.603	2.40	
$\boldsymbol{\gamma_c}$	0.287	5.14	
$\boldsymbol{\mu_{const}}$	1.4	11.39	3.26
$\boldsymbol{\mu_{not\ central}}$	1.04	7.05	0.29
$\boldsymbol{\mu_{scale}}$	0.591	-	-210.31

More details related to the model implementation may be found in [Antonini, Berlaire, Schneider, Robin, 2009] and [Bensator, Härri, Spyropoulos, 2014]

3.7.5 Data Structure

Considering the discretization of the potential movement dynamics, an efficient data structure should be implemented to avoid brute-force search of any pedestrian being in a particular cross-nested logit and in a particular cell. We propose an efficient quad-tree data structure. Conceptually speaking, a Quad-tree provides a finer granularity to cells at close distance to the decision maker, and less at far distance. Accordingly, neighbour search may be eased for recurring search (close range), and a penalty might occur for seldom long distance searches.

The quadtree is a generic name for all kinds of trees that are built by recursive division into four quadrants. We implement here a **Point Quadtree**, which divides the space into four rectangles, the input points are stored in the internal nodes of the tree. The four different rectangles are typically referred to as SW southwest, NW northwest, SE southeast and NE northeast. Searching in the point quadtree consists on, whenever a point is included in the search range it is reported and whenever a subtree overlaps with the search range it is traversed.

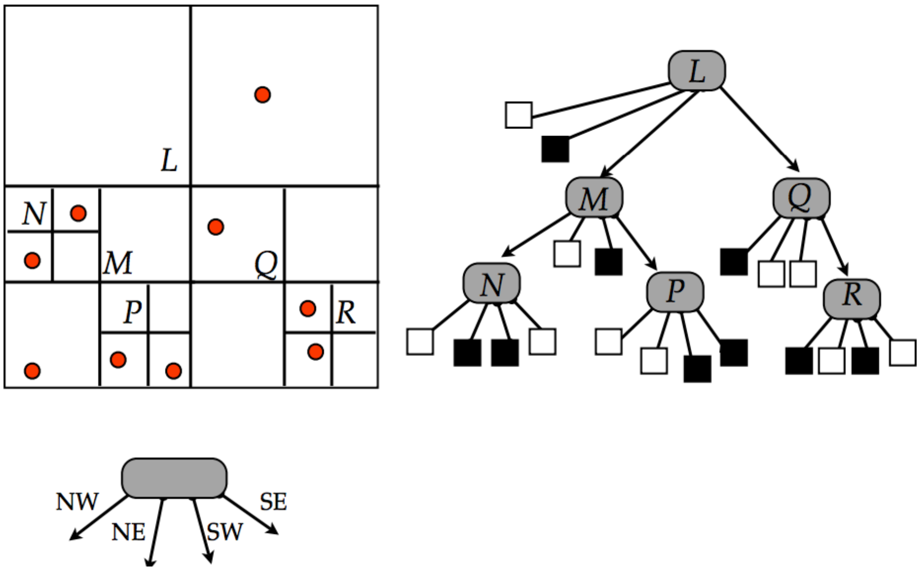

Figure 3.10: Data Representation of a QuadTree structure for pedestrians on sidewalks, where the search for pedestrian could be reduced to three quadrants.

3.7.6 Implementation

The previously described model has been implemented as a first step as an independent simulation to test the model's properties and conformance with the needs of COLOMBO. It will subsequently be integrated into SUMO.

As illustrated in Figure 3.11, the simulation framework is based on three states: **insert**, **update** and **delete**. The insert state corresponds to the creation of pedestrian appearing at one entry/exit point, the selection of the exit point according to an O-D matrix, as well as the selection of an initial speed and direction. The update step corresponds to a recurring update of the pedestrian movement toward its destination according to the previously described models. The delete state corresponds to the pedestrian reaching its assigned exit point. It will be removed from the sidewalk and inserted to the next one or simply be removed from simulation.

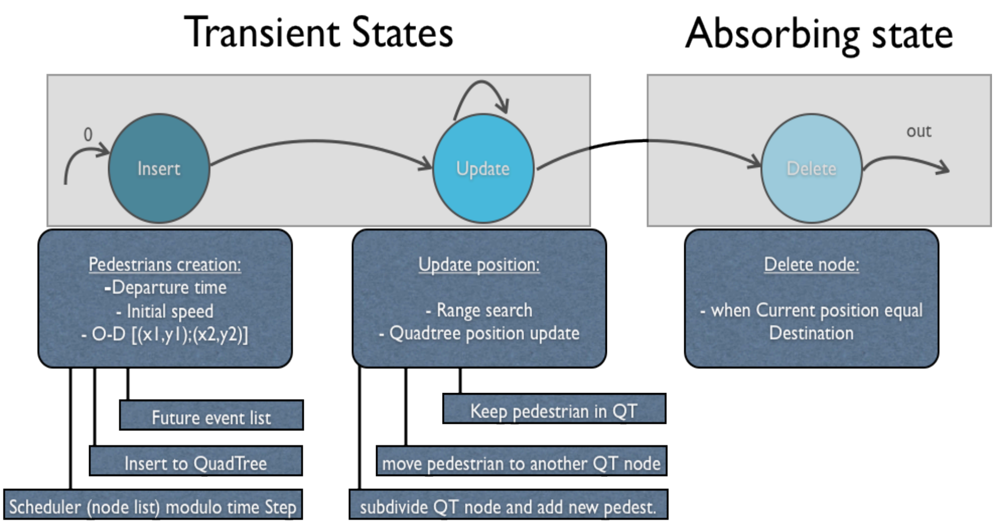

Figure 3.11: Pedestrian Simulation Framework

A UML diagram of the pedestrian mobility models, as well as the general simulation framework is also depicted on Figure 3.12. The **Entry**, **Spatial Discretization**, and **OD-Matrix** blocks correspond to the main function of the behavioral pedestrian mobility model, respectively for initialization, microscopic, and demand modeling.

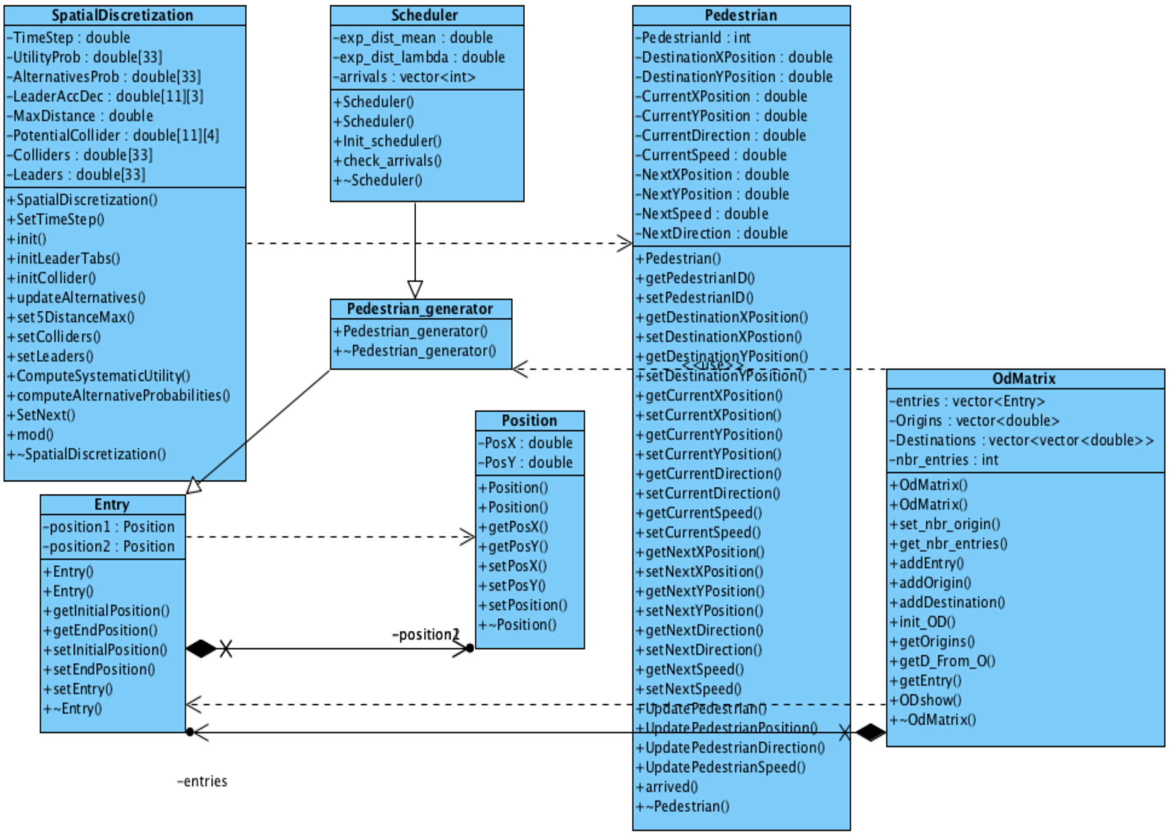

Figure 3.12: UML Diagrams of the implemented Pedestrian Mobility Model

The test scenario corresponds to a single sidewalk as depicted in Figure 3.13. It contains 7 entry/exit points, between which pedestrians will find the most direct path. The width might seem a bit large, but it has been purposely chosen to test the model by leaving enough freedom for pedestrian also in the lateral movement. Subsequent scenarios will assume a sidewalk width of 2 m.

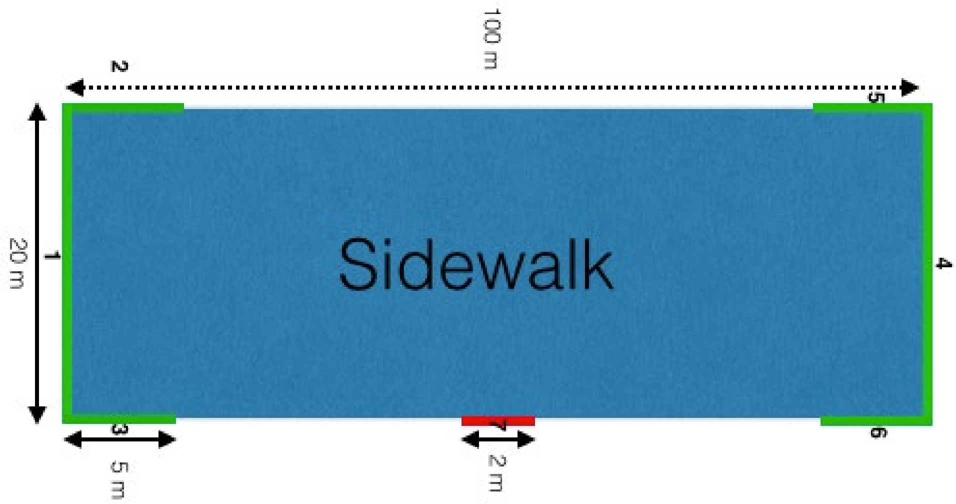

Figure 3.13 Single Sidewalk Scenario, with 7 entry/exit points

Finally, given an entry point, the probability of choosing one of the 7 exit points are given by

$$\{\mathbf{0.2}, \mathbf{0.1}, \mathbf{0.1}, \mathbf{0.2}, \mathbf{0.1}, \mathbf{0.2}, \mathbf{0.1}\}$$

The full O-D matrix represents such probability list for each entry point. We therefore assume in this phase that the probabilities to reach the 7 exit points are equal regardless of the entry point. In a latter phase, we will alter the O-D matrix to show a higher chance to special macroscopic movements. More details related to the model implementation may be found in [Bensator, Härri, Spyropoulos, 2014]

Finally, Figure 3.14 shows a screenshot of a typical pedestrian spatial distribution generated by the behavioural model. From each blue point corresponding to a pedestrian, we can 'guess' the location and dimension of the sidewalk. We could also observe that spatial restrictions corresponding to the sidewalk dimensions have been respected.

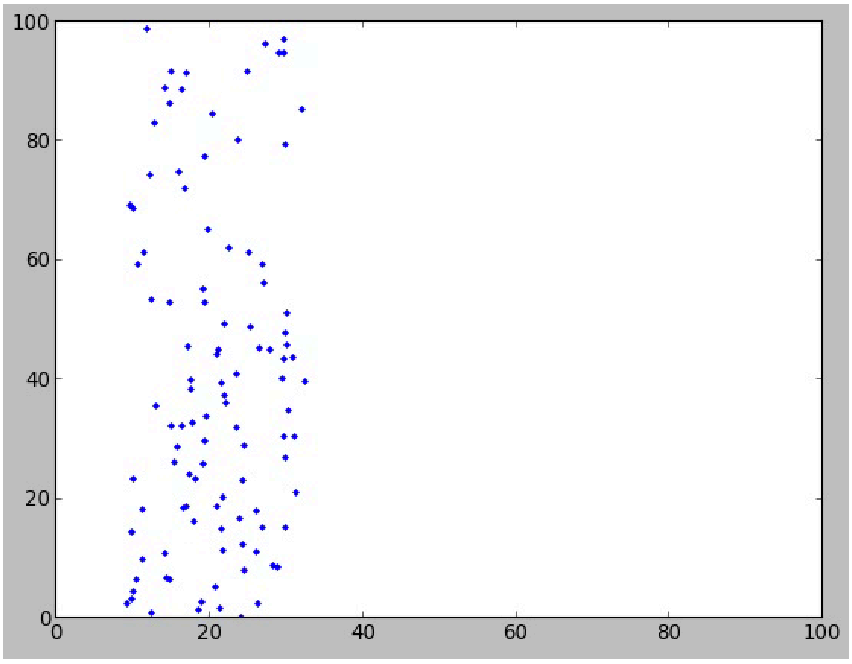

Figure 3.14: Pedestrian Spatial Distribution on a Sidewalk according to the proposed behavioral model

3.7.7 Usage in the Scope of COLOMBO and beyond

As SUMO is a mature tool, larger extensions need to be done in a manner that preserves the basic functionality. The additionally developed standalone application allows a fast adaptation of its core functionalities, allowing to experiment with needs and possible solutions while starting to investigate pedestrians.

The application was designed to be an intermediate tool with aforementioned purposes and was already used for first investigations as reported in section 4.4 and section 4.5. Different possibilities for future use exist. Being implemented in c++, it should be easy to integrate the model into SUMO. As well, the standalone application could be extended to be a full (in terms of user-friendliness) tool for pedestrian simulations.

4 Experiments

The main goal of the described extensions was to describe the interactions between vehicles and other modes of traffic. To obtain a quantitative assessment of these interactions some experiments were conducted using the striping model as implemented in SUMO. These are described in the following.

4.1 Interactions between right-turning Vehicles and crossing Pedestrians

In this experiment, a saturated flow of right-turning vehicles arrives at a single intersection. To complete the right turn, this flow must pass a pedestrian crossing which is frequented by a binomially distributed pedestrian flow of variable strength. The synthetic intersection used for this experiment is shown in Figure 4.1.

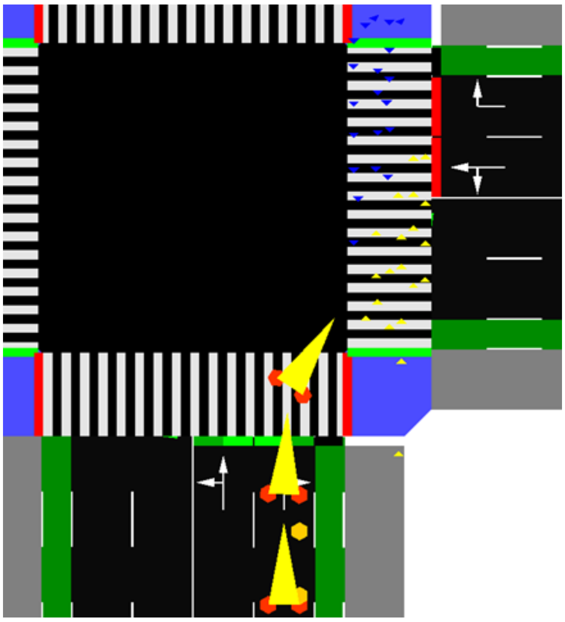

Figure 4.1: Simulation experiment for measuring the relationship between pedestrian flow and right turning vehicle flow (TLS-controlled intersection).

The vehicular and pedestrian models were parameterized as given in Table 4.1.

Table 4.1: Parameter values used for vehicles and pedestrians

Model	Attribute	Value
Vehicles	length	4.5 m
Vehicles	minGap	1.5 m
Vehicles	speedDev	0.1
Pedestrians	Length	0.3 m
Pedestrians	Width	0.6 m
Pedestrians	minGap	0.3 m
Pedestrians	maxSpeed	1.4 m/s

We have measured the vehicular flow behind the crossing in dependence on pedestrian density. The results are shown in Figure 4.2. At low and medium pedestrian flows, the uncontrolled intersection allows for higher flows due to the absence of “red” phases. However, at high pedestrian flows the TLS-controlled intersection allows for higher vehicle flows because vehicles already waiting within the intersection may drive each time, pedestrians have to wait at the red light.

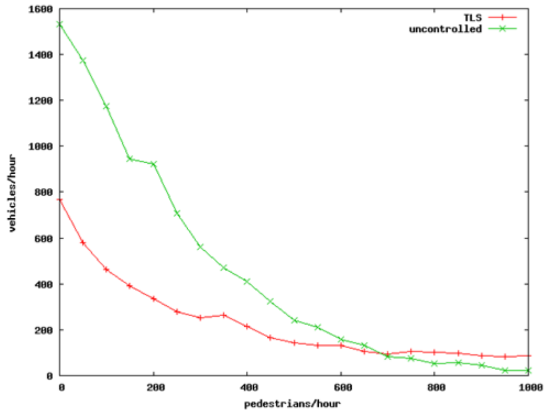

Figure 4.2: Flow of right-turning vehicles as a function of increasing pedestrian flow.

4.2 Interactions between right-turning Vehicles and straight-going Bicycles

In this experiment, a saturated flow of right-turning vehicles arrives at a single intersection. At the same time a straight-going flow of bicycles coming from the same direction is passing the intersection. To complete the right turn, vehicles must wait for a gap in the flow of bicycles. The bicycles arrive according to a binomial distribution with varying frequency. The synthetic intersection used for this experiment is shown in Figure 4.3.

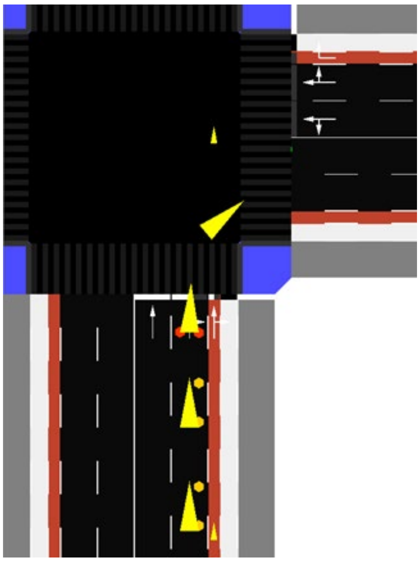

Figure 4.3: Simulation experiment for measuring the relationship between bicycle flow and right turning vehicle flow (uncontrolled intersection).

The vehicular and bicycle models were parameterized as given in Table 4.2.

Table 4.2: Parameter values used for vehicles and bicycles

Model	Attribute	Value
Vehicles	length	4.5 m
Vehicles	minGap	1.5 m
Vehicles	speedDev	0.1
Bicycles	Length	1.6 m
Bicycles	Width	0.6 m
Bicycles	minGap	0.5 m
Bicycles	maxSpeed	4.2 m/s

We have measured the vehicular flow behind the crossing in dependence on bicycle flow. The results are shown in Figure 4.4. Vehicle flow decreases when bicycle flow grows and the relative interference (the slope of the curves) is similar when considering a TLS-controlled and an uncontrolled intersection. Vehicle flow at the TLS-controlled intersection shows slightly asymptotic behavior due to a short green phase which is exclusive to the right turning vehicles.

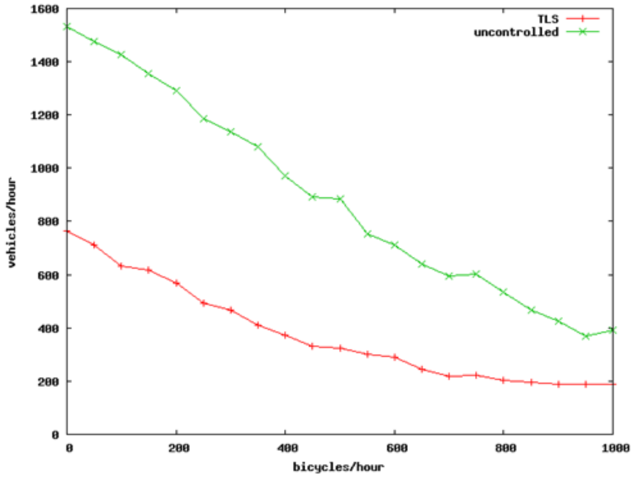

Figure 4.4: Flow of right-turning vehicles as a function of increasing bicycle flow.

4.3 Influence of Pedestrians on an urban Vehicular Scenario

In this experiment an urban vehicular simulation scenario was extended with pedestrian traffic. The simulation scenario named *ACOSTA* comes from the iTETRIS project and models a part of the city of Bologna. It contains 9045 vehicle movements within the space of about 90 minutes in an area of 1.5km^2 and is characterized by high traffic density. The network model consists of 179 nodes and 182 edges.

Figure 4.5: ACOSTA scenario with pedestrian enhancements. Pedestrians are shown at exaggerated size to increase visibility.

To extend this scenario, sidewalks and pedestrian crossings were added to the network model using the NETCONVERT options `--sidewalks.guess` and `--crossings.guess`. A total of 182 sidewalks (1 for each edge) and 164 pedestrian crossings were generated. Of these crossings, 52 are controlled by traffic lights. The existing traffic light programs were modified automatically to also cover the generated crossings. Minor manual changes to traffic light programs were necessary because some connections which are uncontrolled in the original scenario intersect with pedestrian crossings in the new scenario and thus need control information as well. Pedestrian demand was generated randomly using the tool *randomTrips.py* described in section 3.3.6. 3600 pedestrians were generated which enter the network with a spacing of 1 second and then proceed to their destination along the shortest route. The scenario is shown in Figure 4.5.

To measure the influence of the pedestrians on vehicular traffic, we have compared the duration of vehicular trips in both versions of the scenario. Figure 4.6 shows the histogram of trip durations with a binning size of 60 seconds. It can be seen that the overall shapes of the distributions are similar but a small number of trips with much higher durations exist in the pedestrian scenario. Figure 4.7 shows the changes in trip durations for individual drivers. There is an average increase of trip durations of 22 seconds but only a median increase of 1 second. Thus the increase in trip durations is caused by a small proportion of the overall population. It is assumed that this is caused by minor break-downs in traffic flow in the pedestrian scenario due to the already high level of traffic in the original scenario. Such effects could plausibly happen in real life and reproducing them in a simulation model is therefore desirable.

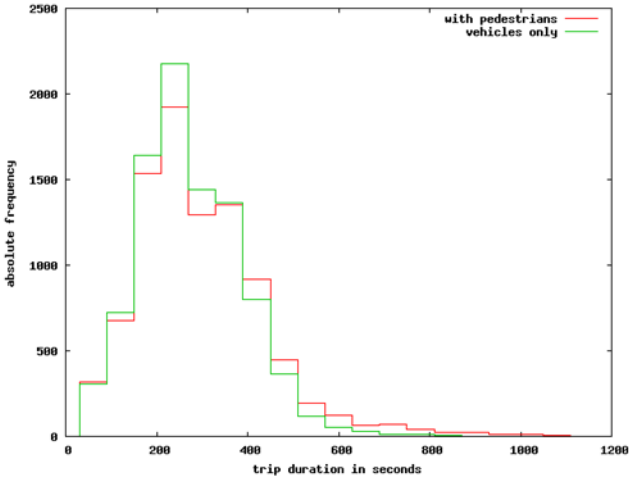

Figure 4.6: Histogram of vehicle trip durations in both versions of the ACOSTA scenario

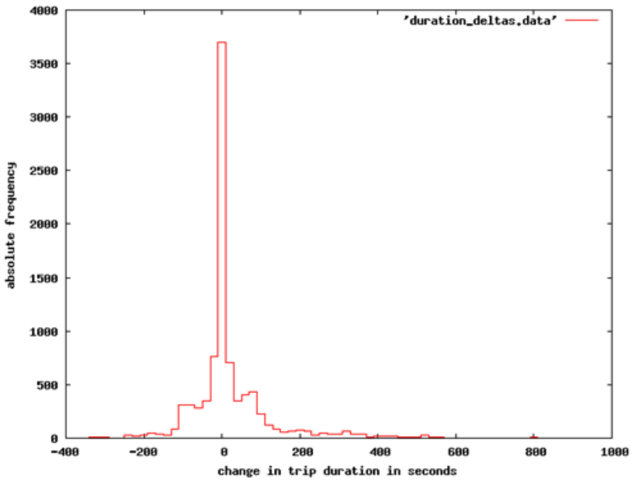

Figure 4.7: Histogram of changes in trip duration.

4.4 Behavioural Model Performance

We designed several scenarios to illustrate the particular mobility patterns and topology created by pedestrians on urban sidewalks. Three major parameters may be tweaked:

- Width of the sidewalk – according to the width of the sidewalk, pedestrian may be more likely to encounter other pedestrian, and would need to reduce speed and avoid the obstacle.
- Pedestrian Arrival Rate – An increasing arrival rate increases the total number of pedestrians on the sidewalk and accordingly, increases the probability to encounter other pedestrians
- O-D Matrices – pedestrians may not aim to go from a south to a north entrance of a sidewalk, and multiple combination of Source-Destination corresponding to the entry/exit point of a sidewalk could be devised.

The experiments were performed using the standalone pedestrian simulation application described in section 3.7.

4.4.1 Scenario Descriptions

Figure 4.8 illustrates a basic sidewalk scenario. We first design the O-D matrices according to Table 4.3 and Table 4.4, where ‘O’ and ‘D’ represents Origin and Destination respectively, and where the numerical values represent arbitrary probabilities to appear at a particular origin or move from on origin to a destination.

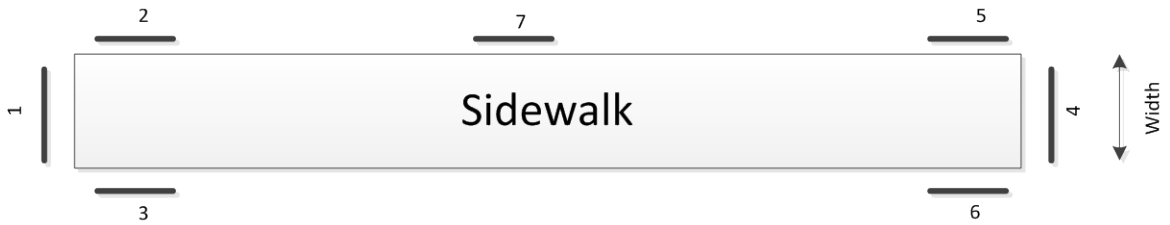

Figure 4.8: Generic Sidewalk with 7 Entry/Exit Points

Table 4.3: Probabilities to be an origin

Origins	O1	O2	O3	O4	O5	O6	O7
probability	0.2	0.1	0.1	0.2	0.1	0.2	0.1

Table 4.4: From Origin to Destination

Origins	D1	D2	D3	D4	D5	D6	D7
O 1	0.	0.1	0.1	0.4	0.2	0.3	0.
O 2	0.	0.	0.15	0.3	0.1	0.3	0.15
O 3	0.1	0.15	0.	0.4	0.2	0.1	0.05
O 4	0.5	0.2	0.2	0.	0.	0.	0.1
O 5	0.3	0.	0. 2	0.2	0.	0.1	0.2
O 6	0.3	0.35	0.1	0.15	0.1	0.	0.
O7	0.3	0.2	0.1	0.1	0.3	0.	0.

We design our scenario with three different sidewalk widths (where the lengths remains 100m in all scenarios):

- **20 m –** represents typical pedestrian streets, where vehicular lanes have been converted to a street-wide sidewalk
- **5 m –** represents a large sidewalk as what we can find in Champs Elisée in Paris for instance
- **2 m –** represents typical sidewalk widths in major urban cities

Finally, we control pedestrian densities by varying their arrival rates according to 2, 3, 5, and 10 pedestrians/sec.

We will depict the pedestrian sidewalk mobility patterns with two figures:

- **Pedestrian Spatial Distribution –** depicts how pedestrians are distributed in space on the sidewalk, illustrating potential non-uniform distributions
- **Average Pedestrian Speed –** depicts the impact of the interactions between pedestrian, from the desired speed to the 'real' speed.

4.4.2 Pedestrian Sidewalk Mobility Patterns

Figure 4.9 illustrates pedestrian spatial patterns for sidewalk width of 20 m, with arrival rate of 2, 3, 5, 10 ped/s resp. We can see from the figures that at low arrival rate, the inter-distance between pedestrians is rather large, and they will spend most of their time in follow-the-leader mode, as collision will be highly unlikely. But when the pedestrian density increases, the pedestrian distribution start showing accumulation points either on the entry/exit points or on the sides of the sidewalk. Both are side-effects of the model. When a large amount of pedestrian enter from three entry/exit points of the same side (E/S 1,2,3 for instance), collisions are more likely to occur than if pedestrians would be appear at a single entry/exit point. Accordingly, non-homogeneous patterns will appear. Also, the accumulation points on the side aspects are also as the increasing 'collision avoidance' tries to make pedestrian change course by making them go 'side-ways'.

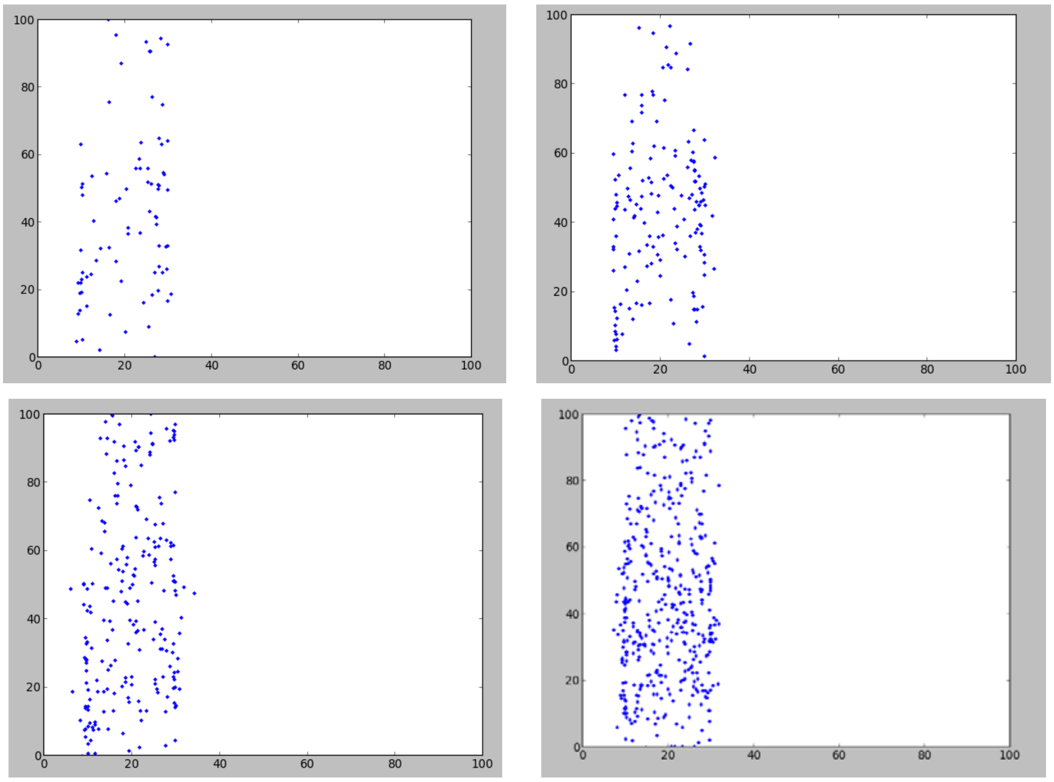

Figure 4.9: Pedestrian Spatial Patterns for Sidewalk width of 20 m, with arrival rate of 2, 3, 5, 10 ped/s resp.

Figure 4.10 depicts pedestrian average speed for sidewalk width of 20 m, with arrival rate of 2, 3, 5, 10 ped/s resp. We configured the pedestrian 'desired' speed (**target speed**) to follow a **normal distribution** with **1.37 m/s mean and 0.34 standard deviation**. We can see from this figure that the results are coherent with the desired mean, as the average speed over the 1000 s second simulated mobility falls within the 1.0 m/s and 1.73 m/s interval.

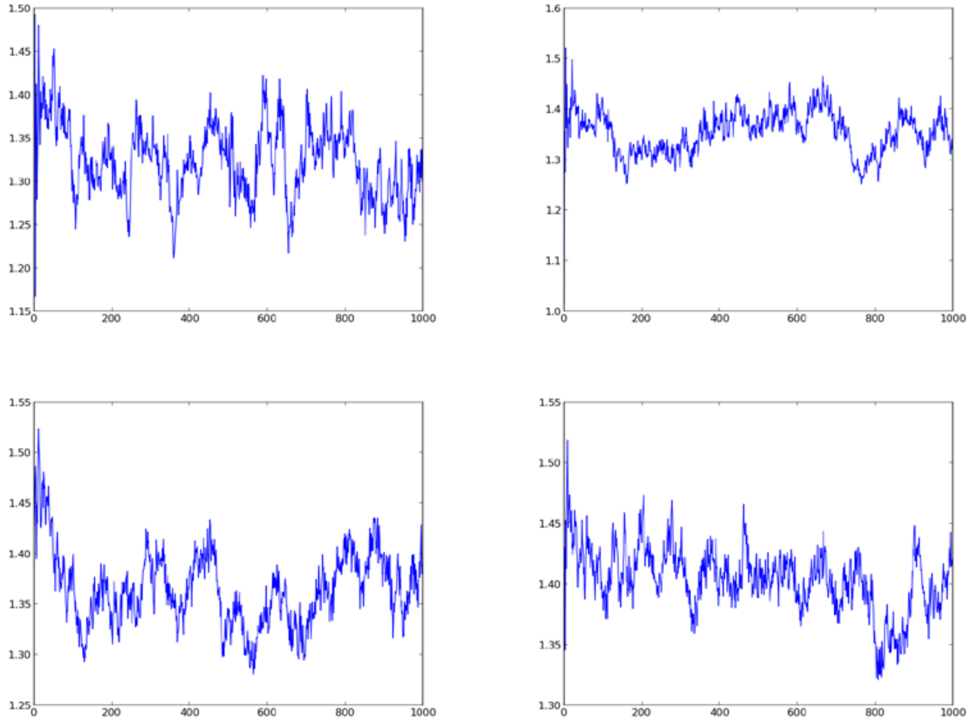

Figure 4.10: Pedestrian Average Speed for Sidewalk width of 20 m, with arrival rate of 2, 3, 5, 10 ped/s resp.

One interesting aspect is that the average speed tends to increase when we increase the pedestrian mobility, which could be seen as counter-intuitive. This can be explained by the collision avoidance phase of our model, which tends to make pedestrian walk faster to avoid another pedestrian, rather than slowing down. This is typically possible, as there is enough space to follow this rule.

On the next set of figures, we reduced the sidewalk width to 5 m. It becomes immediately visible that already at medium pedestrian arrival rate, the spatial distribution already shows non-homogeneous patterns, and more important a lower inter-distance between pedestrians. In this configuration, pedestrians will spend more time in collision avoidance than follow-the-leader modes.

It can be particularly noted that for 10 ped/s arrival rate, the sidewalk is saturated by pedestrians, which makes collision avoidance difficult, and makes a few pedestrians violate the pedestrian width and go on the road. Although extremely dangerous in reality, this is also typically observed in dense urban sidewalks.

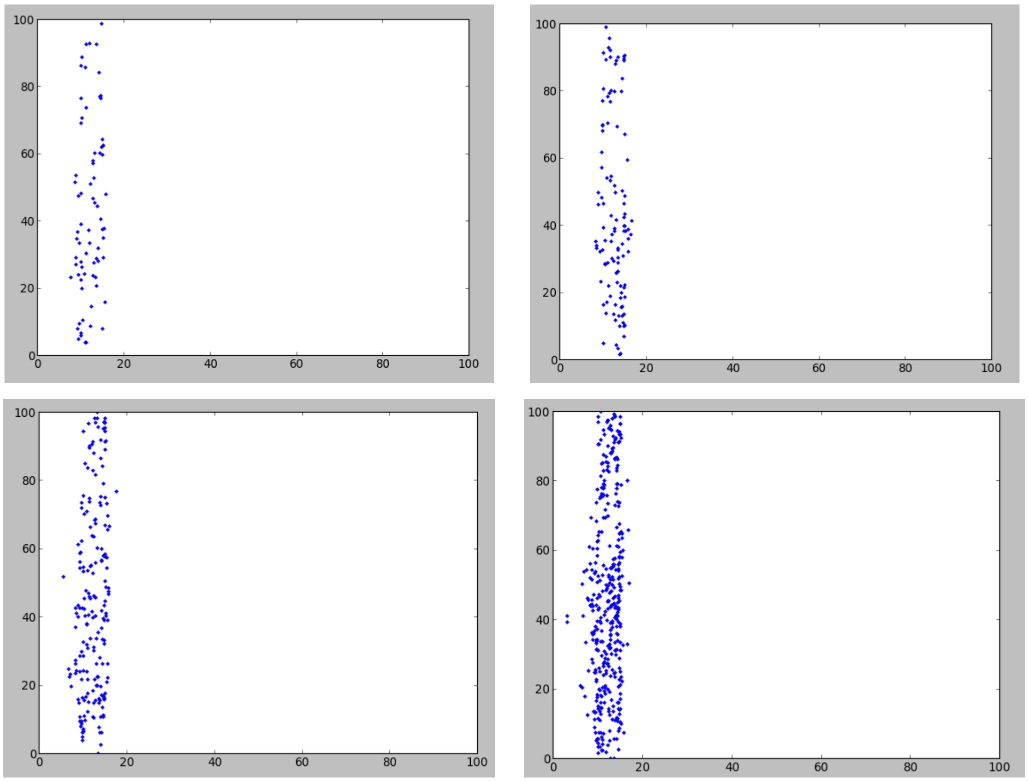

Figure 4.11: Pedestrian Spatial Patterns for Sidewalk width of 5 m, with arrival rate of 2, 3, 5, 10 ped/s resp.

On Figure 4.12, we represent the average speed for a sidewalk width of 5 m and with pedestrian arrival rate of 2, 3, 5, 10 ped/s resp. From these results, we can see that the average speed shows a higher oscillation around the average, most likely due to moving more often to the collision avoidance phase when two pedestrians have colliding courses. Here again, we can observe a slight speed increase by increasing the density, again mostly due to the collision avoidance phase.

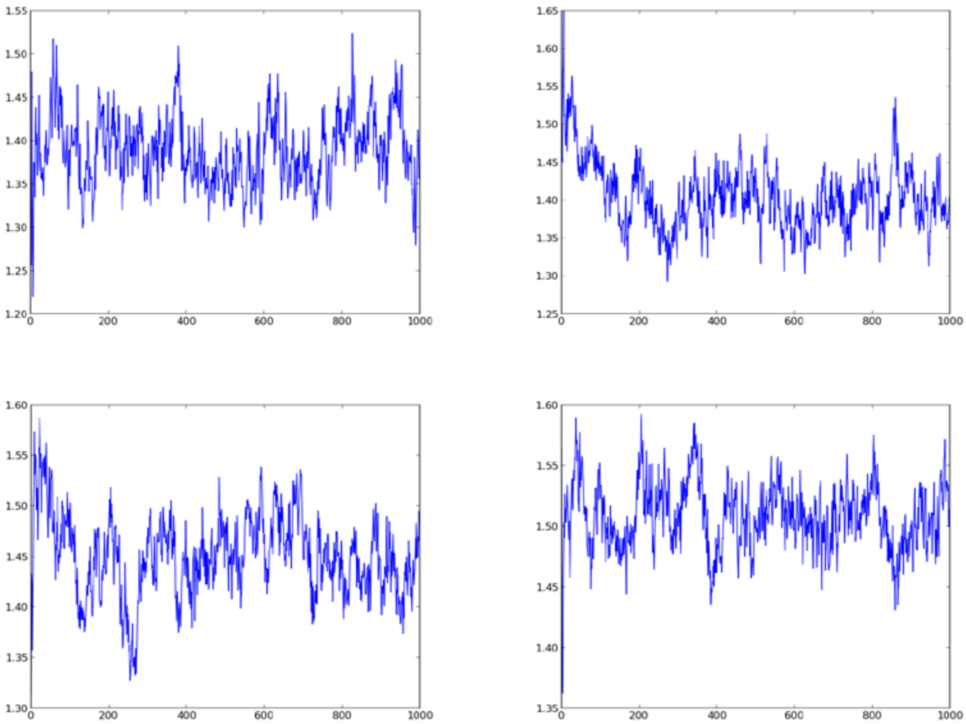

Figure 4.12: Pedestrian Average Speed for Sidewalk width of 5 m, with arrival rate of 2, 3, 5, 10 ped/s resp.

Finally, on the last set of results, we further reduced the sidewalk width to 2 m and depicted on Figure 4.13 the pedestrian spatial distribution for pedestrian arrival rates of 2, 3, 5, 10 ped/s respectively. The behaviour we illustrated for a sidewalk width of 10m is here further visible, as already at 2 ped/s arrival rate, the distribution is well spread on the sidewalk length. When increasing to 3 or even 5 ped/s, we already see pedestrians violating the sidewalk rule and move on the street to avoid collisions. We also start observing a non-uniform distribution of pedestrians on the south side of the sidewalk. This is due to the O-D matrix used for our simulation, and to the collision avoidance phase that tends to make pedestrians make smaller eclectic movements.

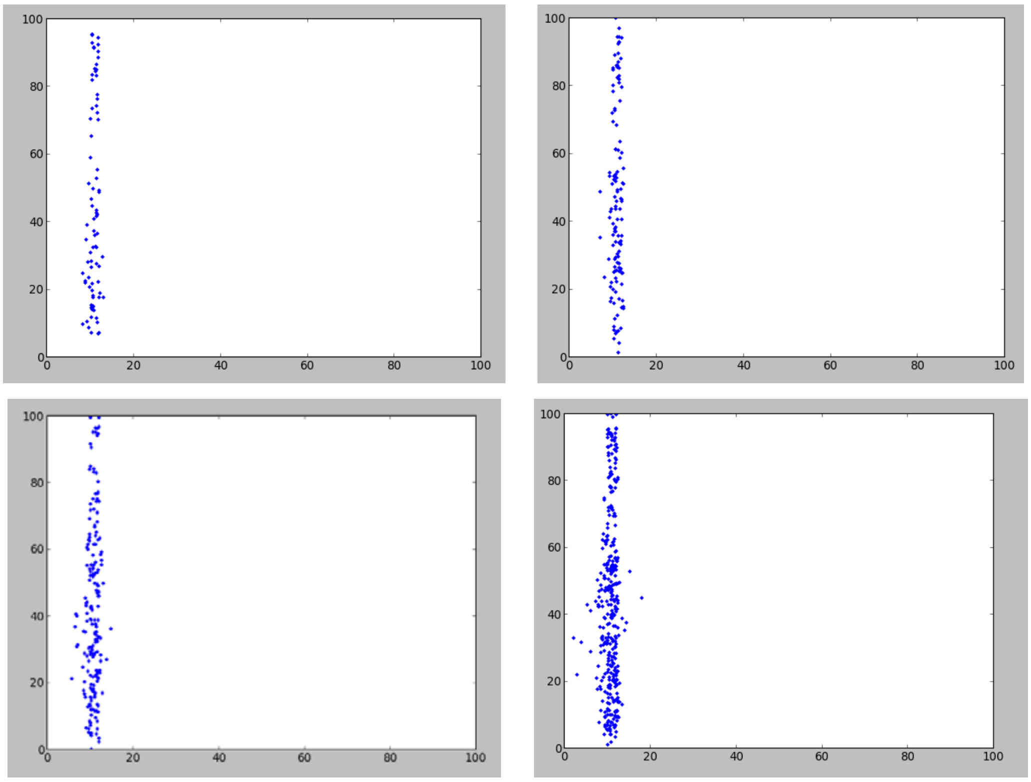

Figure 4.13: Pedestrian Spatial Patterns for Sidewalk width of 2 m, with arrival rate of 2,3,5,10 ped/s resp.

Figure 4.14 illustrates the corresponding average speeds for pedestrian arrival rates of 2, 3, 5, 10 ped/s respectively. We can observe here the same behaviours as in the other figures, but by relying more on a collision avoidance phase rather than follow the leader, pedestrian will show less variation of their average speed as they will align with that of their preceding pedestrians.

To conclude this part, we proposed a microscopic behavioural model for pedestrian mobility, which is capable of modelling the microscopic interactions between pedestrians both spatially and from the speed distributions. We illustrated the particular spatial distribution that pedestrians encounter on sidewalks, and also the impact of an increasing pedestrian density on average speed. Although it would still need to be validated through calibration, this model shows close-to-reality pedestrian mobility patterns and can provide a sufficient level of precision to be used by COLOMBO, in particular for V2X traffic monitoring.

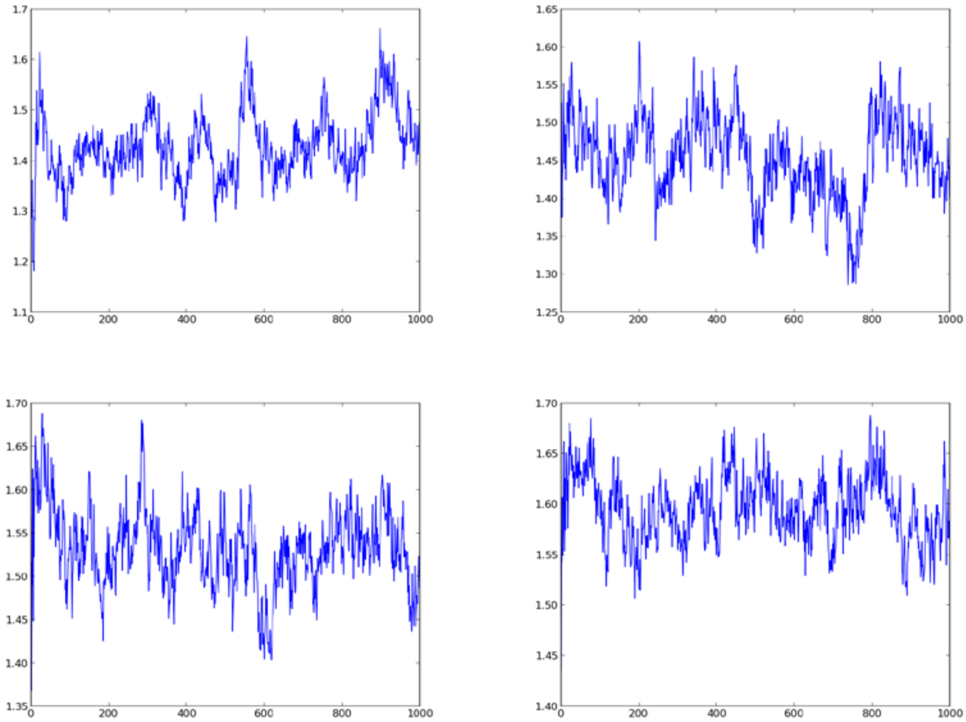

Figure 4.14: Pedestrian Average Speed for Sidewalk width of 2 m, with arrival rate of 2, 3, 5, 10 ped/s resp.

It should finally be noted that we initially implemented the model as a standalone application, but will now work to integrate it into SUMO.

4.5 Pedestrians and Bicycles from V2X Perspective

Dedicated Short Range Communication (DSRC) is the standard for dedicated inter-vehicular communications (V2X), and is a promising technology to improve communications for Intelligent Transportation Systems (ITS). With the recent bandwidth allocation of 30 MHz at 5.9 GHz for ITS in Europe, V2X communication systems are close to market introduction.

In the early deployment stage of communication systems for ITS, the penetration rate of V2X-equipped vehicles should be assumed to be not sufficient to provide the necessary connectivity and traffic sensing required by several ITS applications, e.g., traffic density estimation. At the same time, we are witnessing a very rapid growth in the use of smartphones that drivers as well as pedestrians are adopting more and more. Compared to DSRC-equipped vehicles, smartphones have not been designed to provide the same level of sensing precision, but have the merit of being there and available to complement and enrich data obtained from V2X-equipped vehicles to satisfy the ITS application requirements.

Smartphones have a double benefit in a V2X perspective. First, they integrate a wider range of similar technologies than DSRC-equipped vehicles. In particular, they are also capable of dedicated direct communications either using their Bluetooth or WiFi-Direct interfaces. Second, they are not only linked to the automotive market and, first have a short market turnover (~ 2 years), and second are also owned by other road actors, such as pedestrians and bicycles.

These other road actors have not been considered in general V2X applications, even though they are generally called ‘vulnerable’ traffic users and should be the primary target of ITS traffic safety applications. But their role is not limited to road safety, as due to their peculiar mobility characteristics – bicycles are only merely linked to the flow dynamics created by cars, and pedestrians have a slow motion of protected sidewalks. In both cases, they are expected to by valuable 'observers' of urban traffic. For traffic monitoring, considering their slower speed, bicycles and pedestrians could be more efficiently considered as a 'static' point observer of vehicular traffic flows and can better estimate traffic volumes at given locations. To some extends, they could be considered as slow moving digital 'loop detectors'. And in traffic jam conditions, pedestrians and bicycles are capable of 'slowly' moving upstream, which could let them estimate the size of a traffic jam, without the need of cooperative communications between vehicles. The benefits and impact of pedestrians and bicycles on traffic monitoring applications is subject to investigation in WP1.

Pedestrian and bicycles are yet not only linked to traffic flow and mobility and can also help disseminating V2X data between V2X- or Smartphone-equipped vehicles and traffic lights. Pedestrians (in particular) can interconnect and form an ad-hoc spontaneous network and allow multi-hop data exchange to reach a traffic light. Channeled to follow the same stream (up- or down-), pedestrian could also be used as data mules to reach the upstream or downstream traffic lights, considering delay tolerant applications. From their particular topology, pedestrian walking on sidewalk are therefore expected to create a very particular topology that is worth investigating.

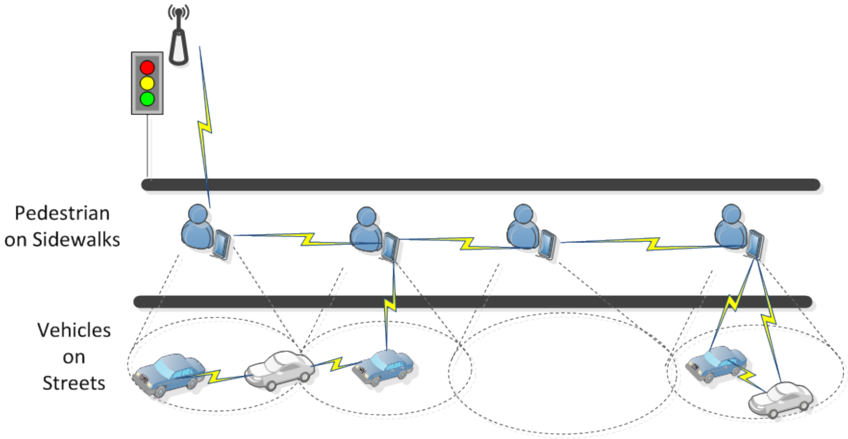

Figure 4.15: V2X benefits of smartphone-equipped pedestrians, where pedestrians may first monitor traffic in given zones, and second where data may be disseminated between smartphone to traffic light or between vehicles over pedestrians.

In the remainder of this section, we will illustrate the graph properties of pedestrian 'communicating' networks, and demonstrate their capabilities in data dissemination for traffic monitoring.

Using WiFi-Direct or Bluetooth interfaces of their smartphones, pedestrians are capable of self-organizing in a communication ad-hoc network. Such network, known as *Mobile Ad-Hoc Networks (MANET)*, may be represented as a graph, where vertices are smartphones-equipped pedestrians, and edges are communication links.

Without loss of generalities, we represent the available communication links of a smartphone using a *Unit Disk Graph (UDG)* concept, where every smartphone being and remaining in a communication range are assumed to have a direct link.

In this section, we arbitrarily represent the communication range in two values for all pedestrians in any condition:

- **4 m radius –** this represents the Bluetooth technology's maximum operation range, as well as WiFi-Direct comfortable range.
- **10 m radius –** this represents the WiFi-direct potential range

Longer ranges have also been investigated and possible – during field tests, we could reach up to 50m communication range in a static configuration - but they do not bring more connectivity considering the pedestrian scenarios used in this document.

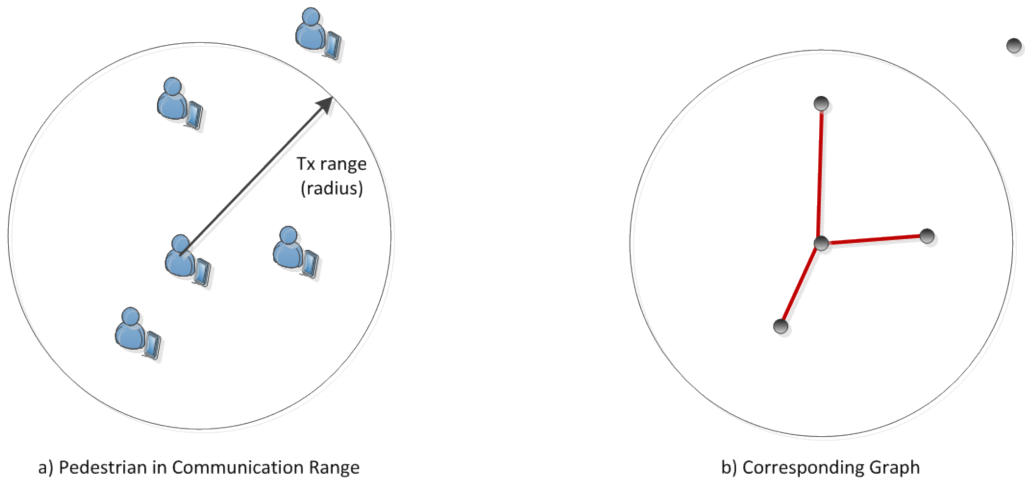

Figure 4.16: Pedestrians graph representation

In the results depicted below, we used the behavioral pedestrian model developed for COLOMBO and described in details in Section 3.7.

We design our scenario with three different sidewalk widths (where the lengths remains *100 m* in all scenarios):

- **20 m –** represents typical pedestrian streets, where vehicular lanes have been converted to a street-wide sidewalk
- **5 m –** represents a large sidewalk as what we can find in Champs Elisée in Paris for instance
- **2 m –** represents typical sidewalk widths in major urban cities

We control pedestrian densities by varying their arrival rates according to 2, 3, 5, and 10 pedestrians/sec in entry/exit points.

We provide in the following figures the connectivity representation of pedestrian mobility. Figure 3 represents such connectivity for a Unit-Disk Graph (UDC) of **4 m radius**, for sidewalk width of 2m, and a pedestrian arrival rate of 2, 3, 5, 10 ped/s respectively. As we wanted to illustrate the graph connectivity, we chose a small sidewalk width. The red edges represent the communication links between pedestrians.

We can observe that due to such small width, and the corresponding promiscuity between pedestrians, the connectivity is sufficient for V2X communication already for 4 m communication range. Considering low pedestrian arrival rate, we do not have a giant component, but large clusters exist between pedestrians. It should be noted that due to its dynamic aspects, these clusters will evolves and some pedestrians will bridge different clusters, allowing V2X communication over multiple clusters. When considering large arrival rates, the whole sidewalk is connected and multi-hop V2X communications between any pedestrian to a sidewalk could be conducted. This first set of results show that assuming such pedestrian density, multi-hop Bluetooth or short-range WiFi-direct could be considered for data dissemination.

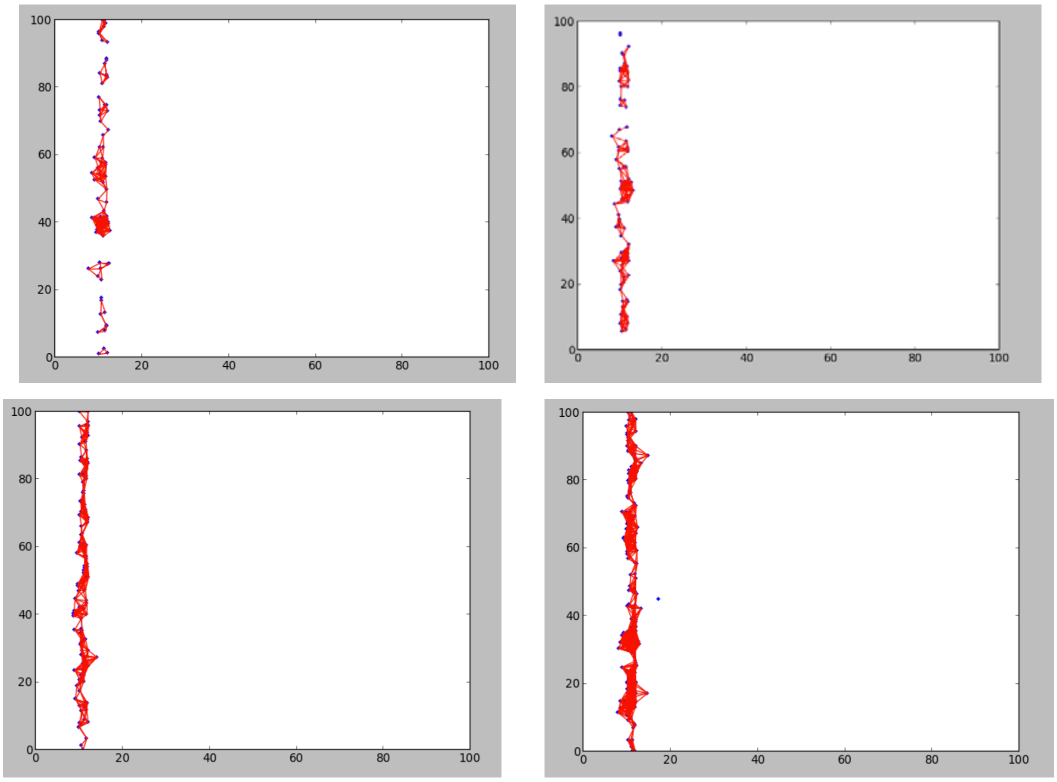

Figure 4.17: Pedestrian Communication Graphs, for a Unit-Disk Graph (UDC) of 4m range, sidewalk width of 2m, and pedestrian arrival rate of 2, 3, 5, 10 ped/s respectively.

When considering a larger communication radius of 10 m as shown in Figure 4.18, the connectivity is larger, and a giant component already appears at low penetration rate. It should be noted that 10m communication radius should be considered as a reasonable transmit range for traffic monitoring, and from such connectivity graph, multi-hop data dissemination between pedestrians will be possible.

In these results, we assumed that all pedestrians are equipped with smartphones and willing to participate to traffic monitoring. This is highly unlikely to be the case in real deployment, as first not all smartphones will have the appropriate interfaces, and second pedestrians may be reluctant to participate in order to save battery lifetime. Accounting for such reduced smartphone penetration, and for low cooperative behaviour may be done by assuming a lower link penetration. As a lot of links are redundant already at 5 ped/s, it is expected that the overall connectivity graph will not change much for pedestrian arrival rate between 5 and 10 ped/s resp. On lower arrival rates, it could be expected that smaller clusters will be visible, and the pedestrian natural mobility would be used to bridge the disconnected clusters and store-carry-forward data dissemination preferred compared to connected multi-hop.

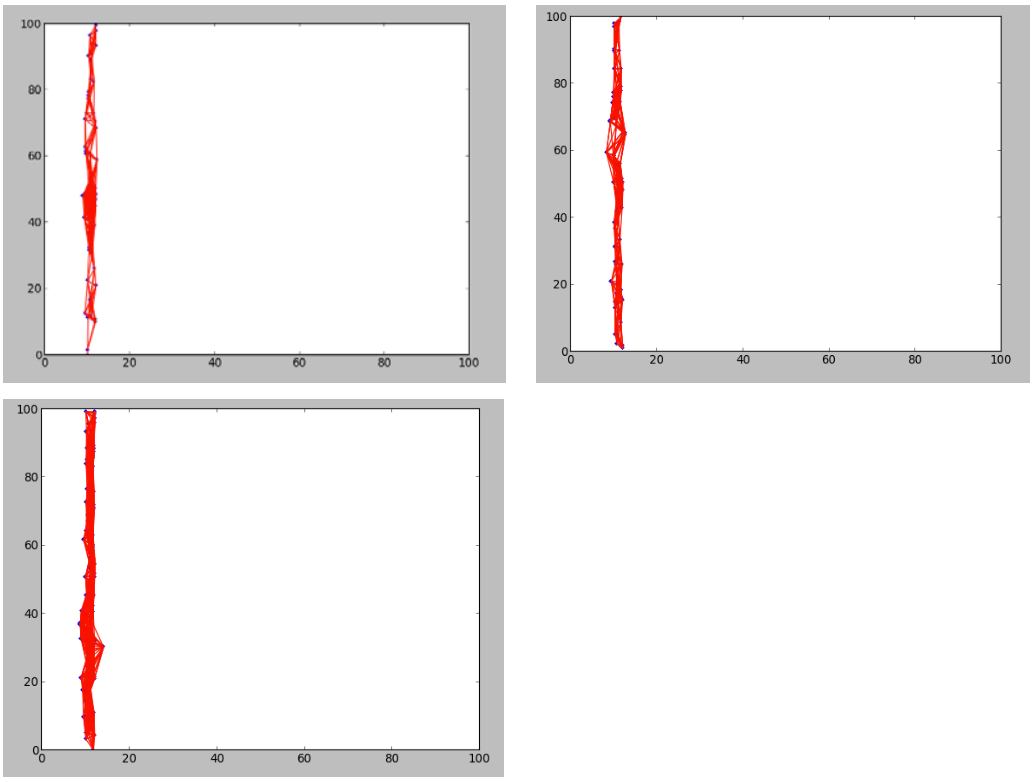

Figure 4.18: Pedestrian Communication Graph, for a Unit-Disk Graph (UDC) of 10m range, sidewalk width of 2m, and pedestrian arrival rate of 2, 3, 5 ped/s respectively.

5 Summary

This deliverable summarizes the work performed in Tasks 5.2 “Traffic Lights for Pedestrians and Bicycles” of the “COLOMBO” project. The objective was to extend the simulation suite by models of pedestrian and bicycle traffic, mainly focussing on their behaviour at intersections.

The major part of the work was performed on the traffic simulation SUMO. The implementation of pedestrian dynamics required several changes on the road network format but was performed in a way that assures backwards compatibility. The implementation hides the dynamics model(s) itself behind an application programmer’s interface (API), allowing to decide which model shall be used per simulation run. Two such models were implemented, a very simple linear movement model that does not regard collisions and a second model that divides the available sidewalk space into stripes.

In accordance with the changes network format, other tools from SUMO’s simulation suite were extended for supporting the user in preparing of simulation scenarios. The major part of this work affects the network preparation module NETCONVERT which reads different network formats and translates them into road networks SUMO (and other tools from the simulation package) can read. Besides reading available information about pedestrian sidewalks, this tool was also extended by heuristics that help in determining pedestrian infrastructure in case this information is missing. Its traffic light generation has been reworked to include signalling for pedestrians.

While performing the named extensions, SUMO’s capability to simulate intermodal trip chains was taken into account by implementing pedestrian dynamics as an extension of SUMO’s “walking” mode of travel. Given this, SUMO became a really microscopic, inter-modal traffic simulation, allowing a large variety of future use cases and investigations. All extensions performed on SUMO were made available as open source in accordance with the simulation’s licence (GPL).

Besides the work on SUMO, the work performed in Task 5.2 resulted in a standalone pedestrian simulation application that realises an implementation of the model described in [Antonini, Berlaire, Schneider, Robin, 2009]. As this application was written using the c++ programming language, the code is assumed to be included in SUMO in the near future. In addition, further applications of the tool itself will be elaborated.

Overall, the work yield in an extension of SUMO that is of a high value for COLOMBO itself and for further usage of SUMO. Making it available as open source with the next SUMO release will also open its usage for foreign parties, making the traffic research community benefit from the work as well.

Appendix A References

[Antonini et al., 2006] Antonini, Gianluca, Michel Bierlaire and Mats Weber: Discrete choice models of pedestrian walking behavior, Transportation Research Part B: Methodological 40.8: 667-687, 2006.

[Antonini, Berlaire, Schneider, Robin, 2009] Antonini, Berlaire, Schneider, Robin: Specification, estimation and validation of a pedestrian walking behavior model, In: Elsevier Transportation Research Part B: *Methodological*, Volume 43, Issue 1, Pages 36–56, January 2009.

[Bak, Kiec, 2012] Bak, R. and Kiec M.: Influence of various types of midblock pedestrian crossings on urban street capacity, Transportation Research Record 2316, Transportation Research Board, National Academy Press, Washington D.C., 2012.

[Ben-Akiva, Bierlaire, 1999] Moshe Ben-Akiva and Michel Bierlaire: Discrete Choice methods and their applications to short term travel decisions, in *Handbook of Transportation Science*, Volume 23, 1999, pp 5-33 1999.

[Bensator, Härri, Spyropoulos, 2014] Saleh Bensator, Jérôme Härri, Thrasyvoulous Spyropoulos: Modelization and Implementation of Urban Pedestrian Mobility, *EURECOM Technical Report RR-14-293*, April 2014.

[Brackstone et al., 2014] Brackstone, M.; Punzo, V. (Eds.): Traffic Simulation: Case for Guidelines, JRC Science and Policy Report, EUR 26534 EN. European Commission, Luxembourg. 2014. ISBN 978-92-79-35578-3. 4. http://publications.jrc.ec.europa.eu/repository/handle/111111111/30680.

[Burstedde et al., 2001] Burstedde, C., K. Klauck, A. Schadschneider, and J. Zittartz: Simulation of Pedestrian Dynamics using a two-dimensional Cellular Automaton. Physica A: Statistical Mechanics and its Applications, Vol. 295, No. 3 - 4, 2001, pp. 507 – 525.

[Curtis and Manocha, 2012] Curtis, Sean, and Dinesh Manocha: Pedestrian simulation using geometric reasoning in velocity space, Pedestrian and Evacuation Dynamics, 2012.

[Daamen, 2004] Daamen, W.: Modelling passenger flows in public transport facilities, Ph.D. thesis, Delft University Press, 2004.

[Dhamaniya, 2014] Dhamaniya, Chandra: Influence of Undesignated Pedestrian Crossings on Midblock Capacity of Urban Roads, TRB 2014

[Dias et al., 2014] Dias, Charitha, Omid Ejtemai, Majid Sarvi, and Nirajan Shiwakoti: An experimental study of pedestrians walking through angled corridors, Transportation Research Board 2014 Annual Meeting, Washington D.C.

[Gaca, Hogendorf, 2007] Gaca D., Hogendorf A.: General rules of operations and comparison of pedestrian crossings „zebra" type in Poland and In Germany. Part II. Transport Miejski i Regionalny, no. 2, 3, 2007 (in Polish).

[Gorni et al., 2014] Gorrini, A., Shimura S., Bandini, S., Ohtsuka, K. and Nishinari, K.: An Experimental Investigation of Pedestrian Personal Space: Towards Modeling and Simulations of Pedestrian Crowd Dynamics, Transportation Research Board 2014 Annual Meeting, Washington D.C.

[Helbing and Molnar, 1995] Helbing, D. and Molnar, P.: Social force model for pedestrian dynamics. Physical Review E 51 (5): 4282–4286, 1995.

[Henderson, 1971] Henderson, L.F.: The Statistics of Crowd Fluids, Nature, 229 (1971), p. 381, Nature Publishing Group. http://www.nature.com/nature/journal/v229/n5284/pdf/229381a0.pdf

[Lachapelle and Wolfram, 2006] Aimé Lachapelle, Marie-Therese Wolfram: On a mean field game approach modeling congestion and aversion in pedestrian crowds, 2006.

[Lasry and Lions, 2006] Lasry, Jean-Michel, and Pierre-Louis Lions: Mean field games, Japanese Journal of Mathematics 2.1: 229-260, 2007.

[Marisamynathan and Vedagiri, 2014] Marisamynathan, S. and P. Vedagiri: Modeling Pedestrian Delay at Signalized Intersections under Mixed Traffic Condition, Transportation Research Board 2014 Annual Meeting, Washington D.C.

[Mohcine, 2012] Mohcine, Chraibi: Validated force-based modeling of pedestrian dynamics, Ph.D. thesis, Schriften des Forschungszentrums Jülich, IAS Series Volume 13, 2012.

[Pajevic and Karlsson, 2013] Pajevic, Ljubica and Gunnar Karlsson: An Analytic Queuing Model for Urban Pedestrian Mobility, Technical Report, KTH Royal Institute of Technology, 2013.

[Saberi and Mahmassani, 2014] Saberi, Meead, and Hani S. Mahmassani: Exploring Area-Wide Dynamics of Pedestrian Crowds Using a Three-Dimensional Approach, Transportation Research Board 2014 Annual Meeting, Washington D.C.

[Teknomo et al., 2000] Teknomo, Kardi, Yasushi Takeyama and Hajime Inamura: Review on Microscopic Pedestrian Simulation Model, Proceedings Japan Society of Civil Engineering Conference, Morioka, Japan, 2000.

[Twaddle et al., 2014] Twaddle, H., Schendzielorz, Tobias, and Fakler, Oliver: Bicycles in Urban Areas: Review of Existing Methods for Modeling Behavior, Transportation Research Board 2014 Annual Meeting, Washington D.C.

[Werberich et al., 2014] Werberich, Pretto and Cybis: Pedestrians' Route Choice Based on Friction Forces Assuming Partial and Full Environment Knowledge, Transportation Research Board 2014

Appendix B User Documentation for Pedestrian Extensions

The creation of a simulation scenario includes the steps of network generation, demand generation and execution of the simulation. In Appendix C and Appendix D we describe these steps for the creation of a synthetic multimodal scenario and for a scenario based on OSM data.

This documentation assumes that the environment variable `SUMO_HOME` points to the root directory of the SUMO installation. We will use SUMO_HOME as shorthand for that directory. As a further requirement the directory `SUMO_HOME/bin` must be part of the application search path (specified in environment variable `PATH`) and the `python` interpreter in version 2.7 must be callable from the command line.

For running the simulation, the same set of options as in the previous version of SUMO is available.

Parameterization

Two types of parameters are available for customizing the simulation: global parameters (also called options) which are given on the command line or included in a *`.sumocfg`* file. All full list of these parameters along with their description can be obtained by calling

```
sumo --help
```

The newly added parameters are:

--pedestrian.model (The pedestrian model to use. Options are ‘striping’ and ‘nonInteracting’

--pedestrian.striping.stripe-width (Width of parallel stripes for segmenting a sidewalk in meters, only used with model ‘striping’)

--pedestrian.striping.dawdling (Factor for random slow-downs from [0,1], only used with model ‘striping’)

The second type of parameters is specific to a simulation entity or a group of entities. Pedestrians as well as bicycles can be customized using the following attributes:

- length: (the required physical space in the direction of movement)
- width: (the required physical space orthogonal to the direction of movement)
- minGap: (minimum distance to the leading entity of that mode)
- maxSpeed: (maximum speed)

An example on how to assign these attributes is given below. The values shown are the default values.

```
<routes>
    <vType id=”pedestrianType” length=”0.22” width=”0.48” minGap=”0.3” maxSpeed=”1.39”/>
    <person id=”p0” type=”pedestrianType” depart=”0”>
        <walk edges=”edge1 edge2”/>
    </person>

    <vType id=”bicycleType” length=”1.6” width=”0.65” minGap=”0.3” maxSpeed=”5.55”/>
    <vehicle id=”p0” type=”bicycleType” depart=”0”>
        <route edges=”edge1 edge2”/>
    </vehicle>
</routes>
```

It should be noted, that only the attribute only the attribute `maxSpeed` is considered when using pedestrian model `nonInteracting`.

Visualization

In the GUI, the drawing persons can be customized with in the *Persons*-tab of the view-settings dialog. Among other things, the drawing style can be changed between arrows, simple shapes and raster images[11]. The size of persons may be drawing with exaggerated size and the drawing of IDs can be toggled as well. Persons may be colored according to their speed, waiting time, walking direction or selection status. Also, colors specified in the person definitions may be used.

Simulation Outputs

To obtain simulation outputs any or all of the following options may be supplied:

```
sumo-gui –c example.sumocfg --tripinfo-output tripinfos.xml
    --fcd-output fcd.xml --netstate-dump netstate.xml
```

The contents of these files are documented at http://sumo-sim.org/wiki/Simulation/Output.

[11] http://sumo-sim.org/wiki/Specification/Persons#Visualization

Appendix C Creating an intermodal scenario with a single intersection

The following files are used to build the simulation scenario described in section 4.7.1.

For a complete syntax definition see

- http://sumo-sim.org/wiki/Networks/Building_Networks_from_own_XML-descriptions
- http://sumo-sim.org/wiki/Definition_of_Vehicles,_Vehicle_Types,_and_Routes

plain.nod.xml

```
<nodes>
    <node id="C" x="100.00" y="100.00" type="priority"/>
    <node id="E" x="200.00" y="100.00"/>
    <node id="N" x="100.00" y="200.00"/>
    <node id="S" x="100.00" y="0.00"/>
    <node id="W" x="0.00" y="100.00"/>
</nodes>
```

plain.edg.xml

```
<edges>
    <edge id="CE" from="C" to="E" priority="2" numLanes="4" speed="13.89">
        <lane index="0" allow="pedestrian" width="3.25"/>
        <lane index="1" disallow="all" width="1.50"/>
        <lane index="2" allow="passenger"/>
        <lane index="3" allow="passenger"/>
    </edge>
    <edge id="CN" from="C" to="N" priority="2" numLanes="4" speed="13.89">
        <lane index="0" allow="pedestrian" width="3.25"/>
        <lane index="1" disallow="all" width="1.50"/>
        <lane index="2" allow="passenger"/>
        <lane index="3" allow="passenger"/>
    </edge>
    <edge id="CS" from="C" to="S" priority="2" numLanes="4" speed="13.89">
        <lane index="0" allow="pedestrian" width="3.25"/>
        <lane index="1" disallow="all" width="1.50"/>
        <lane index="2" allow="passenger"/>
        <lane index="3" allow="passenger"/>
    </edge>
    <edge id="CW" from="C" to="W" priority="2" numLanes="4" speed="13.89">
        <lane index="0" allow="pedestrian" width="3.25"/>
        <lane index="1" disallow="all" width="1.50"/>
        <lane index="2" allow="passenger"/>
        <lane index="3" allow="passenger"/>
    </edge>
    <edge id="EC" from="E" to="C" priority="2" numLanes="4" speed="13.89">
        <lane index="0" allow="pedestrian" width="3.25"/>
        <lane index="1" disallow="all" width="1.50"/>
        <lane index="2" allow="passenger"/>
        <lane index="3" allow="passenger"/>
    </edge>
    <edge id="NC" from="N" to="C" priority="2" numLanes="4" speed="13.89">
        <lane index="0" allow="pedestrian" width="3.25"/>
        <lane index="1" disallow="all" width="1.50"/>
        <lane index="2" allow="passenger"/>
        <lane index="3" allow="passenger"/>
    </edge>
    <edge id="SC" from="S" to="C" priority="2" numLanes="4" speed="13.89">
        <lane index="0" allow="pedestrian" width="3.25"/>
        <lane index="1" disallow="all" width="1.50"/>
        <lane index="2" allow="passenger"/>
        <lane index="3" allow="passenger"/>
    </edge>
    <edge id="WC" from="W" to="C" priority="2" numLanes="4" speed="13.89">
        <lane index="0" allow="pedestrian" width="3.25"/>
        <lane index="1" disallow="all" width="1.50"/>
        <lane index="2" allow="passenger"/>
        <lane index="3" allow="passenger"/>
    </edge>
</edges>
```

plain.con.xml

```
<connections>
    <crossing node="C" edges="NC CN" width="4.56"/>
    <crossing node="C" edges="SC CS" width="4.56"/>
    <crossing node="C" edges="EC CE" width="4.56"/>
    <crossing node="C" edges="WC CW" width="4.56"/>
</connections>
```

The input files defining the demand for the above simulation are (partially) given in the following

multimodal.rou.xml

```
<routes>
    <route id="right" edges="SC CE"/>
    <flow id="right" type="car" route="right" begin="0" end="7200" period="1"
        departSpeed="max" departLane="best"/>

    <person id="0_fwd" type="ped" depart="0" color="yellow">
        <walk edges="SC CN" departPos="0" arrivalPos="-1"/>
    </person>
    <person id="3_bwd" type="ped" depart="3" color="blue">
        <walk edges="CN SC" departPos="-1" arrivalPos="0"/>
    </person>
    <person id="4_bwd" type="ped" depart="4" color="blue">
        <walk edges="CN SC" departPos="-1" arrivalPos="0"/>
    </person>
    <person id="9_fwd" type="ped" depart="9" color="yellow">
        <walk edges="SC CN" departPos="0" arrivalPos="-1"/>
    </person>
    <person id="9_bwd" type="ped" depart="9" color="blue">
        <walk edges="CN SC" departPos="-1" arrivalPos="0"/>
    </person>
    <person id="12_fwd" type="ped" depart="12" color="yellow">
        <walk edges="SC CN" departPos="0" arrivalPos="-1"/>
    </person>
    <person id="13_fwd" type="ped" depart="13" color="yellow">
        <walk edges="SC CN" departPos="0" arrivalPos="-1"/>
    </person>
...
</routes>
```

Using the above files the network is built using the command

```
netconvert –n plain.nod.xml –e plain.edg.xml –x plain.con.xml –o net.net.xml –no-turnarounds
```

The (graphical) simulation is started using the following command

```
sumo-gui –n net.net.xml –r multimodal.rou.xml
```

Appendix D Creating an intermodal OSM scenario

As a first step we download the raw OSM XML-data from the website www.openstreetmap.org. Here we select the 'export' tab. This allows us to select a bounding box in the map frame and export the data for the corresponding area. We select export the area within WGS84 coordinates (13.4601, 52.4230, 13.5439, 52.4540) and receive a file called *map.osm* corresponding to a part of Berlin where the DLR Institute of Transportation Systems resides.

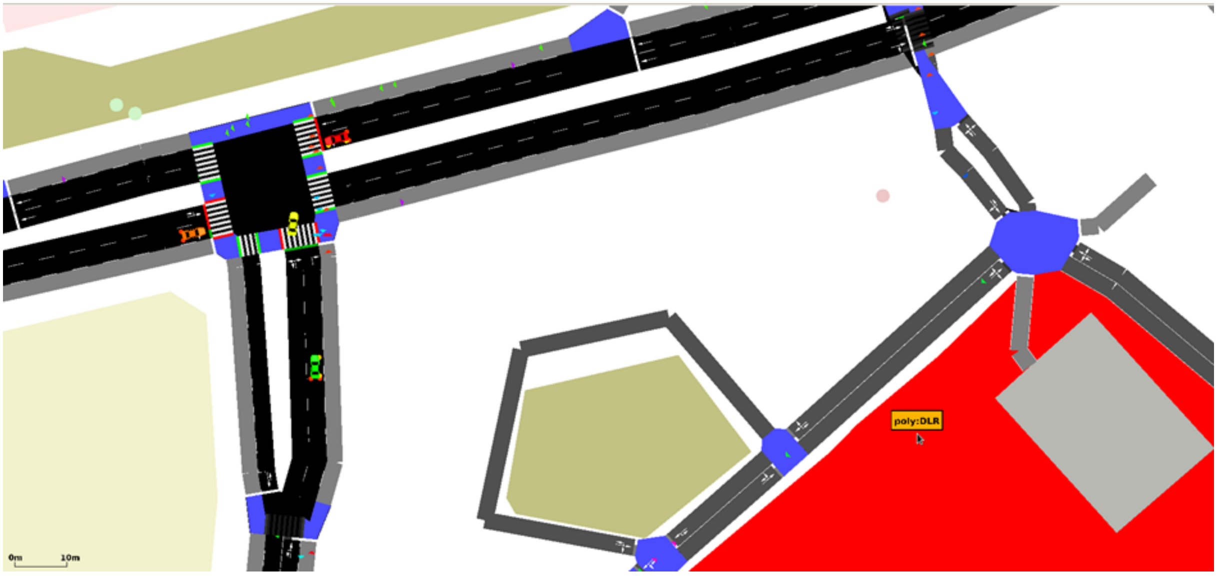

Figure D.19: Intermodal SUMO scenario based on network data from OpenStreetMap.org. The location of the *DLR Institute of Transportation Systems* is marked in red.

Next, we convert this data to a multimodal simulation network using the file *osm.typ.xml* defined in Appendix 1.1.4. The creation of pedestrian structures is triggered by sidewalks definitions within the file *osm.typ.xml* and by the option `--crossings.guess`.

```
netconvert --osm-file map.osm –o example.net.xml --type-files
     osm.typ.xml --junctions.join --roundabouts.guess --crossings.guess
```

In lieu of better data we use random demand for our scenario, generated by the tool *randomTrips.py*. First we prepare additional files that will be used for routing and simulation:

car_type.add.xml

```
<additional>
   <vType id="car" vClass="passenger" length="4.5" minGap="1.5" speedDev="0.2"/>
</additional>
```

And *ped_type.add.xml*

```
<additional>
   <vType id="ped" vClass="pedestrian" length="0.2" width="0.5" minGap="0.3" maxSpeed="1.3" />
</additional>
```

The following command generates vehicular demand. Note the use of the option `--fringe-factor` to let most vehicles start at the fringe of the small network.

```
python SUMO_HOME\tools\trip\randomTrips.py –n example.net.xml –r cars.rou.xml
    -t "type=\"car\" departSpeed=\"max\" departLane=\"best\""
    -c passenger --fringe-factor 10 --additional-files car_type.add.xml –-min-dist 400
    --fringe-threshold 10 –period 1.4
```

The following command generates pedestrians:

```
python SUMO_HOME\tools\trip\randomTrips.py –n example.net.xml –r peds.rou.xml
    –a type=\”ped\” –c pedestrian –-pedestrians –-max-dist 800 –period 0.8
```

The following command runs the (graphical) simulation. A screenshot of the simulation is shown in Figure 1. Note, that only the pedestrian types need to loaded explicitly because the vehicle type is already embedded in the file cars.rou.xml during demand generation

```
sumo-gui –n example.net.xml –r cars.rou.xml,peds.rou.xml –a ped_type.add.xml
```

Appendix E Type file for generating sidewalks for OSM input

The following file is for use with the NETCONVERT option `--type-files` and serves to add sidewalks to some edge types.

osm.typ.xml

```
<types>
      <type id="highway.motorway" numLanes="3" speed="44.44" priority="13" oneway="true" disallow="pedestrian bicycle"/>
      <type  id="highway.motorway_link"    numLanes="1"  speed="22.22"  priority="12"  oneway="true"  disallow="pedestrian bicycle"/>
      <type id="highway.trunk"          numLanes="2" speed="27.78" priority="11" disallow="pedestrian bicycle"/>
      <type id="highway.trunk_link"     numLanes="1" speed="22.22" priority="10" disallow="pedestrian bicycle"/>
      <type id="highway.primary"        numLanes="2" speed="27.78" priority="9" sidewalkWidth="3" disallow="pedestrian"/>
      <type id="highway.primary_link"   numLanes="1" speed="22.22" priority="8" sidewalkWidth="3" disallow="pedestrian"/>
      <type id="highway.secondary"      numLanes="2" speed="27.78" priority="7" sidewalkWidth="3" disallow="pedestrian"/>
      <type id="highway.secondary_link" numLanes="1" speed="22.22" priority="6" sidewalkWidth="3" disallow="pedestrian"/>
      <type id="highway.tertiary"       numLanes="1" speed="22.22" priority="6" sidewalkWidth="3" disallow="pedestrian"/>
      <type id="highway.tertiary_link"  numLanes="1" speed="22.22" priority="5″ sidewalkWidth="2" disallow="pedestrian"/>
      <type id="highway.unclassified"   numLanes="1" speed="13.89" priority="5" sidewalkWidth="2" disallow="pedestrian"/>
      <type id="highway.residential"    numLanes="1" speed="13.89" priority="4" sidewalkWidth="2" disallow="pedestrian"/>
      <type id="highway.living_street"  numLanes="1" speed="2.78"  priority="3"/>
      <type id="highway.service"        numLanes="1" speed="5.56"  priority="2" allow="delivery pedestrian"/>
      <type id="highway.track"          numLanes="1" speed="5.56"  priority="1"/>
      <type id="highway.services"       numLanes="1" speed="8.33"  priority="1"/>
      <type id="highway.unsurfaced"     numLanes="1" speed="8.33"  priority="1"/>

      <type id="highway.footway"        numLanes="1" speed="8.33"  priority="1"  oneway="true" allow="pedestrian"/>
      <type id="highway.pedestrian"     numLanes="1" speed="8.33"  priority="1"  oneway="true" allow="pedestrian"/>
      <type id="highway.path"           numLanes="1" speed="2.78"  priority="1"  oneway="true" allow="pedestrian"/>
      <type id="highway.bridleway"      numLanes="1" speed="2.78"  priority="1"  oneway="true" allow="pedestrian"/>
      <type id="highway.cycleway"       numLanes="1" speed="5.56"  priority="1"  oneway="true" allow="bicycle"/>
      <type id="highway.step"           numLanes="1" speed="1.39"  priority="1"  oneway="true" allow="pedestrian"/>
      <type id="highway.steps"          numLanes="1" speed="1.39"  priority="1"  oneway="true" allow="pedestrian"/>
      <type id="highway.stairs"         numLanes="1" speed="1.39"  priority="1"  oneway="true" allow="pedestrian"/>

      <type id="highway.bus_guideway"   numLanes="1" speed="8.33"  priority="1"  oneway="true" allow="bus"/>
      <type id="highway.raceway"        numLanes="2" speed="83.33" priority="14" allow="vip"/>
      <type id="highway.ford"           numLanes="1" speed="2.78"  priority="1" allow=" army"/>

      <type id="railway.rail"           numLanes="1" speed="83.33" priority="15" oneway="true" allow="rail rail_electric"/>
      <type id="railway.tram"           numLanes="1" speed="27.78" priority="15" oneway="true" allow="tram"/>
      <type id="railway.light_rail"     numLanes="1" speed="27.78" priority="15" oneway="true" allow="rail_light"/>
      <type id="railway.subway"         numLanes="1" speed="27.78" priority="15" oneway="true" allow="rail_light"/>
      <type id="railway.preserved"      numLanes="1" speed="27.78" priority="15" oneway="true" allow="rail"/>
</types>
```

Appendix F Tests for Intermodal Simulation Extensions

In the following, the list of test names is given grouped by application.

SUMO

```
pedestrian_model/striping/3_arm_pathing
pedestrian_model/striping/pedestrian_interactions/bidirectional_flow_4stripes
pedestrian_model/striping/pedestrian_interactions/bidirectional_flow_2stripes
pedestrian_model/striping/pedestrian_interactions/bidirectional_flow_3stripes
pedestrian_model/striping/pedestrian_interactions/bidirectional_flow_5stripes
pedestrian_model/striping/pedestrian_interactions/jam_trafficlight
pedestrian_model/striping/use_crossing/straight_ll
pedestrian_model/striping/use_crossing/turn_ll
pedestrian_model/striping/use_crossing/right_rr
pedestrian_model/striping/use_crossing/left_lr
pedestrian_model/striping/use_crossing/right_rl
pedestrian_model/striping/use_crossing/cross_twice/straight_rl
pedestrian_model/striping/use_crossing/cross_twice/straight_lr
pedestrian_model/striping/use_crossing/cross_twice/left_rr
pedestrian_model/striping/use_crossing/cross_twice/right_ll
pedestrian_model/striping/use_crossing/left_ll
pedestrian_model/striping/use_crossing/right_lr
pedestrian_model/striping/use_crossing/straight_rr
pedestrian_model/striping/use_crossing/turn_rr
pedestrian_model/striping/use_crossing/left_rl
pedestrian_model/striping/block_vehicles/walking_forward
pedestrian_model/striping/block_vehicles/walking_forward_left_turning_vehicle
pedestrian_model/striping/block_vehicles/walking_backward
pedestrian_model/striping/block_vehicles
pedestrian_model/striping/block_vehicles/walking_backward_left_turning_vehicle
pedestrian_model/nonInteracting/3_arm_pathing
pedestrian_model/nonInteracting/pedestrian_interactions/bidirectional_flow_4stripes
pedestrian_model/nonInteracting/pedestrian_interactions/bidirectional_flow_2stripes
pedestrian_model/nonInteracting/pedestrian_interactions/bidirectional_flow_3stripes
pedestrian_model/nonInteracting/pedestrian_interactions/bidirectional_flow_5stripes
pedestrian_model/nonInteracting/pedestrian_interactions/jam_trafficlight
pedestrian_model/nonInteracting/use_crossing/straight_ll
pedestrian_model/nonInteracting/use_crossing/turn_ll
pedestrian_model/nonInteracting/use_crossing/right_rr
pedestrian_model/nonInteracting/use_crossing/left_lr
pedestrian_model/nonInteracting/use_crossing/right_rl
pedestrian_model/nonInteracting/use_crossing/cross_twice/straight_rl
pedestrian_model/nonInteracting/use_crossing/cross_twice/straight_lr
pedestrian_model/nonInteracting/use_crossing/cross_twice/left_rr
pedestrian_model/nonInteracting/use_crossing/cross_twice/right_ll
pedestrian_model/nonInteracting/use_crossing/left_ll
pedestrian_model/nonInteracting/use_crossing/right_lr
pedestrian_model/nonInteracting/use_crossing/straight_rr
pedestrian_model/nonInteracting/use_crossing/turn_rr
pedestrian_model/nonInteracting/use_crossing/left_rl
pedestrian_model/nonInteracting/block_vehicles/walking_forward
pedestrian_model/nonInteracting/block_vehicles/walking_forward_left_turning_vehicle
pedestrian_model/nonInteracting/block_vehicles/walking_backward
pedestrian_model/nonInteracting/block_vehicles
pedestrian_model/nonInteracting/block_vehicles/walking_backward_left_turning_vehicle
output/vehroutes/with_person
```

NETCONVERT

```
import/OSM/adlershof_dlr_sidewalks
function/sidewalks/basic
function/sidewalks/patch_tls_from_sumonet
function/crossings/guessed/tl_controlled_path
function/crossings/guessed/3_arm_1_opposite_sidewalk_3_crossings
function/crossings/guessed/motorway_split
function/crossings/guessed/2_arm_path
function/crossings/guessed/4_arm_complete_priority
function/crossings/guessed/4_arm_parallel_path
```

```
function/crossings/guessed/path_across_street
function/crossings/guessed/adlershof_dlr
function/crossings/guessed/path_across_1way_street
function/crossings/guessed/4_arm_slanted_complete
function/crossings/guessed/2_arm_oneway_WE
function/crossings/guessed/3_arm_path
function/crossings/guessed/4_arm_complete
function/crossings/guessed/2_arm_oneway_EW
function/crossings/guessed/4_arm_3_neightboring_split
function/crossings/guessed/2_arm
function/crossings/guessed/patch_tls_from_sumonet
function/crossings/specified/3_arm_1_opposite_sidewalk_3_crossings
function/crossings/specified/4_arm_3_neighboring_split
function/crossings/specified/3_arm_1_opposite_sidewalk_1_crossing
function/crossings/specified/4_arm_1
function/crossings/specified/4_arm_2_neighboring
function/crossings/specified/2_arm_oneway_invalid
function/crossings/specified/path_across_street
function/crossings/specified/4_arm_slanted_complete
function/crossings/specified/4_arm_complete
function/crossings/specified/4_arm_2_opposite
function/crossings/specified/3_arm_service_entering
function/rightTurnConflict
```

DUAROUTER

```
routes/style2_input/with_persons_only
person/unconnected_ignored
person/crossings_and_walkingareas
person/basic
person/unconnected
```

Appendix G Alternative Architectures

In the design phase of the extensions described in this document, a number of alternative approaches to achieve the aims of modeling interactions between pedestrians and road vehicles were evaluated. They are listed below along with reasons for their rejection.

Pedestrians as special vehicles with special lanes

Pedestrians are modelled as special vehicles. Sidewalks are modelled by adding multiple parallel lanes to existing edges. Crossings are part of existing intersections. *Walkingareas* are not modelled. An experimental scenario is shown in Figure 4.2.

The following problems were encountered:

- inflexible in regard to adding other pedestrian models,
- no bidirectional movements,
- difficulty in modelling the exact location in which pedestrian crossings start and end,
- normal connections always pass across the intersection whereas pedestrians walk along the outside and cross at the narrowest point,
- large number of lanes and connections needed (very large junction logic).

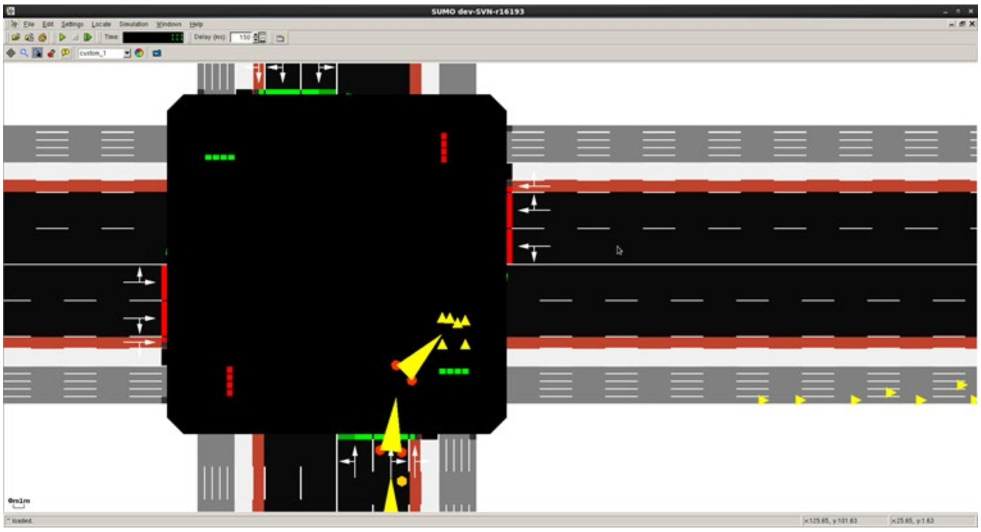

Figure 4.20: Alternative network model with multiple lanes for vehicle-like pedestrians

Pedestrians as special vehicles with special edges

Pedestrians are modelled as special vehicles. Sidewalks are modeled by adding two extra edges (one for each direction) parallel to existing edges. Each crossing is modeled as a new intersection between sidewalk edges and road edges. Each *walkingarea* is also modeled as a new intersection. An experimental scenario is shown in Figure 4.3.

The following problems were encountered:

- large number of extra edges and junctions needed, 1 4-arm junction with 8 (unidirectional) edges additionally needs into 8 junctions (4 crossings and 4 *walkingareas*) with 48 (unidirectional) pedestrian edges (16 running parallel to the original ones as sidewalks, 16 to connect to connect sidewalks with *walkingareas*, 16 to connect *walkingareas* with crossings
- can only use part of a sidewalk for one direction (edges still unidirectional),

- difficulty of coordinating TLS-logic of original junction with the 4 new junctions where pedestrians cross the roads,
- inflexible in regard to adding other pedestrian models.

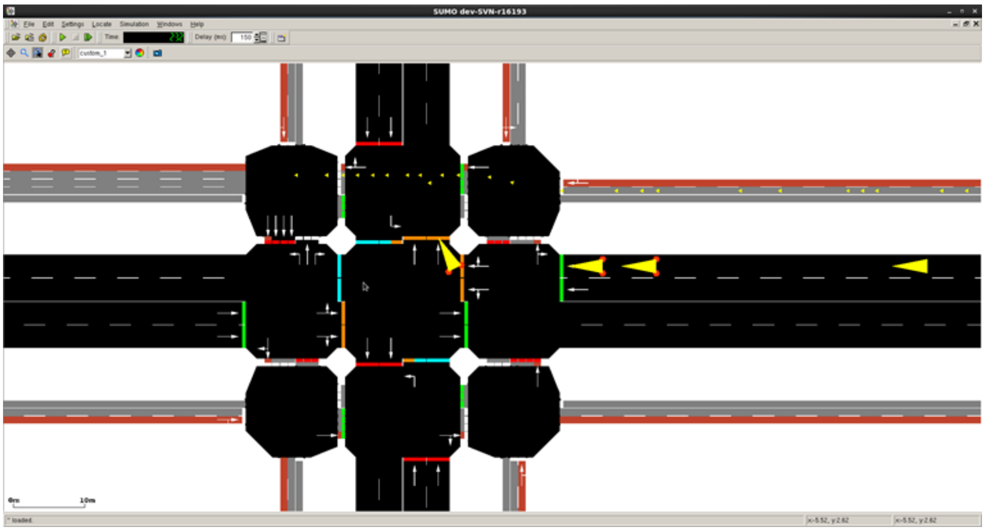

Figure 4.21 : Alternative network model with additional edges for vehicle-like pedestrians

Appendix H Current Online User Documentation

Contents

Pedestrian Simulation

This page describes simulations of pedestrians in SUMO. To build an intermodal simulation scenario with propper interactions between road vehicles and pedestrians, additional steps have to be take in comparisson to a plain vehicular simulation.

Building a network for pedestrian simulation

When walking along an edge, pedestrians use sidewalks where available. A sidewalks is a lane which allows only the sumo vClass *pedestrian*. When crossing a road at an intersection, pedestrians use special lanes of type *crossing*. The area that connects sidewalks with crossings is modeled by special lanes of the type *walkingarea*. In the following, we describe how to build a simulation network that *contains sidewalks*, *crossings* and *walkingareas*.

Generating a network with sidewalks

Sidewalks may be defined explicitly in plain XML input when describing edges (plain.edg.xml). This is done by defining an additional lane which only permits the vClass “pedestrian” and setting the appropriate width.

When importing edges with defined types, it is also possible to declare that certain types should receive a sidewalk. This can be used to automatically generate sidewalks for residential streets while omitting them for motorways when importing OSM data. An examplary typefile can be found in <***SUMO_HOME***>/data/typemap/osmNetconverPedestrians.typ.xml.

```
<types>
   <type id="highway.motorway" numLanes="3" speed="44.44" priority="13"
oneway="true" disallow="pedestrian bicycle"/>
   <type id="highway.unclassified"   numLanes="1" speed="13.89"
priority="5" sidewalkWidth="2" disallow="pedestrian"/>
   <type id="highway.residential"    numLanes="1" speed="13.89"
priority="4" sidewalkWidth="2" disallow="pedestrian"/>
   <type id="highway.living_street"  numLanes="1" speed="2.78"
priority="3"/>
```

```
   <type id="highway.service"        numLanes="1" speed="5.56"
priority="2" allow="delivery pedestrian"/>
   ...
</types>
```

A third option which can be used if no edge types are available is a heuristic based on edge speed. This is activated by using the following options:

Option	Description
--sidewalks.guess ***<BOOL>***	Guess pedestrian sidewalks based on edge speed
--sidewalks.guess.max-speed ***<FLOAT>***	Add sidewalks for edges with a speed equal or below the given limit
--sidewalks.guess.mim-speed ***<FLOAT>***	Add sidewalks for edges with a speed above the given limit

Generating a network with crossings

Crossings may be defined explicitly in plain XML input when describing connections (plain.con.xml) using the new XML element crossings.

The second available method adding crossing information to a network is with the option **--crossings.guess** ***<BOOL>***. This enables a heuristic which adds crossings wherever sidewalks with similar angle are separated by lanes which forbid pedestrians. If the edges to be crossed have sufficient distance between them or vary a by a sufficient angle, two crossings with an intermediate walking area are generated. To use this option sidewalks should be defined for the network.

Generating pedestrian demand

Pedestrian demand may be specified explicitly as described at Specification/Persons#Walks or it may be generated. The tool Tools/Trip#randomTrips.py supports generating random pedestrian demand using the option **--pedestrians**. The option **--max-dist** ***<FLOAT>*** may be used to limit the maximum air distance of pedestrian walks.

Pedestrian Models

The pedestrian model to use can be selected by using the new simulation option **--pedestrian.model** ***<STRING>*** with the available paramters being *nonInteracting* and *striping* (default is *striping*). The interface between the pedestrian model and the rest of the simulation was designed with the aim of having a high degree of freedom when implementing new models. It is planned to implement models with a higher level of interaction detail in the future.

Model *nonInteraction*

This is a very basic walking model. Pedestrians walk bidirectionally along normal edges and "jump" across intersections. They maybe either be configured to complete a walk in a fixed amount of time or to move along the edges with a fixed speed. No interaction between pedestrians and vehicles or other pedestrians takes place. This model has a very high execution speed and may be useful if the pedestrian dynamics are not important.

Model *striping*

This model assigns 2D-coordinates within a lane (of type sidewalk, walkingarea or crossing) to each pedestrian. These coordinates which are defined relative to the leftmost side of the start of the lane are updated in every simulation step. This is in contrast to the coordinates of vehicles, which (generally) only have 1D-coordinates within their respective lane. Pedestrians advance along a lane towards the next node which may either correspond to the natural direction of the lane (forward movement) or it may opposite to the natural direction (backward movement). Thus, the x coordinate monotonically increase or decreases while on a lane. Once the end of a lane has been reached, the pedestrian is placed on the next lane (which may either be unique or determined dynamically with a routing algorithm).

The most important feature of pedestrian interactions is collision avoidance. To achieve this, the "striping"-model divides the lateral width of a lane into discrete stripes of fixed width. This width is user configurable using the option **--pedestrian.striping.stripe-width** ***<FLOAT>*** and defaults to 0.65 m. These stripes are similar to lanes of a multi-lane road are used by vehicles. Collision avoidance is thus reduced to maintaining sufficient distance within the same lane. Whenever a pedestrian comes too close to another pedestrian within the same stripe it moves in the y-direction (laterally) as well as in the x-direction to change to a different stripe. The y-coordinate changes continuously which leads to situations in which a pedestrian temporarily occupies two stripes and thus needs to ensure sufficient distances in both. The algorithm for selecting the preferred stripe is based on the direction of movement (preferring evasion to the right for oncoming pedestrians) and the expected distance the pedestrian will be able to walk in that stripe without a collision. During every simulation step, each pedestrian advances as fast as possible while still avoiding collisions. The updates happen in a single pass for each walking direction with the pedestrian in the front being updated first and then its followers sorted by their x-coordinate. The speed in the x-direction may be reduced by a random amount with the maximum amount defined as a fraction of the maximum speed, using the global option **--pedestrian.striping.dawdling** ***<FLOAT>*** (defaulting to 0.2). As a consequence of the above movement rules, pedestrians tend to walk side by side on sidewalks of sufficient width. They wait in front of crossings in a wide queue and they form a jam if the inflow into a lane is larger than its outflow.

Interaction between pedestrians and other modes

A pedestrian wishing to cross the street at an uncontrolled intersection can only do so if its expected time slot for using the intersection does not interfere with that of an approaching vehicle. It should be noted that the dynamics at unprioritized crossings are conservative in estimating the time required gap. In the simulation, pedestrians will only use such a crossing if the whole length of the crossing is free of vehicles for the whole time needed to cross. At priority crossings, pedestrians cross without regard for vehicles.

Vehicles are prevented from driving across a pedestrian crossing which is occupied by pedestrians. If a pedestrian is found which is not yet past the intersection point (between the crossing and the vehicles trajectory) but within a threshold distance to that point (currently hardcoded as 10m) the crossing is considered to be blocked.

Note:
Using the model 'nonInteracting', no interactions between pedestians and other modes take place.