Modelling Pedestrian Dynamics in SUMO

2 authors:


Jakob Erdmann
German Aerospace Center (DLR)
53 PUBLICATIONS **1,355** CITATIONS


Daniel Krajzewicz
German Aerospace Center (DLR)
131 PUBLICATIONS **2,371** CITATIONS

Modelling Pedestrian Dynamics in SUMO

Jakob Erdmann and Daniel Krajzewicz
German Aerospace Center , Rutherfordstraße 2, 12489 Berlin, Germany

Jakob.Erdmann@DLR.de, Daniel.Krajzewicz@DLR.de

1.1 Abstract

Walking is the most natural way of human mobility. The microscopic simulation package SUMO has supported an intermodal person-based simulation including pedestrian since 2010 (version 0.12.0). However, movements along a road were resembled using a linear interpolation only and pedestrian dynamics at an intersection were not modelled at all. Within the scope of the COLOMBO project, SUMO was extended to simulate pedestrian dynamics in more greater detail. This includes extensions to the road network format, movement models and routing tools.

Keywords: pedestrian dynamics, human locomotion, inter-modality.

1.2 Introduction

Especially in Europe, a long-term shift towards putting soft modes of transport, mainly bicycles and pedestrians, into the focus can be observed. Several reasons motivate this. First, these modes of transport are environment friendly. Second, they are healthier than using vehicles, both for the user himself as well as for other persons. Both circumstances do not only motivate individuals to change to non-motorized traffic. They are as well targeted by authorities and societies that try to avoid penalties for not keeping pollutant density thresholds and avoid long-term costs of an unhealthy population. On the other hand, pedestrians and bicycles require special care due to being more vulnerable than motorized traffic.

Conventionally, traffic simulations are helping in the design and development of both, strategic actions as well as road infrastructure changes. Consequently, established commercial traffic simulations have incorporated pedestrian and/or bicycle dynamics in recent years. Until version 0.21.0 SUMO lacked a model of pedestrian dynamics. Albeit its inter-modal person routes [Behrisch et.al, 2010] include a “walking” stage, pedestrians were moved along roads with a constant speed and jumped over the intersections. No interactions between pedestrians or between pedestrians and traffic were modelled.

Extending SUMO by pedestrian and bicycle dynamics was scheduled within the COLOMBO project. The main goal of COLOMBO is to develop traffic surveillance method and traffic light controls that use data from V2X-enabled vehicles assuming a low equipment rate. In this context, the dynamics of pedestrians was assumed to be necessary for two reasons:

- Pedestrians may deliver additional information that may be used by the developed traffic surveillance and, in case of a sparse connectivity, may be used as additional relay nodes.
- Vehicles which turn left or right at an intersection typically have to yield to pedestrians. Thus pedestrians need to be included in a simulation to correctly model urban intersection capacity.

- The development of environment-friendly traffic light controls should take pedestrians into account to allow research on control strategies which prioritize the environment-friendly transport modes.

The remainder is structured as following. At first, the requirements for pedestrian dynamics are listed. Then, the implementation is described and finally some measurements from pedestrian simulations are given.

1.3 Requirements

As outlined, the major goal was to correctly replicate the behavior of pedestrians at (traffic-light controlled) intersections. This implicates the following functional requirements:

- 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.

The according adaptations had to be performed to the simulation modules (SUMO and SUMO-GUI). One should note that the implementation of the needed models and data structures has to be accompanied by according extensions in the used data files. As well, to match SUMO's philosophy of offering a high level of user support, the supporting tools had to be extended. Thereby, 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

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,

- pedestrian dynamics should be included in the existing inter-modal trip chains.

1.4 Implementation

This section describes the implementation steps performed for obtaining the needed functionality. At first, the changed in representation of the infrastructure are given followed by a presentation of the implementation of the pedestrian dynamics themselves. Finally, additional work performed on the network and routes preparation modules is described.

1.4.1 Infrastructure

One of the most challenging work steps was to find a representation of the road infrastructure used by pedestrians that on the one hand is capable of representing complex intersections, but on the other hand fits well with the existing structures and is efficient enough for large simulations. Different alternatives for modeling intersections were tested as reported in [COLOMBO D5.2, 2014].

In the implemented architecture, 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 where each lane allows 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 (and lanes) are added for modeling pedestrian *crossings* and for modeling sidewalk corners where crossings and sidewalks meet (called *walkingarea*). Figure 1 shows the previous and the extended network model in the simulation GUI.

a)

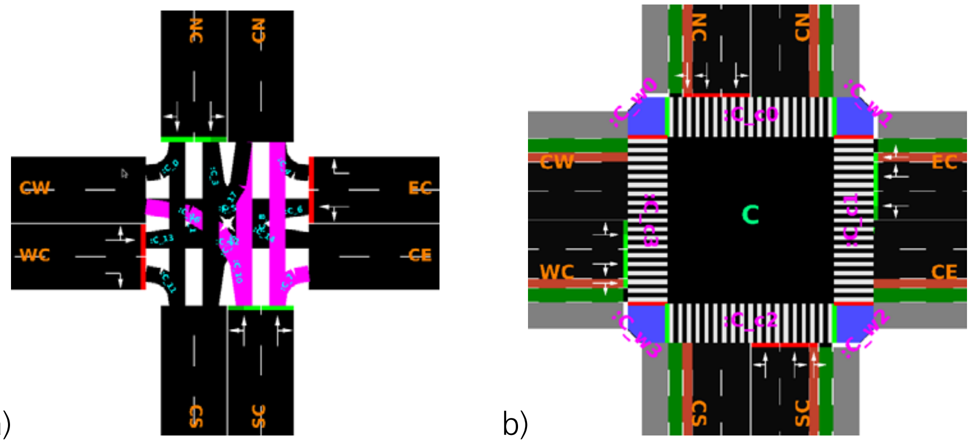

Figure 1 a) Previous network model with “normal” edges labelled in orange and “internal” edges labelled in cyan. The internal edges outgoing from edge “SC” are highlighted in magenta; b) 4-arm intersection with bicycle lanes (brown), sidewalks (grey), walkingareas (blue), crossings (striped), green verges (green). IDs are shown for all edges which may be used by pedestrians.

Each crossing is defined to either give priority to the pedestrian or to the vehicle (The latter case is distinguishable in the GUI by having black/grey stripes instead of black/white). The connectivity among sidewalks, walkingareas and crossings is modelled using unidirectional connections as in the previous network model. However, these connections as well as the edges may be traversed by pedestrians 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 (see 1.4.4). When drawing a *walkingarea* in the GUI its “shape”-attribute is interpreted as the polygonal border around the space, rather than as a polygonal line in the direction of the edge.

Some features of real world traffic such as heterogeneous lane use and bidirectional lane use during overtaking cannot by modelled by this architecture. This is a consequence of keeping the existing vehicular model and only extending the intersection model. However, we expect the extended intersection model to be able to accommodate future extensions along these lines.

1.4.2 Pedestrian Dynamics

One of the initial goals of the traffic simulation SUMO was to support researchers in comparing and validating different traffic models. While this was mainly stated having car-following models in mind, it should as well count for models of pedestrian dynamics. Therefore, not a single dedicated model of pedestrian dynamics was implemented, but rather an API (application programmer interface) for embedding different models. The interface is minimalistic to give the model developer a high degree of freedom. A pedestrian dynamics model has to 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, 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 older versions 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.

Currently, two pedestrian dynamics models are included in SUMO. They are presented in the following subsections.

1.4.3 Model “nonInteracting”

The initial “dynamics model” where pedestrians move with a constant speed, disregard interactions with other pedestrians and “jump” across intersections can be selected using the option `--pedestrian.model nonInteracting`. It may be useful if the pedestrian dynamics are not important and a high execution speed is desired. One enhancement that has been made is that pedestrians may use edges in both directions. The walking direction on a given edge is computed based on the topology of the edge sequence.

1.4.4 Model “striping”:

The “striping” model implements detailed pedestrian dynamics according to the requirements in section 1.3. It is selected using the option `--pedestrian.model striping` and also serves as the new default model. In the following, the main functionalities of the model are described.

Routing within an intersection

When passing an intersection, a sequence of walkingareas and crossings must be used to reach the next “normal” edge in the pedestrians route. When there are multiple routes available, the *PedestrianRouter* described in 1.4.7 is used with its scope limited to the current intersection. The signal states of the traffic lights are used to select a path which avoids waiting when possible.

Interactions of pedestrians with each other

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).

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 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.

The division into stripes in the lateral direction is straightforward for walking areas and crossings which have two main directions of walking. In contrast, walkingareas are used in multiple directions. To apply the above movement rules additional processing takes place. For every combination of sidewalk and crossing adjacent to a walkingarea, a unique path is computed at the start of the simulation. During the simulation each pedestrian uses the unique path which allows it to follow the sequence of walkingareas and crossings prescribed by the *PedestrianRouter*. Each of these paths is computed separately according to the above movement rules. To avoid collisions between pedestrians on different walkingarea-paths, the pedestrians from other paths are mapped into the coordinate system of the current path beforehand.

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 (i.e. pedestrians cannot back up in order to clear space in a crowded area). 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 (Since the coordinate-remapping on walkingareas only happens per path, the effort is linear in the number of pedestrians) .

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 the 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.

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 fixed at 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.

1.4.5 Further Pedestrian Models

Both presented models have not been published before and are thereby not known to the scientific community. Within the COLOMBO project, a further model was implemented that has been discussed in literature intensively [Antonini et al., 2006], [Antonini, Berlaire, Schneider, Robin, 2009]. Being currently implemented within a standalone application, the model has not yet been included into SUMO. This is planned to be done during the next time. As well, several open source implementations of established pedestrian dynamics models exist, e.g. "pedsim" [pedsim] which uses a social force model (see [Helbing, Molnár, 1995]), that are planned to be included.

1.4.6 Extensions for network generation

The NETCONVERT application is part of the SUMO application suite. It is responsible for preparing the simulation network (net.xml) from a wide range of input data formats such as OpenStreetMap (OSM), VISSIM or shape files. 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. 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 sections. 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. For a usage description of the new functionality refer to [3].

Generating Sidewalks

NETCONVERT supports multiple ways of defining sidewalks which are appropriate in different usage scenarios:

- 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 or by including the new attribute `sidewalkWidth`
- When importing edges with defined types (i.e. from *OpenStreetMap*), sidewalks may be added for selected types
- Heuristically for all edges with a speed limit within a defined range
- Based on permissions: edges which allow pedestrians receive a sidewalk.

Below is an example edge definition with two vehicular lanes, a green verge, a bicycle lane and a sidewalk:

```
<edge id="x" from="A" to="B numLanes="5" speed="13.89">
    <lane index="0" allow="pedestrian" width="3.00"/>
    <lane index="1" allow="bicycle" width="1.00"/>
    <lane index="2" disallow="all" width="1.50"/>
    <lane index="3" allow="passenger"/>
    <lane index="4" allow="passenger"/>
</edge>
```

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=”<nodeId>”` `edges=”<listOfEdgesToCross>”` and the optional attributes `width=”<widthInM>”` 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. Below is an example crossing defininition:

```
<crossing node="A" edges="x y" width="4.5"/>
```

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 2.

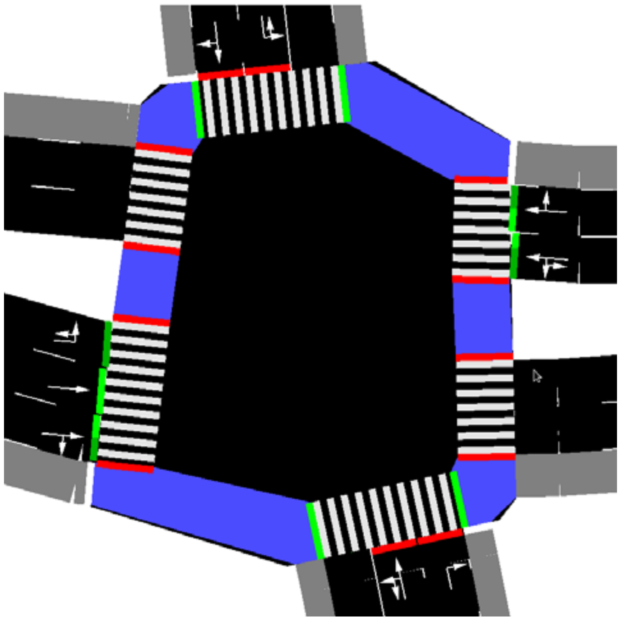

Figure 2 Intersections with split pedestrian crossings

Generating Right of Way Rules

The Intersection model described in [Erdmann, Krajzewicz, 2014] extends naturally to pedestrian crossings. Crossings are simply another set of *internal lanes* which must be considered in the conflict-matrix and the right-of-way 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. Additional phases are generated to allow pedestrians to leave an intersection before giving vehicles the green light.

NETCONVERT can apply these heuristics to existing signal plans in order to "upgrade" vehicular networks in to intermodal networks.

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. 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.

1.4.7 Additional Extensions

Various other additions to the SUMO code base were implemented in order to meet the requirements listed in section 1.3. They are described in the following.

Pedestrian router

A routing application was implemented which allows routing bidirectionally on sidewalks, walkingareas and crossings. This is accomplished for constructing a special routing graph which can be processed with the existing implementation of the Dijkstra routing algorithm. One particular feature where this adapted graph differs is the dynamic treatment of TLS-controlled crossings. In reality pedestrians which need to cross the street twice to reach the diagonally opposite 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.

Output

The existing methods for retrieving simulation data where extended to cover pedestrians

- Option `–fcd-output` now includes positions, speeds and angles of pedestrians
- Option `–nestate-output` now includes positions, speeds and angles of pedestrians
- TraCI allows retrieving 2D-position, edge, edge-position, angle and speed of persons (and thus pedestrians)

Demand Generation

The tool randomTrips.py was extended with the options `--pedestrians` which generates persons with a single walk between random locations. The new option `--max-distance` can be used to limit the distance of walks.

1.5 Simulations

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

1.5.8 Interactions between right-turning vehicles and crossing pedestrians at a single intersection

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 3. The simulation was conducted with a traffic light as well

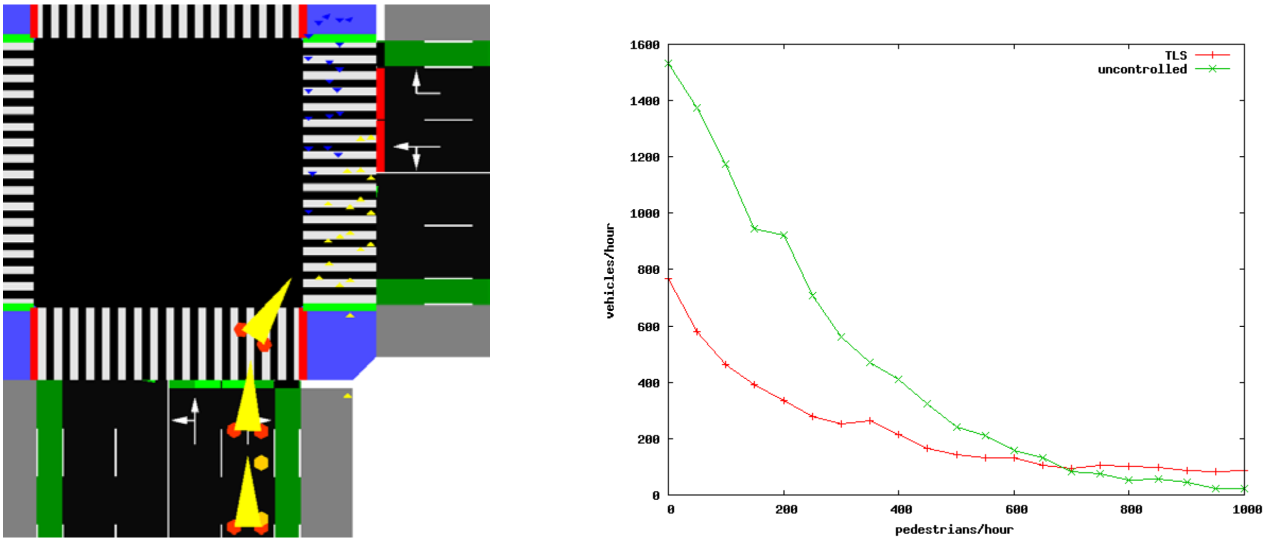

Figure 3 Simulation experiment for measuring the relationship between pedestrian flow and right turning vehicle flow (TLS-controlled intersection).

as with a prioritized pedestrian crossing. The traffic light was following a fixed cycle of 90 seconds with 26 seconds combined green time for vehicles and pedestrians and 5 seconds exclusive green for the vehicles (the rest of the cycle being reserved for other directions of traffic). Figure 4 shows the vehicular flow behind the crossing in dependence on pedestrian density. 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.

1.5.9 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* [Bieker et al., 2013] 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. To extend this scenario,

Figure 4 ACOSTA scenario with pedestrian enhancements. Pedestrians are shown at exaggerated size to increase visibility.

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. The green time allotted to vehicles remained unchanged. Pedestrian demand was generated randomly using the tool *randomTrips.py* described in section 1.4.7. 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 5.

To measure the influence of the pedestrians on vehicular traffic, we compared the duration of vehicular trips in both versions of the scenario. Figure 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. The increased travel times were seen to be caused by the conflict of turning traffic with crossing pedestrians.

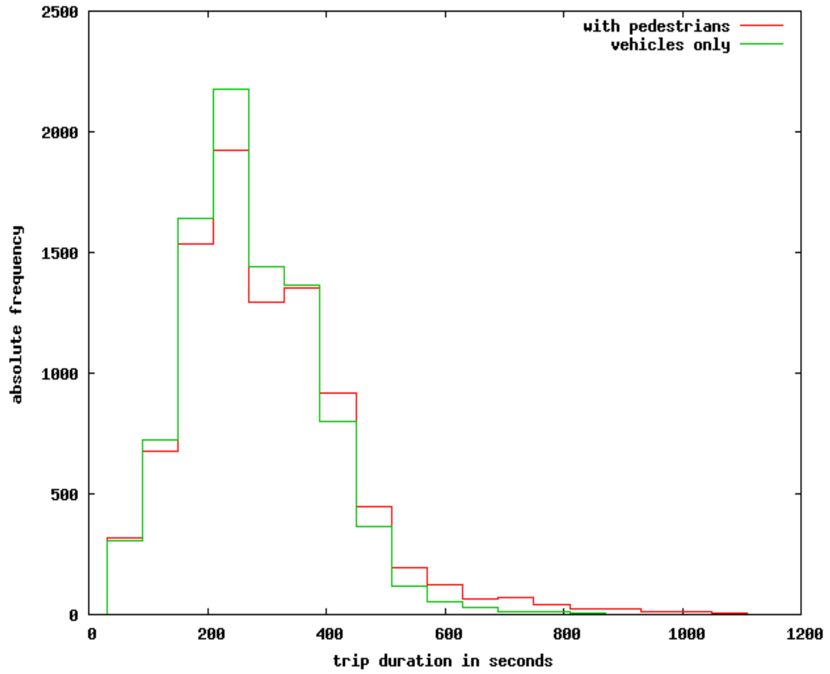

Figure 5 Histogram of vehicle trip durations in both versions of the ACOSTA scenario

We also compared the running times of both scenarios and noted that the average real-time factor (the fraction of simulated time over running time) decreased from 800 to 460 when adding pedestrians. When increasing the number of pedestrians to 9000 the real-time factor decreased further to 330. However, simulating 9000 pedestrians without vehicles still has a factor of 630. This shows that the pedestrian model has similar complexity to the vehicular model and a surprisingly large fraction of the time is spent on the interaction of vehicles and pedestrians.

1.6 Conclusion

An extension of SUMO for modelling pedestrians was presented. The work included modifications to several tools included in the SUMO package, which support the generation, simulation and analysis of multi-modal traffic scenarios.

The presented extensions allow a number of new investigations. While the major focus was put on evaluating the behavior of pedestrians at traffic lights, the implementation allows simulating pedestrians on a city-wide level. Being integrated into SUMO's inter-modal trip chains, it enhances SUMO by allowing microscopic modelling and evaluation of all (common) modes of urban transport.

The currently available models for both vehicle and pedestrian traffic are not yet fine-grained enough to address traffic safety research questions, but the inclusion of pedestrians into a traffic simulation with a modular architecture is assumed to be an important step towards that goal

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 when the focus is on vehicular traffic.

Acknowledgements

The authors want to thank the European Commission for co-funding the work in the context of the "COLOMBO" project under the grant number 318622.

References

[1] Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D.: SUMO – Simulation of Urban MObility: An Overview. In: SIMUL 2011, The Third International Conference on Advances in System Simulation (2011)
[2] DLR and contributors: SUMO homepage. http://sumo.dlr.de/ (2015)
[3] SUMO online documentation on pedestrians 2015 http://sumo.dlr.de/wiki/Simulation/Pedestrians
[4] Krajzewicz, D.: Traffic Simulation with SUMO – Simulation of Urban Mobility. In: Barceló, Jaume (Ed.): Fundamentals of Traffic Simulation, Series: International Series in Operations Research & Management Science, Vol. 145, Springer, ISBN 978-1-4419-6141-9 (2010)
[5] [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.
[6] [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.
[7] [pedsim] http://pedsim.silmaril.org/
[8] [Helbing, Molnár, 1995] Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. Physical Reviews E,
[9] [Behrisch et al., 2010] Behrisch, Michael and Erdmann, Jakob and Krajzewicz, Daniel (2010) Adding intermodality to the microscopic simulation package SUMO. MESM 2010, 01-03.Dez. 2010, Alexandria, Agypten.
[10] [COLOMBO report D5.2, 2014] Erdmann, Jakob and Niebel, Wolfgang and Krajzewicz, Daniel and Härri, Jérôme and Bensator, Saleh (2014) COLOMBO: Deliverable 5.2: Traffic Simulation Extensions; Public Deliverable of the COLOMBO project; http://www.colombo-fp7.eu

[11] [Erdmann, Krajzewicz, 2014] Erdmann, Jakob und Krajzewicz, Daniel (2014) *SUMO's Road Intersection Model.* In: Simulation of Urban Mobility Lecture Notes in Computer Science, 8594. Springer. Seiten 3-17. ISBN 978-3-662-45079-6

[12] [Bieker et al., 2014] Bieker, Laura und Krajzewicz, Daniel und Morra, Antonio Pio und Michelacci, Carlo und Cartolano, Fabio (2014) *Traffic simulation for all: a real world traffic scenario from the city of Bologna.* SUMO 2014, 15.-16. Mai 2014, Berlin, Deutschland