

# UrMo Accessibility Computer

A tool for computing contour accessibility measures

Daniel Krajzewicz and Dirk Heinrichs

Institute of Transport Research

German Aerospace Center

Berlin, Germany

email: daniel.krajzewicz@dlr.de, dirk.heinrichs@dlr.de

**Abstract**— Contour accessibility measures are 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 offers taking the connectivity to its surrounding 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 report describes a tool that realizes this purpose.

**Keywords**—Accessibility measures; performance indicators; intermodality.

## I. INTRODUCTION

With a raising awareness about (road) traffic’s impacts on the environment and the quality of life, especially in urban areas [1], measures and incentives for supporting sustainable modes of transport become increasingly important. Of course, proper performance indicators for determining areas that need improvements as well as for measuring the results of planned or already performed measures are needed. One class of performance indicators for valuing a given area or location are so-called accessibility measures [2]. They describe how well a location is connected to the surrounding activity locations. Accessibility is used for a large number of different applications in the field of transportation and land-use planning.

“Accessibility” is not a single, well-defined function, but rather a set of concepts. 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 reached within the limits. The second one is the existence of locations of the investigated type within this accessible space. But this describes only a subset of known accessibility measures. [2] gives a summary of accessibility measures’ classes and distinguishes between “spatial separation measures”, “contour measures”, “gravity measures”, “competition measures”, “time-space measures”, “utility measures”, and “network measures”.

The different variations of accessibility measures are not the scope of the presented research. Rather, a tool for

computing contour accessibility measures at a fine-grained level of detail of buildings and the transportation network will be described. Often, accessibility measures are used at the level of so-called “traffic 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, macroscopic approaches are replaced by microscopic models where every single entity – household, person or vehicle in transportation context – is modelled and simulated individually. Accordingly, “microscopic” approaches for computing accessibility are attempted. 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.

The remainder is structured as following. In Section II, the context of work and the requirements that yield in designing the application described herein are given. The application’s design and workflow is presented afterwards in Section III. Then, in Section IV, some usage examples are given, focusing on visualization. The report ends with a summary and outlook in Section V.

## II. REQUIREMENTS FROM THE PROJECTS

The described tool was developed in the scope of the project “Urbane Mobilität” (UrMo) [3][4] of the German Aerospace Center (DLR) [5]. The major topic of this project is intermodality – travelling using different modes of transport (e.g. walking, using the public transport, and cycling) along a single journey. Three simulation models are used within this project: the microscopic demand model “Travel Activity PAttern Simulator” (TAPAS) [6][7], the microscopic traffic flow simulation “Simulation of Urban MObility” (SUMO) [8][9], and the location choice model “SimulATING Location Demand and Supply in Urban Agglomerations” (SALSA) [10]. Supporting SALSA with data about a location’s value was the major reason for developing an accessibility computation tool.

SALSA is macroscopic – locations are grouped into areas at the level of “Teilverkehrszellen” (TVZ, English: sub

traffic assignment zone). Berlin's 1223 TVZs are shown in Fig. 1. SALSAs use a large number of different measures for describing different aspects of the locations within a given TVZ. Besides attributes such as the average price, size, or construction year, different accessibility measures can be found among them.

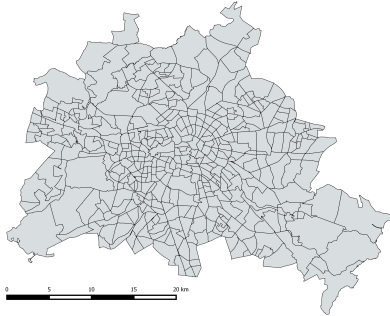


Figure 1. Segmentation of Berlin into "Teilverkehrs-zellen" (TVZ) as used by SALSAs.

The large variety of accessibility measures used by SALSAs is not only a result of using different types of sources and sinks like 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 sinks. Some examples of accessibility measures used by SALSAs are:

- 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 ( $\geq 50000\text{m}^2$ ).

Much of the underlying data used in the UrMo project is available in a disaggregate manner. This includes the positions of dwellings, shops and job offers. In conjunction with an intermodal transport network, it makes sense to use these disaggregated data for computing the needed accessibility measures directly, without any kind of preprocessing and/or aggregation.

### III. APPLICATION IMPLEMENTATION

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. Before describing the different processing steps, it should be noted that within a single call of the application only one transport mode or mode combination is regarded. The application currently supports the modes "walking", "bicycling", "motorized individual traffic", and "public transport", the latter in combination with walking or bicycling. When determining accessibility measures for public transport, a starting time

must be given for choosing according public transport rides. Within the scope of the UrMo project, the peak hour at 8:00 am is used.

Albeit the used methods – mainly routing using the Dijkstra algorithm [11] – are not neither novel or complex, the overall tool is very flexible, realized by some simple features. They include filtering, variable limits, or weighting the sources and will be emphasized in the following subsections.

#### A. Input data

For a regarded region, the application reads the sources' and sinks' positions from a database, as well as a road network. Different disaggregated data are used for sinks and sources, including dwellings, bus stops, job locations, shops, parks, etc. Fig. 2 shows the first two ones 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.

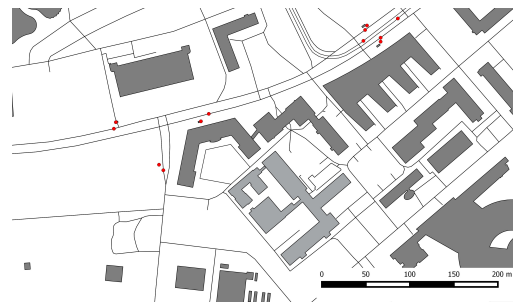


Figure 2. Examples for sources and sinks – dwellings (grey polygons) and bus stop positions (red dots) around the DLR in Berlin (light grey).

When reading sources/sinks 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 a subset only, e.g., only groceries or only shops that are bigger than a given threshold.

Both, sources and sinks can hold a numerical value. For sources, this value is used for weighting the individual source's influence when aggregating. A possible application is weighting dwellings by the number of persons inhabiting them. For sinks, the value's semantics are kept abstract and the values can be summed up for the accessible sinks after computing the accessed space. Usual applications are counting the number of jobs accessible from a location or, as a more abstract measure – determining the selling areas of groceries in a specific range.

Currently, road networks obtained from the OpenStreetMap database (OSM) [12] are used, which were preprocessed and stored into the database. The preprocessing mainly includes a) the determination of intersections by selecting nodes used by more than one way, b) consolidation of access and one-way information for obtaining a unidirectional network with access information for different

transport modes. After these steps, the road network is written into the database. For the city of Berlin, the road network consists of 709713 edges (roads) and 269604 nodes (junctions). The network's edges include the information about the maximum speed allowed for motorized individual traffic and about the allowed modes of transport.

Optionally, the tool additionally reads a public transport network using a database representation of a General Transit Feed Specification (GTFS) [13] data set. The database uses the original GTFS format.

### B. Preprocessing data after reading

In a first step, the objects read from the database to route between are allocated on the road network. For every centroid representing a source/sink, the nearest road is determined that allows the investigated mode of transport. A direct, shortest access to this road is assumed, being a line normal to the road's shape at the point that is nearest to the object. A spatial index, namely the RTree [14] implementation from the Java Spatial Index library [15], 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 bounding box for a given point is not sufficient as the road may be located at the opposite site of the bounding box than the point.

A second issue to solve when allocating objects to roads is to find the road's correct direction. In OSM, as well as in many other road network formats, bidirectional roads are represented by straight lines with no geometrical distinction between both directions. Thereby, both directions have the same distance to the source's/sink'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 halt is located right to. Fig. 3 shows the connections between sources/sinks and the road network.

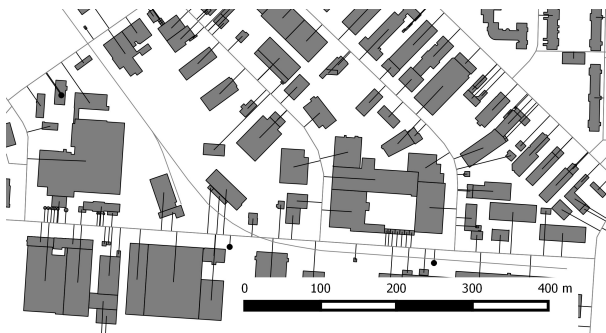


Figure 3. Attaching sources/sinks to the road network; buildings (grey polygons) and public transport stations (black points) are connected to the road network (grey lines) via access paths (black lines).

Optionally read halting positions of public transport have to be assigned to the previously read road graph as well. Again, for each halting position, the nearest road is determined and the halt is mapped onto it. 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, as already done for sources and sinks. To avoid splitting edges at the positions of mapped stations, new edges are introduced that connect the starting and the end node of the edge the station is located at with the station itself. Whether this is more or less performant than splitting edges has not yet been evaluated.

### C. Processing

The process of computing accessibility measures is very straightforward. The application iterates along the read sources. For each, the road network is scanned using the plain Dijkstra algorithm regarding whether the used mode of travel is allowed at a visited edge or not. In dependence to the regarded mode, the travel speed used for determining the travel time to pass an edge is chosen as shown in Table I.

TABLE I. SPEEDS OF THE MODELLED MODES.

Mode	Speed
Walking	5km/h
Bicycling	12km/h
motorized individual traffic	minimum of 200km/h and the road's speed limit
public transport	time schedule (from GTFS)

It should be noted that using the road network as the only factor that limits space accessibility is not correct. Additional travel time delays posed by traffic lights and other traffic participants are neglected when doing so. This issue can be solved by additionally reading travel times or time lines of these. For a given demand, such travel times can be obtained from traffic flow simulations such as SUMO. As well, humans do not walk along edges and cross roads at intersections only. Some approaches for routing across empty – vacant and accessible – space are under development [16] and could be used for extending the application.

When encountering a public transport stop, the available connections to next stations are regarded, using the arrival times as given in GTFS instead of the usually used travel time along the road network's edges.

The process ends as soon as one of the following limits 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 sinks has been visited;
- *maximum variable sum*: stops as soon as the sum of the variable attached to the sinks values' is above the given number;
- *shortest*: stop as soon the first sink is reached.

For each source, the result of this routing consists of the seen edges with attached sinks and their values, as well as the used transport modes or lines. The travel time and the distance are given as well. Of course, a measure used for limiting the accessible space is fixed in the output. The obtained outputs may be aggregated as described in the next section.

D. Aggregation and Output Generation

One of the tool’s major features is the capability to perform different kinds of aggregation. For both, the sources and the sinks, additional aggregation areas (e.g. TVZ) can be read from the database. If given, the measures collected by routing from individual sources are aggregated by averaging them. When being applied to the sinks, the values of all sinks within a given aggregation area that are accessible in the given limits will be joined and averaged. Additionally, an “aggregate all” option is available, which joins all found sinks. Currently, no other statistical characteristics are generated, mainly because the averaging is optional. When needed, such characteristics can be computed from not aggregated results.

As shown in Table II by example, the different aggregation possibilities, in conjunction with flexible limits, allow for the computation of different accessibility measures.

TABLE II. EXAMPLES FOR AGGREGATION IN COMBINATION WITH OTHER OPTIONS (PER TAZ).

Description	Sources agg.	Sinks agg.	Additional parameters
average travel time to other zones	TAZ	all	
travel time to nearest railway station	TAZ		shortest
number of jobs within 30min travel time	TAZ	all	variable=employee; max-tt=30min

IV. EXAMPLE RESULTS AND VISUALISATION

Isochrones are a very simple form of accessibility measures and their visualization. They neglect any kinds of destination locations and regard a single source only. Following [2], isochrones belong to the category of spatial separation measures. Fig. 4 shows isochrones for comparing the space accessible by individual motorized traffic (top) and public transport in combination with walking (bottom).

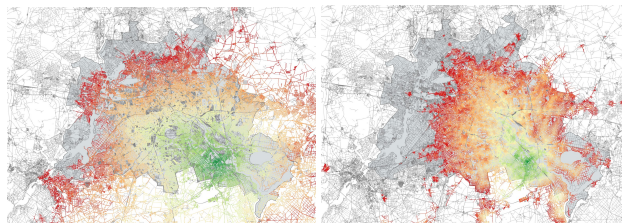


Figure 4. Comparison of isochrones for different transport modes (starting point: DLR center in Berlin); top: motorised individual traffic, bottom: public transport (and walking).

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. Indeed, an aggregation should be performed when visualizing accessibility measures. Fig. 5 demonstrates this by showing the travel times to the next metro or city rail station for every dwelling individually and aggregated into TVZs.

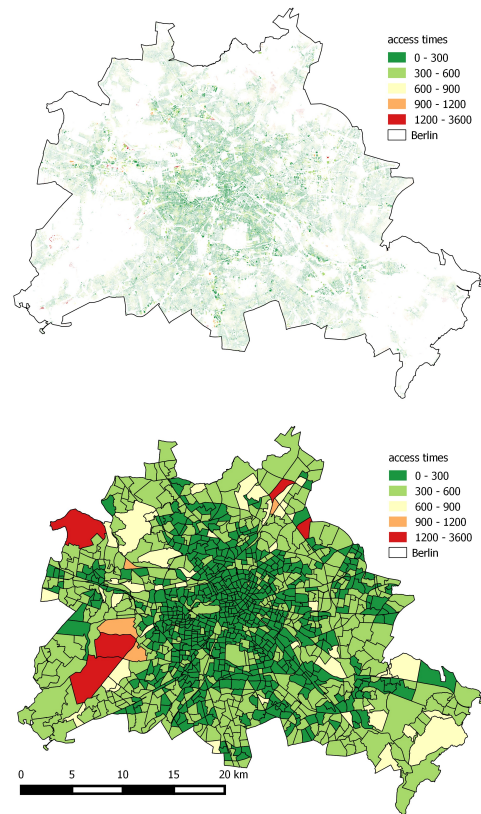


Figure 5. Not aggregated vs. aggregated visualisation of accessibility measures (here: travel time to the next metro or city rail station).

As a final example, the influence of weighting sources is given. Fig. 6 shows the difference between aggregated travel times from dwellings to the respectively next city rail or metro station, once weighted by the households living in each building, once not. As visible, neglecting the households yields in significant deviations.

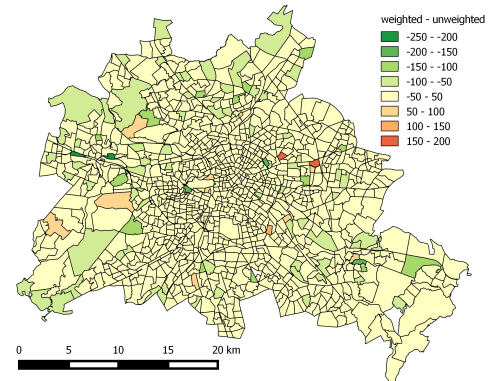


Figure 6. The influence of weighting sources; here, the difference between weighted (by the number of households living in the dwelling) and unweighted access to the next city rail/metro station is shown.

The figures were generated using Quantum GIS (QGIS) [17].

## V. SUMMARY AND OUTLOOK

This report presents a tool for computing accessibility measures as used for benchmarking areas and as used within land use planning models and traffic demand models. The tool follows a fine-grained attempt to compute accessibility measures by routing between individual sources and sinks, mainly dwellings, shops, bus halt 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 put 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 sink was seen. Another one is the possibility to aggregate the individual sources' accessibility values into averages for bigger areas, including a variable weighting of the individual sources. Reading sources and sinks from a database and supporting a pre-filtering when doing so has proved to be valuable as well. Finally, attaching values to the sinks is required for computing some of the needed accessibility measures and was accordingly implemented.

Some microscopic approaches choose subsets of sources and sinks for estimating accessibility. But the experiences with the tool described herein do not prove the necessity for reducing the amount of data to process. In all cases, the complete sets could be processed.

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 sinks via their centroids introduces an error in the paths between the respective source/sink and the road network. Using the nearest position to the road network from a given source's/sink's polygon would be possible. Still, this does not regard the positions of a building's doors or entrances.

The used road networks should be reinvestigated as well. While OpenStreetMap has a usable quality in most parts of Europe and the USA, other parts of the world are only badly covered. Thereby, other digital traffic maps should be investigated. Inclusion of new networks is very simple, because of the low requirements of the tool.

Routing itself should be extended by proper travel times for motorized individual traffic which may be obtained from traffic flow simulations, such as SUMO. As well, approaches for a more realistic person routing could be integrated.

Summarizing, it is surprising how much flexibility can be achieved within a very small and simple application.

## REFERENCES

- [1] 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.
- [2] J. Scheurer, C. Curtis, "Accessibility Measures: Overview and Practical Applications", urbane WORKING PAPER No. 4, Curtin University of Technology, 2007.
- [3] 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.
- [4] DLR, "Urbane Mobilität" project web pages, <http://www.urmo.info/>, 2016, last visited on 23<sup>rd</sup> of April 2016.
- [5] DLR, DLR web site, <http://www.dlr.de/>, last visited on 24<sup>th</sup> of April 2016.
- [6] A. Justen, R. Cyganski, "Decision-making by microscopic demand modeling: a case study", in: Transportation decision making: issues, tools, models and case studies, Venice, 2016, ISBN 9-78-88-96049-06-8.
- [7] DLR, TAPAS web page, [http://www.dlr.de/vf/en/desktopdefault.aspx/tabid-2974/1445\\_read-29381/](http://www.dlr.de/vf/en/desktopdefault.aspx/tabid-2974/1445_read-29381/), 2016, last visited on 23<sup>rd</sup> of April 2016.
- [8] D. Krajewicz, 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.
- [9] DLR, SUMO web pages, <http://sumo.dlr.de/>, 2016, last visited on 23<sup>rd</sup> of April 2016.
- [10] B. Heldt, K. Gade, and D. Heinrichsm "Challenges of Data Requirements for Modelling Residential Location Choice: the Case of Berlin, Germany", European Transport Conference 2014, 2014.
- [11] E. W. Dijkstra, "A note on two problems in connexion with graphs", In: Numerische Mathematik 1, pp. 269-271. 1959, doi:10.1007/BF01386390.
- [12] OpenStreetMap contributors, OpenStreetMap project pages, <http://www.openstreetmap.org/>, 2016, last visited on 23<sup>rd</sup> of April 2016.
- [13] Google, General Transit Feed Specification pages, <https://developers.google.com/transit/gtfs/>, 2016, last visited on 25<sup>th</sup> of April 2016.
- [14] 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
- [15] JSI contributors, JSI (Java Spatial Index) RTree Library web pages, <http://jsi.sourceforge.net/>, 2016, last visited on 25<sup>th</sup> of April 2016.
- [16] S. Andreev, J. Dibbelt, M. Nöllenburg, T. Pajor, and D. Wagner, "Towards Realistic Pedestrian Route Planning", In: Proceedings of the 15th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'15), volume 48 of OpenAccess Series in Informatics (OASICs), pages 1-15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, September 2015.
- [17] QGIS Development Team, QGIS Geographic Information System, Open Source Geospatial Foundation Project. <http://qgis.osgeo.org>, 2016, last visited on 29<sup>th</sup> of April 2016.