

Intermodal Contour Accessibility Measures Computation Using the ‘UrMo Accessibility Computer’

Daniel Krajzewicz, Dirk Heinrichs, Rita Cyganski

Institute of Transport Research

German Aerospace Center

Berlin, Germany

daniel.krajzewicz@dlr.de, dirk.heinrichs@dlr.de, rita.cyganski@dlr.de

Abstract—Contour accessibility measures form a set of performance indicators used to value a location by the amount of accessible activities, places, or space within certain time or distance limits. They are used for evaluating a region’s activity potential taking the connectivity to its surroundings into regard and act as input data for land use planning models and traffic demand models. With the availability of disaggregated data and sufficient computer power, accessibility measures can be computed at a very fine-grained level of single buildings, points of interest or areas. This approach considers single routes through the transportation network and allows for computing intermodal accessibilities which assume a usage of multiple carriers along a single way. This report presents a tool that realizes such a disaggregated computation of accessibility measures. A strong focus is put on the tool’s internal computation steps for supporting a reference to potential users.

Keywords—Accessibility measures; performance indicators; intermodality.

I. INTRODUCTION

In [1], we presented tool for computing so-called accessibility measures. With a raising awareness about (road) traffic’s impacts on the environment and the quality of life, especially in urban areas [2], measures and incentives for supporting sustainable modes of transport become increasingly important. In conjunction, proper performance indicators are needed for determining areas that need improvements as well as for measuring the results of planned or already performed measures. One class of performance indicators for valuing a given area or location are so-called accessibility measures [3]. Briefly spoken, accessibility measures describe how well an area or a location is connected to the surrounding activity places taking into account different modes of travel.

Following Litman [4], the concept of accessibility measures introduces a paradigm shift in transport planning. Conventional mobility-based measures cover mainly the performance of the road infrastructure, including the average speed, capacity, or, when looking at traffic participants, the vehicle miles travelled. In contrary, accessibility measures “favor different strategies, including improvements to alternative modes, incentives to change travel behavior, and more accessible land use” [5]. Summarizing, in contrary to

mobility-based measures, which focus on automobile traffic, accessibility measures put people into focus [5].

Meanwhile, accessibility measures are a common tool for evaluating a region’s transport offer and are not only used in academic context, but also by local administrations [6][7]. Accessibility measures take into account the infrastructure, the distribution of localities and inhabitants in space, the passengers’ preferences for using different modes of transport as well as their disabilities [3].

This paper presents a tool for computing so-called contour accessibility measures at the fine-grained level of detail of single buildings and the transportation network. Often, accessibility measures are used at the coarser level of so-called “travel analysis zones” (TAZ). TAZs usually divide a region such as a city or a bigger area into cells with a most possible homogenous travel behavior. So-called macroscopic demand models compute the amount of traffic between such TAZs, and macroscopic land-use models describe the attributes of locations at this level for computing the development of cities or regions. Increasingly, such macroscopic approaches are replaced by microscopic models where every single entity – household, person or vehicle in transportation context – is modelled and simulated individually. Accordingly, fine-grained “microscopic” approaches for computing accessibility are attempted.

The tool for computing accessibility measures described herein was developed for the project “Urbane Mobilität” [8][9], or “UrMo” for short, and was thereby named “UrMo Accessibility Computer”, or abbreviated “UrMoAC”. In the following, “UrMoAC” and “the tool” are used synonymously. The topic of the project “Urbane Mobilität” is intermodality – travelling using different modes of transport (e.g., subsequently riding a bike, using the public transport, and walking) along a single journey [10]. Accordingly, the tool described in [1] was extended by the possibility to compute intermodal accessibility measures to meet the project’s scope.

Besides describing the extensions needed for computing intermodal accessibility measures, the tool’s functionalities, including reading and processing input data, computing individual accessibility measures and aggregating them in spatial means as well as the generation of outputs is described in the following. The focus on the tool’s internals

shall support potential users with an in-depth understanding about the tool.

The remainder is structured as following. In Section II, a short overview on accessibility measures is given. Section III lists the requirements of the project. In Section IV, implementation details are given. Section V shows some use cases and visualization possibilities. This report closes with a summary and outlook in Section VI.

II. ACCESSIBILITY MEASURES

“Accessibility” is not a single, well-defined function, but rather a set of concepts, see [3][11]. One common understanding is that accessibility is a compound measure that describes how many locations can be approached within a given time from a given starting point. Compound, as accessibility consists of two parts. The first is the space that can be covered within the given limits. The second one is the existence of locations of the investigated type within this accessible space. But this is a very brief view at accessibility measures. [3] and [11] give summaries of accessibility measures’ classes and distinguish between “spatial separation measures”, “contour measures”, “gravity measures”, “competition measures”, “time-space measures”, “utility measures”, and “network measures”. One can treat this set of measurements as a class hierarchy of continuous attempts to regard new information, the prior class did not consider. Figure 1 summarizes this hierarchy, which is also briefly discussed in the following.

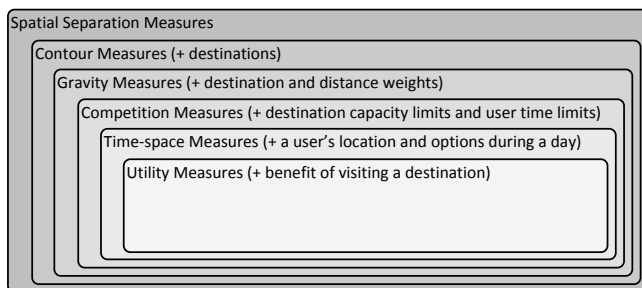


Figure 1. Hierarchy of accessibility measures.

“Spatial separation measures” describe how well (or bad) an area is connected to other areas. Here, distances or travel times are used as a basic measurand. “Contour measures” introduce the land use aspect by incorporating the number of destinations that can be accessed in a given limit, usually given by a maximum travel time. Usually, this number is the desired result of “contour measures”. “Gravity measures” add a weight to the accessible destinations, usually including the respective destination’s attractiveness (e.g., number of employees at a work location or the size of a shop) and a decay imposed by the distance from the source to the respective destination. “Competition measures” additionally incorporate time limits of the users as well as capacity limits of the destinations, partially also the fact that when investigating destinations of the same type, e.g., shops with the same brand, only the nearest ones are of relevance and more distant ones are neglected. “Time-space measures” look at accessible locations taking into account the changing

position and the changes in availability of the transport modes that are available for a single user. Finally, “utility measures” incorporate the benefit a user gains by visiting accessible locations.

One may note that “network measures” have been listed before, yet were not discussed. Although counted as accessibility measures, network measures concentrate on the transportation network, measuring its attributes, such as connectivity, distances, and transfer times. Thereby, in contrary to the other described accessibility measures, they describe the transportation network and not locations or areas.

The inclusion of further weights, attributes and constraints into the computation of accessibilities is meant to increase the fitness of the measures to describe real-world needs and to include the anticipation of accessibility by users. Yet, one may note that these extensions raise the need for corresponding data. Representations of the transportation network as well as positions of locations as needed by contour measures are available, even as a part of publicly available data such as OpenStreetMap (OSM) [12]. But weighting functions, the users’ time limits, constraints, and daily activities or the benefits of visiting a certain location can only be guessed, extrapolated from surveys, or computed using models and are thereby usually vague.

III. REQUIREMENTS AND DESIGN

In the following, the derivation of the requirements for the discussed tool will be outlined. First, the context of the project “Urbane Mobilität” the tool was developed for will be given. Then, the derived requirements are presented, followed by some design decisions.

A. Project Context

Besides evaluating nowadays intermodal mobility behavior, the project “Urbane Mobilität” aims at predicting the effects of measures for increasing the share of intermodality in everyday choice of transport modes to use. Three simulation models are used for this purpose: the agent-based demand model “Travel Activity Pattern Simulator” (TAPAS) [13][14], the microscopic traffic flow simulation “Simulation of Urban MObility” (SUMO) [15][16], and the location choice model “SimulAting Location Demand and Supply in Urban Agglomerations” (SALSA) [17].

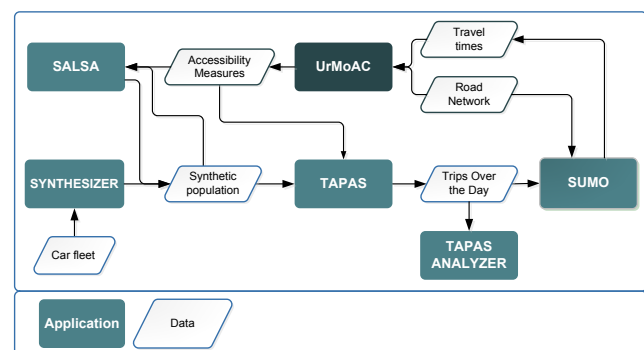


Figure 2. Coupled simulations used in project “Urbane Mobilität”.

Within the project context, SALSA is applied for computing long-term mobility decisions of the population of Berlin by choosing a place of residence. TAPAS models medium- and short-term mobility decisions, mainly the selection of the transport mode, but as well the locations to visit during a day. This is done individually for each of the persons the modeled population in the analysis region consists of. Finally, SUMO simulates the traffic on the transportation network using the demand generated by TAPAS and returns travel time information. This information is given back to TAPAS and SALSA for iteratively obtaining a valid representation of the traffic in the city. The iteration ends as soon as an equilibrium state between the demand and the resulting travel times is reached. The overall workflow is presented in Figure 2. The shown “SYNTHESIZER” application is responsible for generating a disaggregated population [18].

Supporting SALSA with data about a location’s accessibility was the major reason for developing an accessibility computation tool. SALSA itself is macroscopic – locations are grouped into areas at the level of so-called “Teilverkehrszellen” (TVZ, English: sub traffic analysis zone) that represent the TAZ SALSA uses. Berlin’s 1223 TVZs are shown in Figure 3. Albeit being microscopic (agent-based) in its nature, also the demand model TAPAS requires matrices that describe travel time and distances between the centers of TVZ.

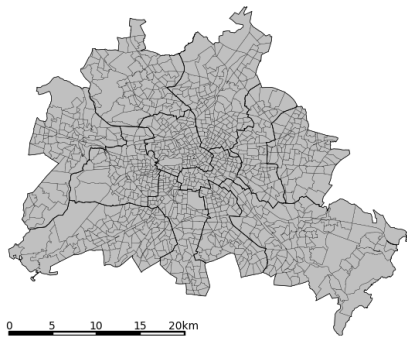


Figure 3. Segmentation of Berlin into “Teilverkehrszellen” (TVZ) as used by SALSA.

SALSA uses a large number of different measures for describing different aspects of the dwellings within a given TAZ. Besides attributes such as the average price, size, or construction year, different accessibility measures can be found among them. It is not known which measures are significant a-priori. Instead, different measures have to be computed first, and their significance has to be determined afterwards by using correlation and regression models that explain their influence in choosing the respective inhabitation area [17][19].

As a starting point, commonly found accessibility measures were defined to be tested for their relevance in household location choice. They include measures such as:

- average travel time to other zones;
- average distance to other zones;
- travel time to nearest commercial center;

- travel time to nearest railway station;
- travel time to closest grocery store ($\geq 200\text{m}^2$);
- travel time to closest small park ($\geq 10000\text{m}^2$);
- grocery retail floor space within 10min travel time;
- green space area within 30min travel time;
- number of jobs within 30min travel time;
- travel time to closest large park ($\geq 50000\text{m}^2$).

One may note that the large variety of accessibility measures used by SALSA is not only a result of using different types of sources and destinations such as dwellings, job locations, shops, or parks. Instead, one may as well find different rules for limiting the investigated area, different types of aggregation, or an optional collection of a variable that is attached to the accessible destinations. The large variety can yet be simplified to the formula (1) that describe the accessibility of the area Z_i in means of a contour measure:

$$A(Z_{i,m}^{res}) = \frac{\sum_{\{O_k \in Z_i\}} \sum_l (g_O(O_k) \cdot g_D(D_l) \cdot f_{res}(c_{k,l,m}))}{\sum_{\{O_k \in Z_i\}} \sum_l (g_O(O_k) \cdot f_{res}(c_{k,l,m}))} \quad (1)$$

with:

$A(Z_{i,m}^{res})$: the accessibility of aggregation area Z_i when

using the mode m , bound by the limit res

i : the index of the source aggregation area (e.g., a TAZ)

m : the used mode of transport

k : the index of a single source (e.g., building)

l : the index of a single destination

O_k : source k

D_l : destination l

$g_O(O_k)$: the weight of source O_k (e.g., the number of households)

$g_D(D_l)$: the weight of destination D_l (e.g., the number of work places)

$c_{k,l,m}$: the costs of the route (e.g., the travel time) between k and l when using mode m

$f_{res}(c_{k,l,m})$: restriction function that realizes the limits

Hereby, the different limits (nearest destination, max. number of accessible destination, maximum travel time or distance) can be realized by choosing a proper restriction function f_{res} . E.g., the following restriction formula (2) can be used for limiting the accessible destinations by a maximum travel time c_{max} :

$$f_{res}(c_{k,l,m}) = \begin{cases} c_{k,l,m} & \text{if } c_{k,l,m} \leq c_{max} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The named accessibility measures have to be computed for the transport modes “walking”, “bicycle”, “motorized individual traffic” (MIT), and “public transport” (PT). In addition, as the project’s scope is intermodality, these

measures have as well to be computed for the major combinations of different transport modes, namely “bicycle and PT” and “MIT and PT”. Here, one has to consider that some public transport modes allow entraining a bicycle while others do not. Private cars, of course, have always to be left at the place the public transport is entered and after leaving the public transport, the remaining way has always to be passed by foot. Using a car- or bike-sharing service for a part of the route is currently neglected.

B. Requirements Summary

All formulated accessibility measures needed by SALSA fall into the class of “contour measures”. The measures needed by TAPAS belong to the class of “spatial separation measures”, which are the superset of “contour measures”. Thereby, the tool should be able to compute “contour measures” and cover “spatial separation measures”.

For computing the needed accessibility measures, the tool shall read information about the sources and the destinations from the project’s database as the underlying data used in the “UrMo” project is available in a disaggregate manner and is stored as unique tables in a database. This includes the positions of dwellings, shops and job opportunities. Accessibility measures have to be computed regarding the traffic on roads that affects the travel times. The application shall support the modes “walking”, “bicycling”, “motorized individual traffic”, and “public transport”. All of those should be routed only at the roads they are legally allowed to use. In addition, intermodal combinations of public transport with a private car or alternatively with a bike shall be supported. For modelling public transport, real-world schedules have to be used.

Albeit using disaggregated descriptions of the sources and destinations, the accessibility measures shall be computed for TAZ areas. The tool shall provide different criteria for limiting the accessible space, namely routing to the first encountered destination, routing bound by a time limit or unbound routing over the complete analyzed area. In the first and the last case, the travel times and distances shall be computed. When searching bound by the travel time, the number of accessible locations, optionally weighted by an attribute is demanded.

Neither the computation time nor the needed memory was limited as the accessibility measures are usually computed once for a given area and no iterations are needed.

C. Major Design Considerations

The major design aspect was to use disaggregated data as a starting point even though SALSA uses measures aggregated at the level of TAZ. This was motivated by the availability of disaggregated positions of sources and destinations and the attempted usage of a digital road network representation that includes the information about the allowed transport modes and velocities. This approach resembles the current development of transport and land-use models where disaggregated, microscopic models are increasingly replacing macroscopic models. A fine-grained accessibility computation makes use of available disaggregated data, should be more exact than macroscopic

approaches and may come along with the inclusion of further information, such as elevators, stairs or other hindrances. Thereby, it not only fulfills the formulated requirements, but is also extensible for future research questions and tasks.

The tool was written in the Java programming language. Currently, it is a command line application what means that no graphical user interface is provided. This is surely a usability restriction, as operating applications at the command line is not common to a large number of users. Yet, besides reducing the implementation effort, command line applications may be run from external scripts or batch files more easily than ones having a graphical user interface what increases the ability to compute a large number of different accessibility measures with least manual interaction. Being a command line application, the data to read and the processing itself is controlled via command line parameters.

The tool computes accessibility measures for a defined set of sources and destinations, yet only for a single mode or intermodal mode combination and for a single route starting time at once. For comparing the accessibilities of different modes, the tool has to be rerun with different parameters. Within the scope of the “UrMo” project, usually the peak hour at 8:00am is used. When determining accessibility measures for public transport, a date must be additionally given for choosing according public transport rides.

IV. IMPLEMENTATION AND PROGRAM FLOW

The overall workflow of the application consists of reading the necessary data, preprocessing it, performing the computation, and generating the outputs. The individual steps are described detailed in the following subsections.

Albeit the used methods – mainly routing using the Dijkstra algorithm [20] – are neither novel nor complex, the overall tool is very flexible due to some simple features. They include filtering, variable limits, or weighting the sources and will be emphasized in the following subsections.

A. Input Data

While for a basic computation of accessibility measures within an investigated region only the positions of destinations and sources and a road network representation are needed, further input values may get necessary, either for defining the aggregation areas, for regarding the public transport offer, or for using travel times of the motorized individual traffic as caused by the respective traffic volume. In the following subsections, the different input data will be presented, distinguishing the description of a) the sources and the destinations, b) the road network, and c) the public transport offer.

1) Sources and Destinations

For the study area, the application first reads the positions of the sources and destinations from a database. As mentioned, different disaggregated spatial locations can be used as sources and destinations, including dwellings, public transport stops, job locations, shops, parks, etc. Figure 5 shows the first two of the named as a visual example. While some of the locations are represented as their footprints using polygons, the tool currently uses the centroids of them only.

This is surely an approximation that introduces an error. But the error is assumed to be small and better models of the access/egress to the road network could only be achieved if the positions of the dwellings' entrances would be known, what is not the case. In later investigations, the usage of CityGML [21] descriptions of buildings, which include the entrances, is planned. Locations with other geometries than polygons may be used as sources/destinations as well, as long as the geometry can be converted to simple points by the used geometry library "JTS" [22]. Besides the need for having a geometry for being allocated in space, each source and destination must have a unique numeric identifier (ID) for later reference.



Figure 4. Examples for sources and destinations – dwellings (grey polygons) and bus stop positions (red dots) around the DLR in Berlin (light grey).

When reading sources/destinations from a database a simple filter realized as a SQL WHERE-clause can be given. For bigger datasets – e.g., the locations of all shops in a city – this can be used to select only a subset of the locations, e.g., only groceries or only shops that are bigger than a given threshold. The filter may either be applied to a column of the database table that defines the respective source/destination or be computed using a more complex query, for example using the PostGIS [23] extensions. Thereby, operations, such as filtering objects by their size are possible, even if this information is not explicitly given as an own database table column.

To both, sources and destinations, a numerical value can be attached. For sources, this value is used for weighting the individual source's influence when computing the average value by aggregation, see also Section IV.D. A possible application is weighting dwellings by the number of persons inhabiting them. For destinations, the value's semantics are kept abstract and, if given, the values of the accessible destinations are added together. Usual applications are counting the number of jobs accessible from a location or, as a more abstract measure – determining the selling space of groceries in a specific range. Similar to the filtering option described above, the numerical value may either be read directly from the database or be the result of a more complex computation as long as it is based on the values of a single row of a PostGIS database table.

In summary, the definition of a source and/or a destination consists of a set of variables, which is given in Table I. The database tables used within the project had a consistent naming of these variables. Yet, during further usage of the tool, datasets have occurred where both the ID as well as the geometry were stored under different names. To avoid changing the used database tables, the tool has been extended by possibilities to name the database columns that contain the IDs and the geometries.

TABLE I. VARIABLES OF SOURCES AND DESTINATIONS READ FROM THE DATABASE

Variable	Meaning	Type	Default Name
ID	The object's identifier	bigint	rid
Geometry	The object's location	geometry	the_geom
Value	The object's weight	real	none (opt.)

Besides the sources and destinations, further geometry objects may be loaded that describe the areas the computed accessibilities for single source/destination relationships shall be aggregated within. Here, the objects' geometries must be of the type polygon and – as for the sources and destinations – they must support a unique ID. A weight is not loaded. The usage of aggregation areas will be described more detailed in Section IV.D.

2) Road Network

The tool uses a specific database representation of a road network as given in Table II.

TABLE II. THE ATTRIBUTES THE EDGES OF THE ROAD NETWORK ARE DEFINED BY

Column	Meaning	Type
id	Edge (road) identifier	serial
oid	Original edge identifier / name	text
nodefrom	ID of the node the edge starts at	bigint
nodeto	ID of the node the edge ends at	bigint
numlanes	The number of the edge's lanes	smallint
length	The edge's length (in meters)	double
vmax	The allowed speed (in km/h)	double
street_type	Abstract street type (not used)	text
mode_walk	Is walking allowed / possible?	boolean
mode_bike	Is riding a bike allowed?	boolean
mode_pt	Are PT carriers allowed? (not used)	boolean
mode_mit	Is driving a private car allowed?	boolean
the_geom	The edge's geometry	multilinestring

Currently, three external import modules exist that generate such representations from the free OpenStreetMap data, from Navteq networks, and from PTV VISUM networks. The import of OSM networks needs some

preprocessing steps, which mainly include a) the determination of intersections by selecting nodes used by more than one way, and b) the consolidation of access and one-way information for obtaining a unidirectional network with access information for different transport modes. After performing these preprocessing steps, the import script writes the road network into the database.

Please note that albeit the geometry is currently stored as a PostGIS “MultiLineString” geometry object, it in fact always consists of a single “LineString” only. It is assumed that this will be corrected in the near future. The road network of the city of Berlin as used in subsequently presented evaluations consists of 709,713 edges (roads) and 269,604 nodes. It is shown in Figure 5.

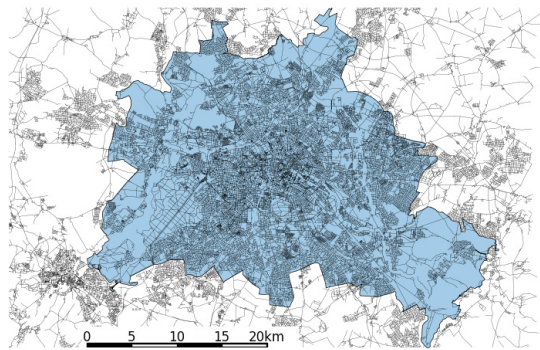


Figure 5. The used road network (black) of the city of Berlin (blue).

Nowadays, road network representations often distinguish between bi- and unidirectional roads. A road with a green space or a parking lane between both directions is usually encoded as two unidirectional roads, each with its own geometry. On the contrary, if both directions are not separated, the road is marked as bidirectional and has one geometry only. The network import tools translate this information and generate a road network representation that contains unidirectional edges only. Yet, for allowing pedestrians to use a road in both directions, the tool additionally builds an edge in the opposite direction for those edges that permit walking and for which no opposite direction exists. This additionally built edge allows walking only. The usage of a road for bicycling into a – legally forbidden – direction what may be found in real life is not supported.

As shown in Table II, the network’s edges include the information about the maximum speed allowed for motorized individual traffic. Yet, this information disregards the decrease in average velocities over the day caused by a changing traffic volume. To accommodate this, additional information about the average speeds at each edge over time can be read from a database. Table III shows the structure of such tables. Within the project, the travel time information is generated from the outputs of the microscopic traffic flow simulation SUMO. An import script, which reads SUMO’s edge-based traffic measures (see documentation at [16]) and stores them into the database, is supported.

TABLE III. STRUCTURE OF THE ROWS WITHIN THE EDGES’ SPEED PROFILES TABLE

Column	Meaning	Type
ibegin	Begin of the time interval (in s)	real
iend	End of the time interval (in s)	real
eid	The original edge identifier / name	text
speed	The average speed (in m/s)	real

3) Public Transport Offer

Optionally, the tool additionally reads a public transport network using a database representation of a General Transit Feed Specification (GTFS) [24] data set. The database representation keeps the original GTFS format and a script for importing GTFS files into database tables is supported. “UrMoAC” itself reads the information about stops, routes, the services offered at the specified date, and the respective trip schedules. As soon as a read public transport line connects two stops, a new public transport edge that connects these stops is added to the network. Instead of the information about the allowed speed, these public transport edges hold the read connections between them, including the departure and arrival times. Using these connections during routing will be discussed in Section IV.C.3.

When computing intermodal accessibility measures, a definition of entrainment possibilities is necessary for a complete description of the intermodal transport offer. This description is again stored in an optionally read database table which has the structure as given in Table IV. For a maximum flexibility in conjunction with keeping the definition of public transport carriers as given in GTFS data, the carrier is described using two fields. One may note that time-dependent entrainment is not supported and that all lines using the same carrier will have the same entrainment constraints.

TABLE IV. THE DEFINITION OF ENTRAINMENT

Column	Meaning	Type
carrier	The name of the carrier vehicle (mode name, e.g., “pt” or “car”)	character(40)
carrier_subtype	GTFS route type enum	smallint
carried	The name of the carried vehicle (mode name)	character(40)

4) Map Projection

While dealing with real-world descriptions of geospatial data, the tool has to cope with different types of geographic projections. Because the desired outputs use the metric system, usually a re-projection of the data’s original coordinate system to metric measures is needed. The tool allows defining the projection to use on the command line and when reading input data, all coordinates are transformed into this target projection using native PostGIS functions.

B. Preprocessing Data After Reading

In a first step, the objects read from the database to route between are allocated on the road network. As mentioned,

their centroid is computed first. For every centroid representing a source/destination, the nearest road that allows to be passed using the investigated mode of transport is determined. A direct, shortest access to this road is assumed, being usually a line normal to the road's shape at the point that is nearest to the object. Yet, in some cases, a source/destination may be located behind the nearest road's beginning or end.

A spatial index, namely the RTree [25] implementation from the Java Spatial Index library [26], is used during this process for increasing the computation speed by searching for roads in the objects' vicinities only. It should be noted that because the RTree structure stores the roads via their bounding box, obtaining the closest road for a given point is not sufficient as the road may be located at the opposite site of the bounding box than the point. Figure 6 shows the connections between sources/destinations and the road network.

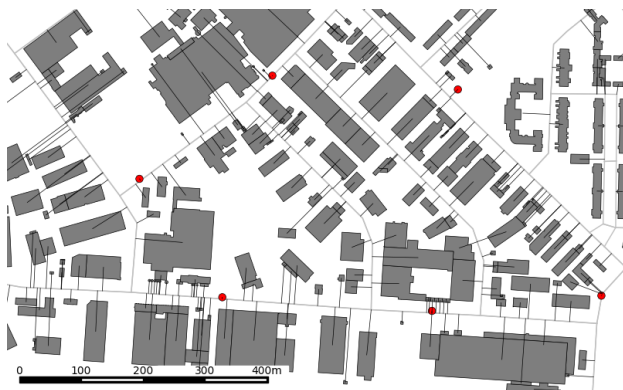


Figure 6. Attaching sources/destinations to the road network; buildings (grey polygons) and public transport stations (black points) are connected to the road network (grey lines) via access paths (black lines).

A second issue to solve when allocating objects to roads is the identification of the correct road direction. As mentioned, bidirectional roads are often represented by a single "multiline", a sequence of lines, with no geometrical distinction between both directions. Thereby, both directions have the same distance to the source's/destination's position and an arbitrary one of them would be chosen. This ambiguity is solved by determining the road direction and mapping it onto the direction the source/destination is located right to. The direction is computed as given in formula (3).

$$dir = (x_{le} - x_{lb})(y_p - y_{lb}) - (x_p - x_{lb})(y_{le} - y_{lb}) \quad (3)$$

with:

dir: direction (right if negative, left if positive)

x_A : the x-coordinate of point *A*

y_A : the y-coordinate of point *A*

where *A* is one of the following points:

lb: begin of the line nearest to the point

le: end of the line nearest to the point

p: the point (location position)

Usually, sources and destinations are located besides a road. Yet, in some cases, the access to bigger, areal locations (e.g., parks) has to be computed. Such amenities can be usually approached from different directions and are often crossed by roads. In such cases, it does not make sense to allocate them at a single road. Instead, they have to be assigned to all edges that surround and/or cross them. Yet, this is a matter of future extensions.

Optionally read positions of public transport stops have to be assigned to the previously read road graph as well. This is done in a similar way as for the sources/destinations. For each position of a public transport stop, the nearest road is determined and the stop is mapped onto it. Both directions of the road are used if the road was originally bidirectional. Albeit OSM partially includes detailed information about paths across a station or a hub, stations are allocated at the road network via their centroids only. In contrast to sources/destinations, the edge(s) a stop is located at is split at the position of the stop. To this new node, pathways that connect the respective stop and allow to be passed using the modes "walking" and "bicycle" are added. This forms a connection between the original road network and the public transport stops.

C. Processing

In its basics, the process of computing accessibility measures is very straightforward. The application iterates over the read sources. For each, the road network is scanned using the Dijkstra algorithm, taking the available and allowed modes of transport and the respective travel times into account. Yet, the wish to compute fine-grained accessibility measures including public transport and intermodal mobility as well as supporting variable routing limits made several extensions necessary, which will be described in the following.

1) Approaching a Destination

One major extension is needed due to the allocation of the sources/destinations along a unidirectional edge. When approaching a destination that is located at the opposite road site, the router would need to move along the current edge to the next intersection (node) to change the moving direction by entering the opposite edge for finally approaching the respective destination. For avoiding this behavior, the router assumes that the edge may be crossed at the position of the destination. Figure 7 visualizes the difference per example.

To implement the functionality of "crossing the road", the router starts in both directions of the edge the respectively processed source is located at. In addition, when approaching an edge, not only the destinations at this edge, but also those that are located at the opposite edge, if given, are collected. One may note that this makes a post-processing of the collected destinations necessary, as discussed later in Section IV.C.4. In case of edges where both directions are separated, e.g., by a green space, crossing (both) is assumed to be not possible.



Figure 7. Difference in approaching a public transport stop (S) from a dwelling (D) if crossing the road is not possible (left) or possible (right).

2) Travel Times

The travel speed used for determining the travel time needed to pass an edge mainly depends on the chosen travel mode. The tool uses the speeds as given in Table V. As shown, an edge's speed restrictions are only regarded for motorized individual traffic and bicycling. Having a low velocity, walking is "naturally" bound to the roads' speed limits.

TABLE V. SPEEDS OF THE MODELLED MODES

Mode	Speed
walking	5km/h
bicycling	15km/h
MIT	minimum of 200km/h and the road's speed limit; the latter optionally replaced by read travel time timelines
PT	time schedule (from GTFS)

It should be noted that using the length of edges and the travelling speed for computing the travel time of motorized traffic is not correct as in this case, additional travel time delays posed by traffic lights and other traffic participants are neglected. This issue is solved by additionally reading speed time lines for the loaded edges, as outlined in Section IV.A.2. When entering an edge, these loaded time lines are scanned for finding a time span that matches the time the edge was entered at. The speed stored for this time span is used. There is an additional workaround in case the loaded speed is equal to zero, what e.g., may happen due to grid locks within the simulation that was used to compute the speed timelines. In this case, the half of the allowed velocity is used as travelling speed.

Public transport connections are treated in a different way. When encountering a public transport stop, the available connections to next stations are regarded, for each line operating at this stop. The router chooses the connection with the earliest departure time that is higher than or equal to the arrival time at the stop and returns the time difference between the time of the arrival at the subsequent stop and the time the current stop was approached at.

3) Changing the Mode and Intermodality

The tool is called with a list of available modes of transport as a command line option. When starting the routing, the first of those is selected for being the currently used mode of transport. Yet, the available modes are kept for later routing steps.

When approaching an edge, modes not allowed to be used at this edge are removed from the list of available modes. The mode of transport to switch to is chosen by selecting the mode from the remaining available ones that is the fastest one for passing the edge. The priority queue of the extended Dijkstra algorithm holds not only the visited nodes, but as well the remaining available modes when approaching them. Given this, the consecutive competition of modes and mode combinations along the route is maintained while progressing through the road network.

Public transport stops are connected to the remaining network using edges that allow the modes "walking" and "bicycling". Other modes of transport are therefore abandoned when approaching a public transport stop. Up to now, no penalties for leaving a car or for parking a car are regarded. Allowing using a bike when approaching a stop may be wrong as often one has to dismount and walk. But, as discussed in Section IV.B, no fine-grained information about the access paths from the road network to the public transport stops are given. The stops are only connected to the road network using a shortest line. Allowing to approach a stop by cycling seems therefore to be a good solution for keeping the bike as an available travelling mode for simulating the entrainment of bikes in public transport.

Whether the bike can be entrained between two stops or not is stored in the respective connection between those stops and is based on the information about the carrier used between these stops and the read entrainment table.

4) Variable Limits

One of the tool's strengths is the capability to use different limits for routing. The search for destinations ends as soon as one of the following limits that has to be specified at the command line is reached:

- maximum travel time: stops as soon as the given travel time is exceeded;
- maximum distance: stops as soon as all objects in the given distance have been visited;
- maximum number: stops as soon as the given number of destinations has been visited;
- maximum variable sum: stops as soon as the sum of the variable attached to the destinations is above the given number;
- shortest: stops as soon the first destination is reached.

It must be noted that the determination of the accessible destinations does not end as soon as the limit was reached for the first time. Because the destinations are allocated along the edges, the edges' length cannot be used as a proxy for the search depth in Dijkstra. Even when searching for a nearest destination and finding one at the starting edge, the travel time between the current source and the found destination may be higher than that to a destination located at a different

edge. Thereby, the identified destinations have to be temporarily stored and the search limit has to be adapted to be at least the same as the distance to the farthest destination found so far. Storing found destinations temporarily is also necessary as they are collected from both, the travelled edge as well as an optionally given opposite edge during the routing process. The correct set of destinations that fulfills the limits can be only determined by post-processing the destinations collected during routing. Thereby, after finishing the search with adapted limits, the collected results are sorted by their travel time and those exceeding the limit are removed from the collection.

5) Routing Results

For each source, the result of the routing process consists of a set of edges with assigned destinations that are accessible in given limits. For each edge, the travel time, the distance, the used and the available modes, the PT line in case PT was used as well as the information whether this edge was on the opposite side of the originally travelled one are given. Furthermore, the edge information contains a pointer to the predecessor edge for reconstructing the complete route.

The so obtained distinct paths between a single source and the accessible destinations are given to aggregators and output generators discussed in the following section.

D. Aggregation and Output Generation

Given the results for single source/destination relationships, the tool supports several outputs and is capable to apply different kinds of aggregation. Both will be discussed individually in the following.

1) Generated Measures

The results for each source/destination are given to so-called “measurement generators” first. These measurement generators process the routing results and transform them into measurements of different kinds. This kind of post-processing offers some benefits, such as the possibility to generate only the desired outputs or the reduction of needed memory by avoiding keeping unnecessary information. The currently available measurements generators are described in the following. All outputs can be either written to a file or into a database table generated by the tool.

The “n:m output” computes the basic accessibility measures, namely the distance, the travel time, the number of accessed destinations, and their weight. When using an aggregation option, these numbers represent average values, optionally weighted by the sources’ weights. The given IDs of the source (fid) and the destination (sid) either name the source / destination itself or, in case of aggregating them by areas they are located within, the ID of the respective aggregation area. If the aggregation option “all” is set for sources and/or destinations, the respective field contains the value -1. The structure of an “n:m output” database table is given in Table VI.

TABLE VI. THE STRUCTURE OF THE “N:M OUTPUT”

Column	Meaning	Type
fid	The id of the source (or source aggregation area)	bigint
sid	The id of the destination (or destination aggregation area)	bigint
avg_distance	The (average when aggregating) distance between the source and the destination in meters	real
avg_tt	The (average when aggregating) travel time between the source and the destination in seconds	real
avg_num	The (average when aggregating) number of seen destinations	real
avg_value	The (average when aggregating) weight of seen destinations	real

The “extended n:m output” extends the “n:m output” by further measurements that can only be computed by traversing each path between a source and a destination. The additional measures include the personal energy needed for the trip, the trip’s price, the generated amount of CO₂ emissions, and the list of used modes. The structure of the table is given in Table VII.

TABLE VII. THE STRUCTURE OF THE “EXTENDED N:M OUTPUT”

Column	Meaning	Type
fid	See Table VI	bigint
sid	See Table VI	bigint
avg_distance	See Table VI	real
avg_tt	See Table VI	real
avg_v	The (average when aggregating) velocity between the source and the destination in meters per second	real
avg_num	See Table VI	real
avg_value	See Table VI	real
avg_kcal	The (average when aggregating) personal energy consumption in kcal	real
avg_price	The (average when aggregating) price of the trip/ride in Euro	real
avg_co2	The (average when aggregating) CO ₂ emission in g	real
modes	The modes used during the trip	text

The mode-dependent constants for the generated measurements are given in Table VIII. The CO₂ emission is based on information from the German federal environment agency [27]. ADAC’s (Germany’s major automotive club) price list for passenger vehicles was used for determining the costs of using MIT while the price for using PT is based on the assumption of the availability of an annual pass and the usage of PT three times a day within 20 working days per month. The values for personal energy consumption were collected and cross-checked using different web sites, e.g., <http://gesuender-abnehmen.com/>. The work in [28] explains the derivation of these constants in more detail.

TABLE VIII. CONSTANTS USED FOR COMPUTING EXTENDED TRIP MEASURES

Mode	CO ₂	Personal Energy Consumption	Price
walking	0g/km	280kcal/h	0€/km
bicycling	0g/km	300kcal/h	0€/km
MIT	150g/km	85kcal/h	0.45€/km
PT	75g/km	170kcal/h	0.95€/trip

In addition, the tool supports a so-called “interchanges output” that writes information about the usage of interchanges. Both, public transport interchanges as well as the points (nodes) at which the mode of transport is changed are counted. For each interchange, the ID of the PT stop / network node the interchange took place at, the mode or public transport line the node was approached by, and the one used to leave the node are given. In addition, the number of interchanges at this stop between these modes is given. The structure of this output is given in Table IX.

TABLE IX. THE STRUCTURE OF THE “INTERCHANGES OUTPUT”

Column	Meaning	Type
fid	See Table VI	bigint
sid	See Table VI	bigint
halt	The ID of the described public transport stop or road network node	text
line_from	The mode or line used to approach the stop / node	text
line_to	The mode or line used to leave the stop / node	text
num	The number of interchanges at this node of the given type	bigint

1) Aggregation

Another major feature of the tool is the capability to perform different kinds of aggregation. For both, the sources and the destinations, additional aggregation areas (e.g., TAZ) can be read from the database. If given for the sources, the measures collected by routing from individual sources within an aggregation area are averaged. When being applied to the destinations, the values of all destinations within a given aggregation area that are accessible in the given limits will be joined and averaged. Additionally, an “aggregate all” option is available for joining the measures for all sources or respectively destinations. Not all of the possible aggregation combinations are meaningful.

Subdivision of a city into abstract areas, such as TAZ, can be found quite often when dealing with the urban development of a single city or regarding the needs of single districts or quarters. Yet, for a better comparability between different cities or areas, aggregation areas of similar size and shape are of benefit. In such cases, so-called fishnet- or hexagon-grids are used. The tool itself does not compute such grids, yet they can be generated a-priori and loaded as aggregation areas.

V. USE CASES AND VISUALISATION

In the following, some examples for using the tool are given, focusing on showing the tool’s capabilities and on pointing out some findings on using accessibility measures.

A. Isochrones

Isochrones are a prominent method for visualizing a single source’s accessibility. The tool is capable to compute isochrones by limiting the number of sources to one and searching for destinations within a given time limit. Given the list of destinations with assigned travel times needed to access them, isochrones can be visualized using, e.g., the “contourf” method from matplotlib [29], a visualization library for the Python programming language. An example for an isochrones generated this way is shown in Figure 8 for the area accessible within half an hour when starting at the Institute of Transport Research in Berlin Adlershof at 8:00am and using public transport in combination with bicycling. The figure shows the typical accessibility “islands” occurring when using fast public transport connections between distant locations.

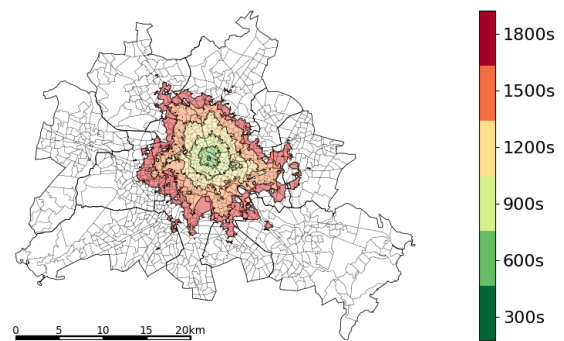


Figure 8. Simple isochrones representation; isochrones for using the public transport and a bicycle, starting at Berlin Alexanderplatz at 8:00am.

When investigating intermodal mobility, one may be interested in visualizing the gain of using public transport in addition to another mode of transport. Figure 9 shows the isochrones when using a bicycle only (light blue) and combining it with public transport (purple).

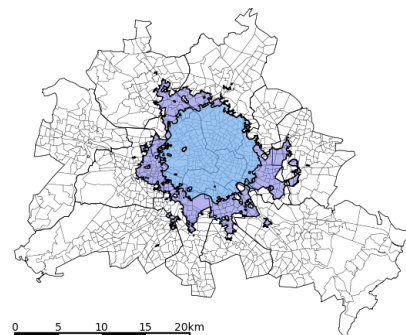


Figure 9. A comparison between the area accessible within half an hour by bike only (light blue) and when combining biking with public transport (purple), starting at Berlin Alexanderplatz at 8:00am.

B. Contour Measures

One often used accessibility measure is the distance or travel time to the next public transport stop. When visualizing the results, one may note that showing disaggregated data (e.g., dwellings) within a bigger area and coloring them is not meaningful, because they vanish due to their small size in comparison to the region and the vacant land. Thereby, an aggregation should be performed. Figure 10 demonstrates this by showing the travel times to the next metro or city rail station for every dwelling individually (top) and aggregated by TAZ (bottom).

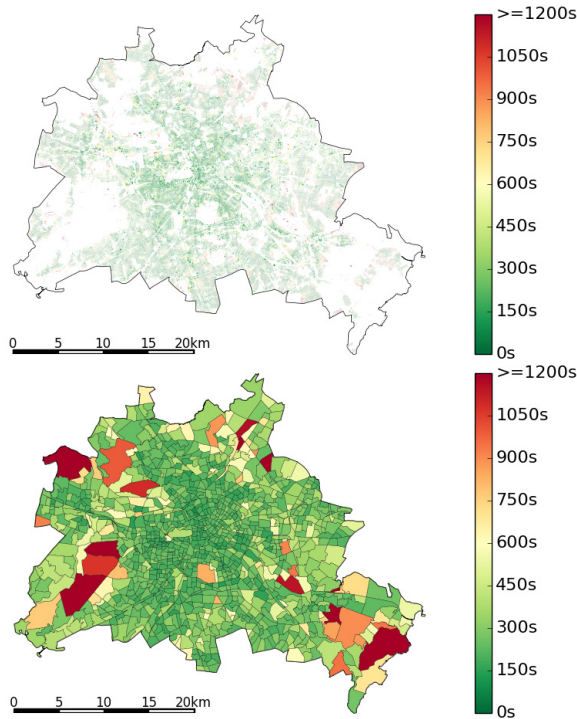


Figure 10. Differences when aggregating accessibility measures.

Using the same example, the influence of weighting sources is shown in Figure 11, which displays the difference between weighted by household number and unweighted travel times from dwellings to the next city rail or metro station aggregated by TAZ.

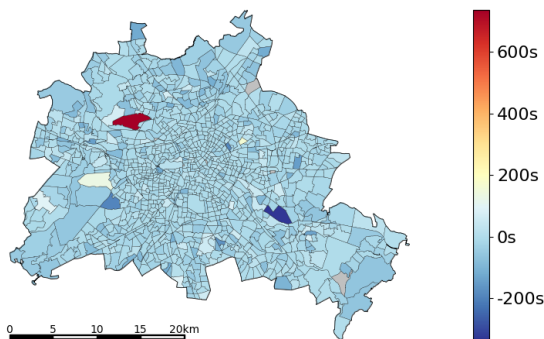


Figure 11. Difference in the access time between a weighted and an unweighted computation.

As visible, neglecting the weight yields in significant deviations.

C. Beyond Travel Time

While intermodality is often appraised to generate less pollution and to be healthier than the monomodal usage of motorized vehicles due to incorporating active modes of transport, only few quantitative evaluations exist. In [29], the performance of intermodality was quantified in means of the personal energy consumption, travel time, price, CO₂ emissions, and the number of accessible places. For this purpose, the number of accessible work places when starting at selected locations in Berlin was computed. The “extended n:m output” was used, and no aggregation was applied. This was done for all implemented modes and mode combinations. The so obtained statistics about mono- and intermodal routes were then evaluated by different means. As an example, Figure 12 shows the performance of the intermodal combination of bicycling and using the public transport against driving a passenger car. The shown lines display the progress when using the respective average velocity. A detailed explanation is given in [28]. Still, one may note that the combination of public transport and bicycling even outperforms the usage of a passenger car.

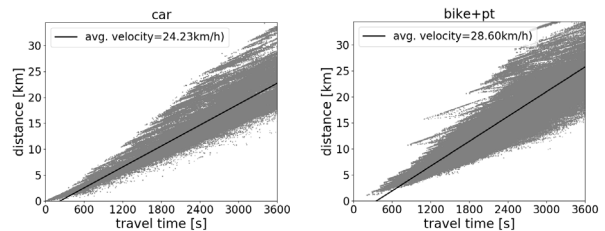


Figure 12. Performance of the intermodal combination of bicycling and using public transport in comparison to driving a passenger car.

Of course, such a disaggregated representation is hard to interpret and raises the amount of needed figures. A summarizing view at the modelled modes’ performance in terms of the speed and the number of work places accessible within one hour is given in Figure 13.

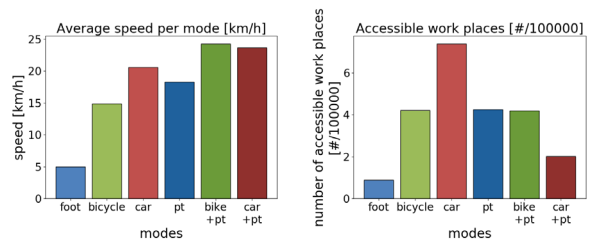


Figure 13. The performance of the mono- and intermodal modes of transport in terms of average speed (left) and the number of accessible work places (right).

One may note that albeit intermodal mode combinations are faster, the number of accessible work places is the same or even lower than for the unimodal modes, especially the car. The reason is that especially when looking at the combination of public transport and a passenger car, only few destinations are approached faster when combining the

modes, even though the combination covers bigger distances. This is already visible in the spread of accessibilities when combining public transport with a bicycle as shown in Figure 12, right. As shown, near work places are missing, because it does not make sense to use the public transport to access them when the direct path using a bicycle is already faster.

As mentioned, other measures than speeds were computed for this study, using the “extended n:m-output”. Figure 14 shows the average price, the average CO₂ emissions, and the average personal energy consumption for using the modelled modes and mode combinations. In summary, the results prove that intermodal mode combinations outperform the monomodal usage of a car in terms of health, pollutant emission, price, and travel times.

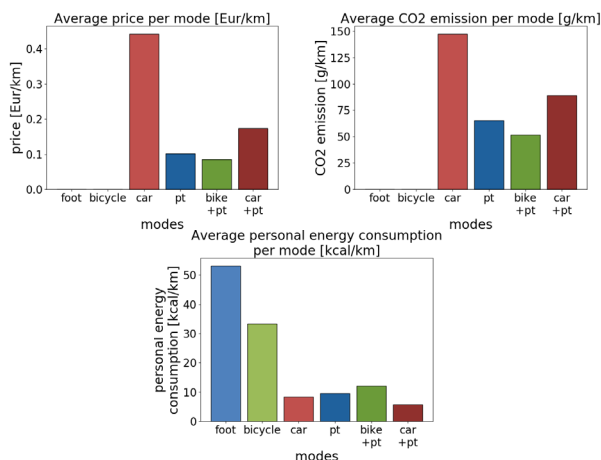


Figure 14. Further performance measures of the mono- and intermodal modes of transport.

VI. SUMMARY AND OUTLOOK

This report presents a tool for computing accessibility measures that can be used for benchmarking areas and for generating measures needed by land use planning models and traffic demand models. The tool follows a fine-grained approach to compute accessibility measures by routing between individual sources and destinations, mainly dwellings, shops, bus stop positions and other man-made objects.

Albeit contour accessibility measures come in many different variations, only some simple methods seem to be sufficient for enabling the tool to compute a large amount of them, fulfilling the requirements given by the “UrMo” project. One important feature are flexible limits, including a maximum travel time, distance, or the possibility to abort the search when a first destination was seen. Another one is the possibility to aggregate the individual sources’ accessibility values into averages for bigger areas, including variable aggregation areas and a variable weighting of the individual sources. Reading sources and destinations from a database and supporting a pre-filtering when doing so has proved to be valuable as well. Finally, attaching values to the destinations is required for computing some of the needed accessibility measures and was accordingly implemented.

Some approaches for computing accessibility measures choose only subsets of sources and destinations for estimating accessibility. But the experiences with the tool described herein do not prove the necessity for reducing the amount of data to process.

While being usable as-is, some improvements to the tool and the data it uses seem to have the capacity to improve the results and enable investigations of further research questions. As mentioned, the currently used representation of sources and destinations via their centroids introduces an error in the paths between the respective source/destination and the road network. Using the nearest position to the road network from a given source’s/destination’s polygon would be possible. Still, this does not regard the positions of a building’s doors or entrances. Routing itself should be extended by approaches for a more realistic person routing, which considers slopes, barriers, elevators, and the possibility to walk across free places. Finally, the user friendliness could be increased by adding a graphical interface to the tool.

Summarizing, it is surprising how much flexibility a very small and simple application can achieve. The release of the tool under an open source license is planned for the near future.

REFERENCES

- [1] D. Krajzewicz and D. Heinrichs, “UrMo Accessibility Computer - A tool for computing contour accessibility measures,” in: SIMUL 2016, The Eighth International Conference on Advances in System Simulation, pp. 56-60, ISBN 978-1-61208-501-2, ISSN 2308-4537, 2016.
- [2] UN Habitat, “Planning and Design for Sustainable Urban Mobility: Global Report on Human Settlements 2013,” Global Report on Human Settlements Series, 978-92-1-132568-3, 2013.
- [3] K. T. Geurs and B. van Wee, “Accessibility evaluation of land-use and transport strategies: review and research directions,” in: Journal of Transport Geography, Volume 12, Issue 2, pp. 127-140, ISSN 0966-6923, doi:10.1016/j.jtrangeo.2003.10.005, 2004.
- [4] T. A. Litman, “Evaluating Quality of Accessibility for Transportation Planning,” in: Transportation Research Board 87th Annual Meeting. No. 08-0495. 2008.
- [5] T. A. Litman, “Evaluating Accessibility for Transportation Planning: Measuring People’s Ability to Reach Desired Goods and Activities,” Victoria Transport Policy Institute, 2016.
- [6] M. Pitot, T. Yigitcanlar, Tan, N. Sipe, and R. Evans, “Land Use and Public Transport Accessibility Index (LUPTAI) tool: the development and pilot application of LUPTAI for the Gold Coast,” 29th Australasian Transport Research Forum, 2006.
- [7] A. Dahlgren, “Geographic Accessibility Analysis - Methods and Application,” Real Estate Science, Department of Technology and Society, Lund University, 2008.
- [8] L. Gebhardt et al., „Intermodal urban mobility: users, uses, and use cases,” in: Transport Research Arena, Elsevier Ltd. Selection and peer-review. Transport Research Arena (TRA), Warsaw, Poland, 2016.
- [9] DLR, “Urbane Mobilität” project web pages, <http://www.urmo.info/>, 2017, last visited on 11th of December 2017.

- [10] W. B. Jones, C. R. Cassady, and R. O. Bowden, "Developing a Standard Definition of Intermodal Transportation," in: *Transportation Law Journal* 27 (3), pp. 345-352, 2000.
- [11] J. Scheurer and C. Curtis, "Accessibility Measures: Overview and Practical Applications," URBANET working paper No. 4, Curtin University of Technology, 2007.
- [12] OpenStreetMap contributors, OpenStreetMap project pages, <http://www.openstreetmap.org/>, 2017, last visited on 11th of December 2017.
- [13] M. Heinrichs, D. Krajzewicz, R. Cyganski, and A. von Schmidt, "Introduction of car sharing into existing car fleets in microscopic travel demand modelling," in: *Personal and Ubiquitous Computing*, pp. 1-11, Springer. DOI: <https://doi.org/10.1007/s00779-017-1031-3>, ISSN 1617-4909, 2017.
- [14] DLR, TAPAS web page, http://www.dlr.de/vf/en/desktopdefault.aspx/tabid-2974/1445_read-29381/, 2017, last visited on 11th of December 2017.
- [15] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO - Simulation of Urban Mobility," in: *International Journal On Advances in Systems and Measurements*, 5 (3&4), pp. 128-138, 2012, ISSN 1942-261x.
- [16] DLR, SUMO web pages, <http://sumo.dlr.de/>, 2017, last visited on 11th of December 2017.
- [17] B. Heldt, K. Gade, and D. Heinrichs, "Challenges of Data Requirements for Modelling Residential Location Choice: the Case of Berlin, Germany," *European Transport Conference 2014*, 2014.
- [18] A. von Schmidt, R. Cyganski, and D. Krajzewicz, „Generierung synthetischer Bevölkerungen für Verkehrsnachfragemodelle - Ein Methodenvergleich am Beispiel von Berlin,“ in: HEUREKA'17 - Optimierung in Verkehr und Transport, pp. 193-210, FGSV-Verlag, ISBN 978-3-86446-177-4, 2017.
- [19] B. Heldt, K. Gade, and D. Heinrichs, "Determination of Attributes Reflecting Household Preferences in Location Choice Models," in: *Transportation Research Procedia*, 19, Elsevier B.V., pp. 119-134, doi: 10.1016/j.trpro.2016.12.073, ISSN 2352-1465, 2016.
- [20] E. W. Dijkstra, "A note on two problems in connexion with graphs," in: *Numerische Mathematik* 1, pp. 269-271. 1959, doi:10.1007/BF01386390.
- [21] 3D Geoinformation group at TU Delft, CityGML homepage, <https://www.citygml.org/>, last visited on 11th of December 2017.
- [22] Location tech (earlier: Vivid Solutions), JTS homepage, <https://www.locationtech.org/projects/technology.jts>, last visited on 11th of December 2017.
- [23] PostGIS PSC, PostGIS homepage, <http://postgis.net/>, last visited on 11th of December 2017.
- [24] Google, General Transit Feed Specification pages, <https://developers.google.com/transit/gtfs/>, 2017, last visited on 11th of December 2017.
- [25] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," in: *Proc. ACM SIGMOD International Conference on Management of Data*, pp. 47-57, 1984, doi:10.1145/602259.602266
- [26] JSI contributors, JSI (Java Spatial Index) RTree Library web pages, <http://jsi.sourceforge.net/>, 2017, last visited on 11th of December 2017.
- [27] Umweltbundesamt, "CO₂-Emissionsminderung im Verkehr in Deutschland," available at: <https://www.umweltbundesamt.de/sites/default/files/medien/461/publikationen/3773.pdf>, 2010.
- [28] L. Gebhardt, D. Krajzewicz, and R. Oostendorp, "Intermodality – key to a more efficient urban transport system?" in: *Proceedings of the 2017 ECCEE Summer Study*, pp. 759-769, ISBN 978-91-983878-1-0 (online)/978-91-983878-0-3 (print) ISSN 2001-7960 (online)/1653-7025 (print), 2017.
- [29] The Matplotlib development team; matplotlib web site, <https://matplotlib.org/>, last visited on 11th of December 2017.