Shadow as Route Quality Parameter in a Pedestrian-Tailored Mobile Application

Cristina Olaverri-Monreal *Member, IEEE*, Matthias Pichler, Gerd Ch. Krizek and Sebastian Naumann

***Abstract*—Most of the applications for pedestrian navigation are based on information that is useful for car navigation systems, while the basic needs of pedestrians are seldom considered. We contribute to the current research in the field by proposing a pedestrian-tailored navigation application based on path attributes, that as of yet have not been considered in a pedestrian route calculation. We focus in particular on shaded paths, which are considered preferable during hot, sunny days, especially by pedestrians of advanced age, those with small children or persons that suffer from sensitivity to light. Evaluation results regarding the performance of the application provided a significantly better acceptance rate by the end users in terms of light intensity, amount of shade and positive affect that was derived from walking through the suggested routes.**

***Index Terms*—Pedestrian, Comfort, User Experience, Shadow**

I. Introduction

Available maps often provide rather general data, based on information that is useful for automotive navigation systems but not tailored to the actual needs of the pedestrian. Some companies (i.e. Walkonomics [1]) are starting to consider pedestrian needs, for example rating the walkability of streets, but still most commercial navigation tools fail to encompass a comprehensive organization of route quality information in the user interface. Because the basic demands of pedestrians are seldom considered, there is still great potential for improvement in order to provide them with appropriate routing and navigation options for an optimal user experience. According to [2], [3], the basic characteristics associated with pedestrian patterns are the following:

1) A pedestrian needs very little room (0.5 m^2) when moving along a given route compared to the various motorized and non-motorized means of transport.
2) Walking speed is rather slow compared to other modes of transport.
3) Pedestrians are vulnerable road users.
4) Walking is a mode of locomotion, like any other, to get from an origin to a destination. However, it also evokes feelings that are essential for health and well-being, such as relaxation and joy.

Manuscript received June 14, 2016.

C. Olaverri Monreal was with the AIT Austrian Institute of Technology, Business Unit Technology Experience, Innovation Systems Department, Vienna Austria. She is now with the University of Applied Sciences Technikum Wien, Department of Information Engineering and Security, Vienna Austria. Email: olaverri@technikum-wien.at

M. Pichler was with the University of Applied Sciences Technikum Wien, Vienna Austria. E-mail: matthias-pichler@aon.at

G. Krizek is with the University of Applied Sciences Technikum Wien, Vienna Austria. E-mail: krizek@technikum-wien.at

S. Naumann is with the Institute for Automation and Communication (IFAK), Magdeburg. Email: sebastian.naumann@ifak.eu

Pedestrians are especially vulnerable as road users. Because they are not protected by an armored vehicle, nor do they wear any protective artifacts (i.e. helmet), they are exposed to different weather conditions such as rain and sun. The relaxing quality associated with walking influences path selection based on specific patterns that guarantee a pleasurable experience, such as a beautiful path in an unfamiliar or aesthetically appealing environment. In line with this, this paper focuses on path attributes that have so far not been considered in a route calculation tailored to pedestrian needs.

The remainder of this paper is organized accordingly: The following section presents related work in the areas of pedestrian navigation and routing approaches. Section III presents the system architecture of the proposed application. Section IV describes the routing algorithm. Section V describes the data storage process and presents the final user interface. Section VI describes the assessment of the proposed system. Finally, Sections VII and VIII discuss, summarize and conclude the paper.

II. Related Work

Pedestrian routes are characterized by so called walkability indicators which may affect the selection of routes. A comprehensive overview is provided by [4] in which more than 150 indicators have been compiled from previous works such as [5], [6], [7], [8] and subsequently classified in groups. The factors that have an effect on the selection of routes include, for example, aspects that might connote a positive feeling such as the availability of stores, shopping centers, parks, public spaces, etc., and also aspects that might affect the route negatively such as the existence of graffiti, vandalism and dereliction. There is no unanimous opinion on how to group all of these walkability indicators, but in some works [4] has been suggested that certain dimensions are established such as connectivity, comfort, coexistence, convenience, etc. to subsequently generate for each dimension subgroups that contain walkability indicators such as wind [6] or "thermal comfort perception". According to [9] in warm to hot climate conditions, "shade" can determine the willingness to walk (i.e. for shopping). In line with these indicators, we consider in this work pedestrian paths that provide shadow. This subject has been investigated in previous work [5] with weather condition factors such as rain. However, the research was just limited to tree canopy factors and did not consider shadow casted by buildings.

In recent years, routing approaches for pedestrians have been developed using different digital maps. For example, open source approaches like OpenStreetMap [10] or the Graph

 DOI: 10.1109/MITS.2016.2605358

Integration Platform [11] in Austria, include information tailored for pedestrians.

The A* algorithm [12], firstly described in 1968, is a frequently encountered shortest-path algorithm for road networks that is an extension of Dijkstra's algorithm [13]. A* achieves a better performance by using heuristics to guide its search. Compared to the Dijkstra approach, where no further information about the routing graph is used apart from the weights of the edges, A* constantly estimates the costs from the currently investigated node to the destination node in order to determine which node to consider next. This estimation is possible because the coordinates of the nodes in the graph are known. The A* search is more targeted than in the Dijkstra algorithm and therefore achieves a better performance. However, there is no general agreement among experts concerning the performance of A*. On the one hand, [14] reports a better performance when the graph is preprocessed and a more sophisticated path-planning method is used for practical travel-planning systems, but on the other hand [15] has found A* to be superior to other approaches.

In our work the graph network is static and the selected test site (16th district of Vienna) is with 8,7 km^2 relatively small. As a consequence, time performance was not a major criteria, and we selected the Dijkstra algorithm to calculate the shortest path. However, if the route is longer and requires more travel time, the weights of the edges are subject to greater variation due to the continuous change of the sun's position. In cases where the edge conditions are variable, the D* algorithm [16] is able to re-route the path faster than planning from the start since it modifies its previous search results locally. A further candidate for variable edge conditions might be the D* lite algorithm [17] which is based on the Lifelong Planning A* algorithm [18] which in turn is based on A*. D* lite implements the same behavior as D* in a simpler approach and in fewer lines of code.

Existing pedestrian navigation systems with different features and gadgets have been summarized in related literature. Most of the approaches relied on weighting scores, similar to the works published in [19], [20]. In [21] approaches have been presented for users with special needs, such as visually impaired persons, or applications that do not rely on GPS-based systems but rather on other technologies such as Radio Frequency Identification (RFID) or data transfer via Bluetooth to mobile devices.

Furthermore, other works have highlighted different aspects of the selected routes: The authors in [19] developed the NaviComf framework to build navigation systems that take advantage of information collected through multi-modal sensors. Particular importance was given to ambient conditions such as temperature. The system stored and predicted values related to environmental conditions to classify edges as unattractive or attractive for pedestrian use.

A further speech-based application provided pedestrians guidance in an unfamiliar environment [22]. The system focused on audio guidance, but some landmarks were also provided through pictures as reference points. The combination of landmarks, street names and audio instructions was classified as positive. However, additional studies provided conclusions

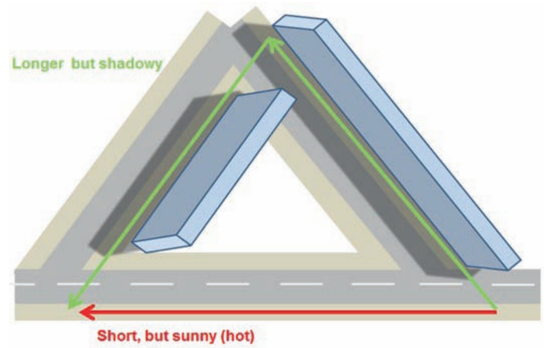

Fig. 1. Paths including elements that provide shade

confirming that visual approaches with digital maps produced the best results [23].

Additional literature has acknowledged the fact that most of the existing pedestrian navigation approaches are adapted car navigation systems. Therefore, the focus of several works that rely on studies of route quality (including route safety, comfort, attractiveness and accessibility) lies on the manner in which instructions should be provided to pedestrian users. This is achieved by studying user requirements for an enhanced tailored pedestrian navigation system that considers route quality parameters [24].

Some classifications of users of pedestrian navigation tools according to diverse route quality parameters have been performed in previous work (i.e. [25]). However, a comprehensive categorization that aligns user preferences to different route quality parameters still needs to be implemented.

In this context, we contribute to the current research in the field by developing an application for pedestrian navigation specifying path attributes that have so far not been considered in other approaches. Explicitly, we target paths that include elements that provide shade, as they are considered particularly pleasant in hot and sunny summer days and usually preferred by pedestrians of advanced age, those with small children, or persons that suffer from sensitivity to light. Figure 1 illustrates the idea. A final evaluation regarding the performance of the application is performed.

III. System Architecture

As the sensor technology available in smart devices enables the monitoring of mobility patterns [26], we developed a mobile application for pedestrian routing and navigation based on the openly licensed world map OpenStreetMap (OSM) data [10], running on any devices with the Android operative system. The implemented system architecture is illustrated in Figure 2 and consists of the following components:

- a Smart Device Application, which contains a Router and a Display Unit,
- a Data Storage Component, which contains the map data
- and a Converter Component to make the raw data from OSM routable.

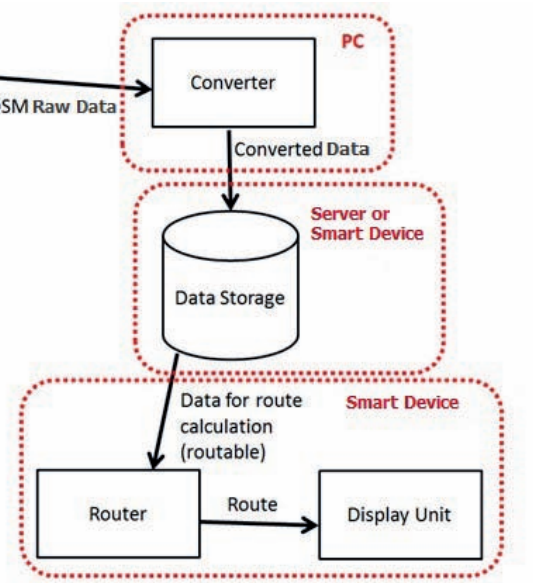

Fig. 2. System architecture of the proposed mobile application for pedestrian navigation

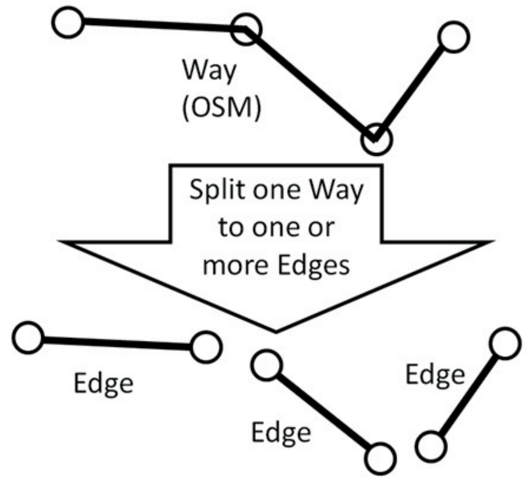

Fig. 3. Conversion process to obtain edges from a specific way

A. OSM Data Processing

In order to implement the navigation tool, we relied on the data provided by OpenStreetMap. The selection was based on the ease of access to the raw data compared to other map providers that are also available (i.e. Google Maps, Bing Maps, Graph Integration Platform (GIP)). OpenStreetMap raw data can be easily downloaded as an .osm file. As .osm files are formatted in the Extensible Markup Language (XML), no special Geographic Information Systems (GIS) software is necessary for reading the file. Additionally, there are editors available to display these raw data as a 2D map free (i.e. JOSM).

Other map providers such as GIP might supply more accurate pedestrian data, but our work creates the basis for a pedestrian-tailored routing and navigation system that is currently unavailable. Within this work, pedestrian needs are addressed as part of a framework that will include routing graphs composed of sidewalks using the tagging information contained in the "sidewalk=*" key to indicate the presence or absence of a sidewalk and in the attribute values left, right and both.

In order to use OSM data for navigation purposes we first made it "routable", converting it to graphs consisting of nodes and weighted edges. This structure is the basic requirement for routing algorithms like Dijkstra or others similar. A way consists of an array of nodes (that determine the shape of the street, footway, etc.) and of tags, which contain attributes like name, maximum speed, illumination information, sidewalk (seldom available), etc..

For efficient data conversion, we first analyzed the raw data structure and determined the number of nodes, ways, relations and what they represented, as well as any other tags that provided additional information about the object. The information provided by the k="highway"-attribute, which characterizes the type of the way (footway, residential, etc.), was especially relevant for our application.

Nodes and ways are sufficient to create a routable graph, which consists of nodes and edges. The nodes are directly transferred into the graph and additional attributes can be subsequently added (i.e. edge degree). The edge degree gives information about how many edges are connected to the nodes.

During the conversion process, nodes which are not necessary for routing are not transferred. This applies to nodes that do not make routing decisions possible, such as nodes with an edge degree of two. An edge only consists of one end node and one start node; there are no nodes in between. Therefore, to make the OSM data usable for routing algorithms like Djikstra, we split all the ways into several edges, resulting in a whole routable graph. Figure 3 depicts the conversion process from a way to edges.

B. Automatic Conversion Tool

In order to automatically convert the OSM data into a routable graph, we studied several open source converters that in some cases also included a routing engine. Most of the tools converted the .osm raw data file into a POSTGIS database SQL file that also incorporated attributes. For the sake of simplicity and in order to be aware of all the attributes that characterized our graphs, we decided to determine the attributes to be added to the routable graph ourselves rather than use converted data that was not always traceable. On account of this, we developed our own automatic conversion tool that used the attributes that were relevant for our approach and had been previously filtered by the OSMFILTER [27] command shell tool. The converter component is illustrated in Figure 2. It was programmed in Java and performed the following tasks:

- Reading in the .osm input file (via XML reader)
- Creating three classes: Nodes, Ways and Edges
- Creating Lists (Arraylists) of these classes
- Storing the Ways and Nodes from the .osm file in the proper classes
- Converting the Ways and storing them as Edges (class Edges)

- Creating CSV output files for Nodes and Edges

The automatic conversion tool converts the OSM data into a routable graph and stores the graph on an external MYSQL database. The smartphone application then uses the converted data from the MySQL database to calculate the routes considering shade. Integrating the converter into the smartphone application would make the conversion rather CPU intensive. As this conversion only needs to be done once, there is no advantage when doing it on the smartphone itself.

C. Processing of Additional Data

Concerning the development of the routing algorithms we characterized paths through a weighting system. To this extent, we assigned high values to way characteristics that matched the attributes selected for our approach, namely routes with shade.

In the same manner, features that indicated that a specific path was not ideal (e.g. sunny path) scored the path with a lower value, determining in this process the quality of a specific way. The weighting system was implemented in our converter, so that the algorithm selected the most appropriate route according to the most favorable conditions for the pedestrian. In order to be able to perform the weighting we performed the following calculations:

1) Edge Length: The edges were first classified according to the length of the edge. For the calculation the distance between circles of longitude and circles of latitude was required. This distance is estimated with 111.3 km at the equator [28]. In the case of Vienna, where we developed and tested the application, the distance of the circles of latitude was the same, but the distance between the circles of longitude needed to be multiplied by cos (lat) due to the spherical earth shape. Figure 4 shows the calculation of the edge length via Pythagoras.

2) Cardinal Direction of the Edge: To calculate the routes with shade it was required to know the cardinal direction of the edges. OpenStreetMap raw data does not contain any cardinal direction information. However, as depicted in Figure 4, X and Y could be computed via the longitude and latitude coordinates of the nodes. The tangent and arctangent functions respectively made it possible to determine the cardinal direction and the angle of the route. For example, 0° describes the cardinal direction East-West. If the angle is located between -22.5° and 22.5° the East-West cardinal direction is added to the edge. If the angle is located between 22.5° and 67.5° the cardinal direction will be Northeast-Southwest.

3) Shadows: In order to be able to determine if a route provided shade, it was necessary to know if there were objects, particularly buildings or trees, which dispensed shadow to the sidewalk or to the footway. To this end, we combined the information from the cardinal direction of the sidewalk and the current Sun position (dependent on the day time) and determined if a building was located to the right or to the left of a sidewalk. Data related to the building's location and shape is provided by OSM. Data about the height is not available, but in this work we assume that every building in Vienna is high enough to cast an adequate shadow. For example, in the area where the evaluation took place the height of buildings ranges from 12m to 16m [29].

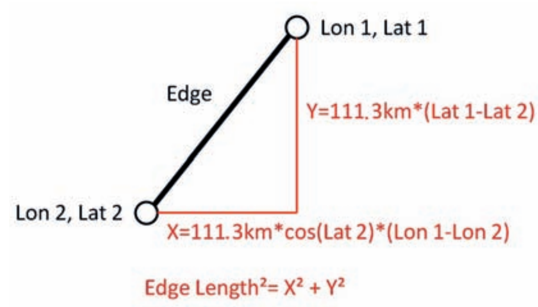

Fig. 4. Calculation of the Edge length via the longitude and latitude coordinates of the nodes

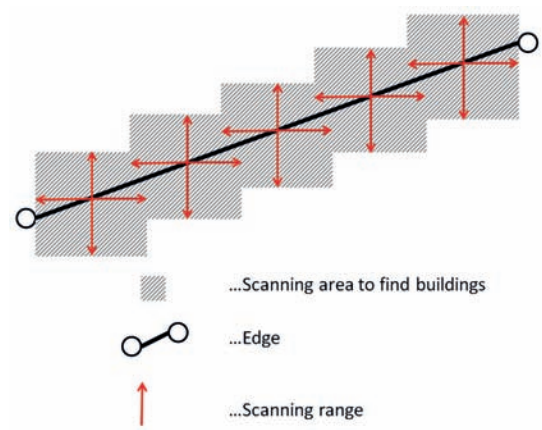

Fig. 5. Scanning process to determine the location of buildings

In order to detect a building we determined if it was located next to an edge. In OSM data, a building consists of polylines. There are not attributes providing information about whether a building is next to a street or not. Therefore, we first scanned the area on both sides of the edge to detect a building, testing several scanning area values. We selected the value of 20m horizontally and vertically as it provided the best results. Figure 5 shows the principle of the scanning process to find buildings next to an edge. We characterized the scanning area as a square for reasons related to the quality of the results.

To filter the data related to buildings we ran the OSM-FILTER, which also obtains points of interests, shops and restaurants that could later be used as landmarks. After this filtering process, we started the scanning process, which relies on the converter tool from Section III B.

Our application also considered trees as objects that provided shade. Therefore, we used "numbertrees" as edge-attribute containing a numeric value of the number of trees along an edge and "costs" containing a value for the edge length. We increased the edge costs if no trees or only a few of them were present, taking into account the length of the edge. As a consequence, the routing algorithm calculated a comfortable route for the user avoiding sections where trees were not existent. For example, as denoted by the following formula:

```
tree density=(numbertrees)/(cost [m])
```

if the tree density was 1, then the average distance between two trees was one meter. The edge costs were only increased if the tree density was smaller than 0.1 (average tree distance bigger than 10 m.)

IV. Route Calculation

We specify in this section the requirements that the router component from the system architecture needed to fulfill, in order to calculate the optimal path with shade from the selected origin A to the given destination B. The route is automatically recalculated when the shadow situation significantly changes (i.e. the timespan between planning and starting is significantly higher than 1 hour) or when the user has left the recommended route.

In order to compute the shortest path from a start node to an end node within the given weighted and routable graph described in section III B, we used the Dijkstra routing algorithm [13]. We disregarded other variants of the algorithm that provided a better time performance, such as the A* algorithm, because the selected test site was rather small and time performance was not a major criteria of the implemented application. In order to determine if a route had shade or not, we considered the following parameters:

- Current time. Due to the fact that the Sun's position is dependent on the time of day. To this end, we used the time of the system's smart device.
- Sun position. The position of the Sun dependent on the time (morning sun in the East, at noon in the South and in the evening in the West).
- Cardinal direction of the edge, as described in the system architecture section.
- Presence of buildings. This feature is especially relevant if the direction of the Sun is lateral to the edge. If there are no buildings providing shade, then the edge receives direct sunlight, hereafter referred to as a "sunny edge". Two different situations can produce sunny edges: the Sun's rays come from the same cardinal direction to which the edge is oriented, or the Sun shines laterally to the edge and there are no objects that block the sunlight and cast a shadow onto the edge.

A. *Direction of the Sun is Equal to the Cardinal Direction of the Edge*

If the cardinal direction of the edge is equal to the current Sun position, then this edge is sunny and has to be avoided. To avoid sunny edges on the route, we increased the weight or cost of sunny edges in our algorithm, thus making them unattractive for the routing algorithm.

To this end, we implemented the code below where the shadow factor is multiplied by the edge costs. The factor is initially 1. If the concerning edge has a specific cardinal direction and if the current time is daytime when the Sun comes from that specific cardinal direction, then this edge is assessed as sunny and the costs are multiplied by 10. Otherwise, the factor is still one and the costs do not change.

For testing purposes we set the value that determines the grade of shade if the Sun is shining directly into the street (+-5 degree) to 10. This value can be calibrated depending on the angle between sun and street. If the angle between sun and street is between 5 and 22.5 degrees, we set the factor to only 5 arguing that if the Sun is directly shining into the street, the probability of a higher temperature increases.

We first calculated the solar azimuth angle, which represents the direction of the Sun. Then this angle was compared with the cardinal direction of the edge to determine if an edge was sunny or not. To calculate the solar azimuth angle we relied on the computations presented in [30] and [31] considering the day of the year, the time and the position on the earth. We then calculated the equation of time that represents the difference between the mean solar time (12 o'clock according to a precise watch) and the apparent solar time (moment when the Sun is exactly in the south direction). Afterwards, we calculated the declination of the Sun, which gives information about the location of the zenith of the Sun. For example, in the summer months the zenith is located on the northern hemisphere.

Further parameters that needed to be computed were the hour and altitude angle of the Sun, which give information about the height of the Sun. The code in the "SunCardinal" procedure determines the parameters to compute the solar azimuth angle. Day, hour and minute in Vienna, are calculated using summer time and thus the hour value is reduced by 1 hour. The position is then determined by selecting the coordinates and performing the necessary calculations (i.e. equation of time, etc.).

The factor we selected to determine the grade of shade was initialized with the value 1, and the difference between the edge cardinal direction and the current solar azimuth angle was calculated. If the difference was smaller than 5 degrees, then the edge was assessed as sunny and the factor was set to 10. If the difference was smaller than 22.5 degree but bigger than 5 degrees, then the edge was assessed as half sunny and the factor was set to 5.

B. *Direction of the Sun is Perpendicular to the Edge*

In this case the Sun shines from a lateral direction, more or less perpendicular to the edge. As described above, if there is no building along the edge that provides shade, the edge is sunny. The code in the "SunSouth" procedure shows the case where the Sun shines from the South (i.e. midday). The other cases can be resolved similarly. The code determines whether there is a building on the sunny side. If there is not any building, then the edge is assessed as sunny, and the edge costs are multiplied by 10. The principle with the increase of the edge costs is the same as in the case where the Sun's position is equal to the cardinal direction of the edge.

C. *Additional Factors: Weather and Night Information*

The weather information is an important factor to consider when calculating routes with shade. A cloud covering the Sun will make it unnecessary to take into account buildings and trees that provide shade. As a consequence, a further parameter to calculate the routes is the weather forecast.

```
procedure SunCardinal
    calendar ← (GregorianCal) GregorianCal.getInstance()
    date ← newDate()
    calendar.setTime(date)
    hour ← calendar.get(Calendar.HOUR)
    min ← calendar.get(Calendar.MINUTE)
    day ← calendar.get(Calendar.DAY_OF_YEAR)
    if calendar.get(Calendar.AM_PM) == Calendar.PM then
        hour ← hour + 12
    hour ← hour − 1 ▷ taking into account summer time
    lat ← 48.308 ▷ position in Vienna
    lon ← 14.207
    constant ← Math.PI/180
    declination ← −23.45 ∗ Math.cos(constant ∗ 360 ∗ (day + 10)/365) ▷ declination of the Sun, dependent on the time of the year. Solar azimuth calculations in [31]
    timeequation ← 60 ∗ (−0.171 ∗ Math.sin(0.0337 ∗ day+0.465)−0.1299∗Math.sin(0.01787∗day−0.168)) ▷ equation of time, necessary because of the difference between the time when the Sun is exactly in the South and midday (12:00)
    angle_hour ← 15 ∗ (hour + min/60 − (15.0 − lon)/15.0 − 12 + timeequation/60)
    x ← Math.sin(constant∗lat)∗Math.sin(constant∗declination) + Math.cos(constant ∗ lat) ∗ Math.cos(constant∗declination)∗Math.cos(constant∗angle_hour)
    hoehe ← Math.asin(x)/constant
    y ← −(Math.sin(constant ∗ lat) ∗ x − Math.sin(constant∗declination))/(Math.cos(constant∗lat) ∗ Math.sin(Math.acos(Math.sin(constant ∗ hoehe))))
    azimut ← 0.0
    if hour + min/60 <= 12 + (15 − lon)/15 − timeequation/60 then
        azimut ← Math.acos(y)/constant
    else
        azimut ← 360 − Math.acos(y)/constant
    shadow_factor ← 1
    indicator1 ← Math.abs(edge.direction_numval − azimut)
    indicator2 ← Math.abs(edge.direction_numval + 180 − azimut)
    if indicator1 < 5 or indicator2 < 5 then ▷ azimut==edgedirection +5 to -5 then sunny
        shadow_factor ← 10
    if indicator1 >= 5 and indicator1 < 22.5 or indicator2 >= 5 and indicator2 < 22.5 then ▷ half sunny
        shadow_factor ← 5
```

```
procedure SunSouth
    if azimut > 157.5 and azimut <= 202.5 then ▷ Sun comes from the South
        if edge.direction == East − West
          or edge.direction == Northeast − Southwest or
          edge.direction == Southeast − Northwest then
            if buildingonsunnyside == "no" then ▷ if there is no building on the side of the edge,
          increase the edge cost by factor 10
                shadow_factor ← 10
```

During overcast conditions, there is no need to calculate a shaded route. However, if the sky is partially cloudy with intervals of Sun and clouds, a more accurate weather model is required to determine if there will be sun on the route while the pedestrian is walking. We considered such restrictions in our application and made it extendable to consider such a model in the form of XML, JSON or HTML format, as those are the usual formats in which such data is provided (i.e. wetter.com, openweather [32], [33]). The application reads these XML or JSON formatted data and if the weather information says that a cloud will cover the Sun along the route, then the corresponding calculations (solar azimuth angle, shadow checking, etc.) are skipped and have no relevance for the final route. Our algorithm reflected the night time according to the time for sunrise and sunset in Austria on the 21st of June (longest day of the year), disabling the option for selecting routes with shade depending on the time when it was dark.

V. Data Storage and End User Application

A. Data Storage Process

The data storage component from our system architecture contained the information related to the map that was stored in a server database. The data were exchanged between the client application and server via 3G using Universal Mobile Telecommunications Service (UMTS) technology.

Figure 6 shows the database model. The red keys represent primary keys, the green keys represent foreign keys. NN denotes Not Null. The establishment of a connection and data exchange between an Android application and a MySQL database occurred through a PHP-script on the server side, which had the function of a link between the Android application and the MySQL database. After the Android application called the PHP script on the webserver via HTTP, the PHP script established the database connection, and then executed the SQL query and presented the data.

B. End User Application

The Display Unit component in the system architecture is in charge of displaying to the user the calculated route. To this end, we relied on Google Static Maps API [34]. With this approach, the parameters of the route (longitude and latitude pairs) are sent to Google via URL. The map with the

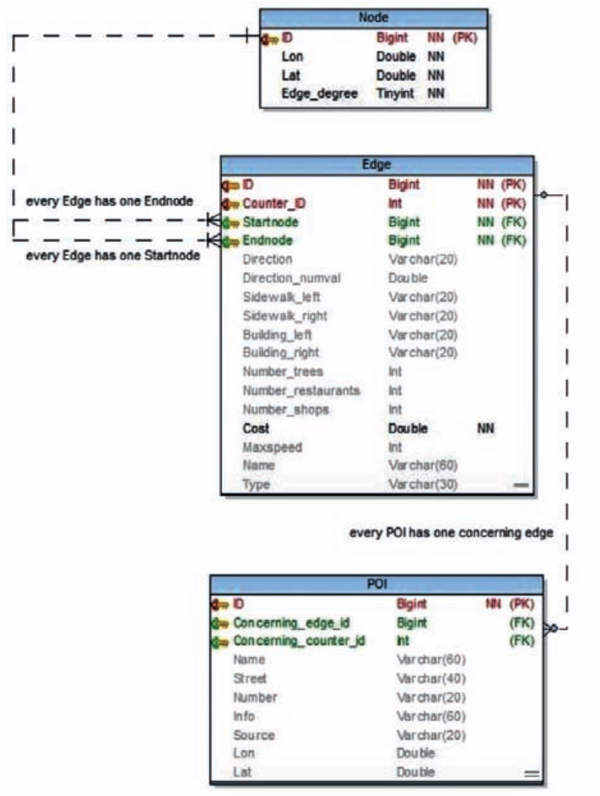

Fig. 6. Database model for the pedestrian navigation application

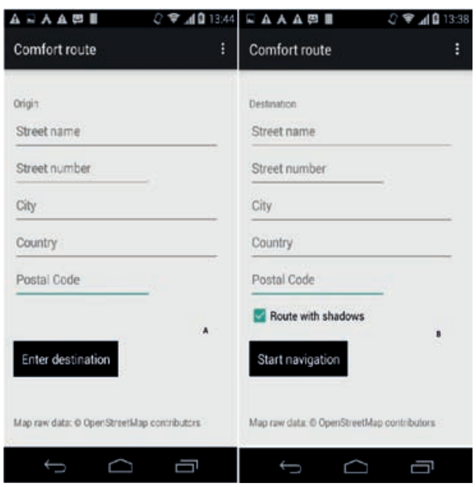

Fig. 7. User interface for the pedestrian navigation application. A depicts the interface to enter the origin and B to enter the destination

route is created via a HTTP-Request. Figure 7 shows the user interface for our pedestrian navigation application, with the menu options to enter origin and destination and the "Shaded Route" option.

VI. Evaluation of the Application

A. Experimental Setup

To evaluate the performance of our application and determine if the suggested route provided significantly more shade than a route where the shade option was not selected, we defined an evaluation path for a field test in a real world scenario. We then measured the light intensity per unit area using a Lux Meter and compared baseline values without having the shadows option selected, with values resulting from shade conditions.

To this end, we selected the 16th district of Vienna as testing area for our evaluation. This urban area is characterized by traffic-calmed side streets with and without trees and main streets with shops and restaurants. The testing time was 9.30am (summer time) in July, 2015 for both testing routes. The device used for the measurement was a Lux Meter PCE-174. Figure 8 shows the selected routes located in an open street map, without the shade option (A, Length: 765m) and with the shade option (B, 1020m) selected. A clear difference in the displayed routes can be identified.

- The shadow projected by the person using the Lux meter was not allowed to interfere with the shadows on the path.
- The light was to be measured in the center of the path at a constant walking speed.
- The sonde of the Lux meter was held approximately 1.2m over the floor in a horizontal position.

Additionally, we performed a summative evaluation with 5 test persons to investigate the positive affect or experience of feeling or emotion [35] that was derived from walking through routes that provided shade, and to assess the users satisfaction with the thermal environment. To measure the users perceived thermal conditions, we developed a 5-point Likert-scale questionnaire, ranging from the worst value 1 to the best value 5. To this end, we used items that covered questions related to how the probands felt in the shade or in the sun regarding the temperature and the amount of light as well as two additional items regarding the usefulness of the app to provide shaded paths. Additional questions related to demographic information and a field to enter comments completed the survey. To collect the data, the subjects followed the routes suggested by the application, once after having selected the option to provide routes with shade, and once without having selected the option.

To get realistic results we performed the tests on a clear and very hot day with temperatures over 35°C. The sample consisted of 5 probands (2 females, 3 males), with an average age of 34.6, SD = 14.24. We alternated the order of the paths (with or without shade) for each subject in order to avoid bias that could affect the results by any eventual advantage caused by the order. Each participant walked alone. The difference between the starting time was of 30 seconds for each participant. Figure 9 illustrates the user experience evaluation. We then analyzed the responses to measure the extent of the person's agreement with the set of questions from our survey. A non parametric Chi square test (level of α = .05) was conducted to test statistical significant results regarding the dependence of positive feelings on walking through shaded or sunny routes.

B. Results

1) Light Intensity: Figure 10 shows the values corresponding to the total light in paths without (A) and with shade (B).

Fig. 8. Testing route for the light intensity measurement considering paths without shade (A) and shaded paths (B)

As the paths differ in longitude, and the accumulated amount of total light depends on these length values, the amount of light (sun) for both paths is shown as the generalization of illumination values along sunny sections on both areas A and B as illustrated in C. The shaded route has nearly 55% less light than the route suggested without the option activated.

TABLE I
QUALITY ROAD VALUES WITH AND WITHOUT SHADE IN A RANGE FROM THE WORST VALUE 1 TO THE BEST VALUE 5.

Item	Path with shade		Path without shade		χ (α = 0.05, df = 4)
	Mean	*SD*	*Mean*	*SD*	*p*
Freshness *	4.4	0.49	1.8	0.75	0.04
Light	4.4	0.80	2.6	1.6	0.28
Usefulness *	4.8	0.40	1	0	2.46E-09

2) Users Perceived Thermal Conditions: Table I shows the results regarding comfort provided by shaded routes. The values ranged from 1 (worst) to 5 (best). Results indicated that the temperature and amount of light on the shaded route were perceived as much more comfortable than on the sunny route.

Fig. 9. Person evaluating the app in a path without shade (A) and with shade (B).

In the sunny path, the Sun was shining on the backs of the probands, therefore it did not affect the vision of the participants. Nevertheless the majority of the participants perceived the sunlight as disruptive. Clear differences concerning the usefulness of the app regarding the integration of additional attributes to provide shaded routes were apparent.

VII. DISCUSSION

A. Shadow Location Accuracy

Our system is reliable in the context of performing its required function: showing a path with shade to pedestrians in a digital map. In order to determine the exact shape and location of a shadow cast by a building, values related to the building's height and a 3D model of the building, including the type of the roof (e.g. flat or gabled), are required. OSM provides tags to define data regarding building shapes and their location, and this information forms the base required to generate the 3D city model elements through procedural modeling. However, data related to the height of buildings is not provided by OSM and this information needs to be accessed from other databases if accuracy regarding the shape and size of shaded areas is required.

Assuming the height of the buildings has been made available to OSM, it is possible to determine if a sidewalk is shaded or not by using this information along with the parameters mentioned in Section IV (time, sun position, cardinal direction of the edge and presence of buildings) to determine the size

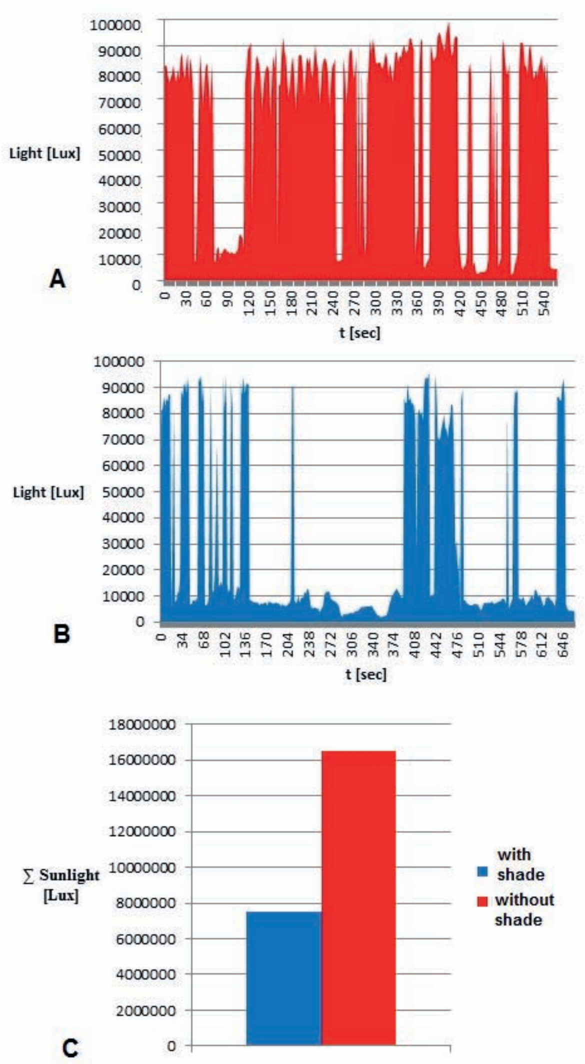

Fig. 10. Total amount of light in paths without (A) and with shade (B). The amount of light for both paths is shown in C

of the shaded area created and compare it with the width of the road and sidewalk.

B. Smart Device Localization

In order to appropriately localize pedestrians, inertial measuring units are used to estimate the movement of a person, by detecting steps, estimating stride lengths and the directions of motion [36], [37]. However, precise localization information on the starting point is necessary. A problem of consumer smart devices is that built-in GNSS (Global Navigation Satellite System) receiver modules do not allow raw data access such that correction techniques could be applied for more precise positioning. Therefore, an additional device is needed which has to be coupled with the user's smart device via Bluetooth for example. The additional device mainly consists of a GNSS receiver module which provides the GNSS raw data. Available modules are offered by the u-blox NEO-7 series [38] for example. The open source library RTKLIB (Real Time Kinematic) [39] provides a set of methods for correcting the GNSS raw data for more precise localization - among others PPP (Precise Point Positioning) and DGNSS (Differential GNSS) as described below. Some more expensive GNSS modules, like the NEO-7P module [40], already come with PPP support such that the RTKLIB is not necessary. Fundamental methods for more accurate positioning via GNSS are:

1) Differential GNSS: Differential GNSS (DGNSS) [41] is a technique that significantly improves both the accuracy and the integrity of the GNSS. It requires high quality GNSS reference receivers at known, surveyed locations. The reference station estimates the slowly varying error components of each satellite range measurement and forms a correction for each GNSS satellite in view. This correction is broadcast to all DGNSS users on a convenient communication link. Typical ranges for a local area DGNSS station are up to 150 km. DGNSS improves location accuracy to less than 1 m. Different vendors such as Axio-Net [42] or SAPOS [43] operate commercial reference station networks and provide tailored correction data. However, ongoing costs for the use of these correction data services as well as the costs for transmitting the correction data via mobile radio are unfavorable to the user.

2) Precise Point Positioning: Compared to the DGNSS method where differential correction data with a reference station are needed, the Precise Point Positioning (PPP) technique [44] is mainly based on precise satellite orbits and clock information. This information is provided free of charge by the International GNSS Service (IGS) and the various Analytical Centers (ACs) of the IGS. By using the precise satellite orbits and clock data together with the raw data of the GNSS receiver module, a positioning accuracy on centimeter level is possible.

As stated above, both correction techniques DGNSS and PPP are not usable in combination with consumer smart devices today due to the impossibility of accessing the raw data of the GNSS receiver module. We hope this will change in the future such that there will be no additional device needed anymore.

VIII. Conclusion and Future Work

In this paper, we have proposed a mobile application for pedestrian navigation that considers needs that are particularly relevant during hot, sunny days. The resulting pedestrian-tailored approach can be used by a variety of persons, but will be especially useful for those that suffer from an increased sensitivity to light, or certain severe disorders. Therefore, the implemented tool would be particularly useful for improving existing systems as it helps create feelings relevant to health and well-being.

The application assessment showed statistically significant differences between routes where the "with shade" option had been activated and between routes in which the option was not activated. Particularly, positive feelings derived from the light intensity, amount of shade and, generally speaking, from walking through the suggested routes could be established.

In the evaluation phase of the application we observed that routes that provided shade were in average longer than routes for which the shade option was not activated. Accordingly, the illumination level increased with the length of the route. As a consequence, the advantages of shaded routes become smaller because the sum of the illumination accumulates and depending on the length can be similar to the sum of illumination of the sunny route. However, if the difference in length is not important, a shaded route would likely be preferred over a route without shade on hot days.

Further research will focus on the integration and elaboration of further attributes that are relevant for optimal pedestrian routing and navigation, focusing on the development and testing of prototypes relying on the acquired knowledge, and on raising pedestrians awareness of risky situations and behavior while using their mobile smart devices.

As we expand to new and larger test sites, future work will also evaluate the most adequate algorithm with the best time performance to calculate the shortest path, examine access databases that contain information related to building heights, and apply procedural modeling to generate 3D city models.

ACKNOWLEDGMENT

This work was supported by the Perron Project - Enhanced Pedestrian Routing and Navigation as well as Walkability Assessment of Pedestrian Ways, funded by European funds through the Flagship Call 2013 Future Travelling, ERA-NET TRANSPORT III.

REFERENCES

[1] "Walkonomics: Find a Walkable Route," http://www.walkonomics.com/.
[2] R. Methorst, "Identification of quality needs," *Pedestrians' Quality Needs*, p. 49, 2010.
[3] R. Methorst, H. Monterde i Bort, R. Risser, D. Sauter, M. Tight, and W. Jim, "Pedestrians' Quality Needs, Part C, Executive Summary," Tech. Rep., 2010.
[4] P. J. M. de Cambra, "Pedestrian accessibility and attractiveness indicators for walkability assessment," Ph.D. dissertation, Thesis for the Master Degree (MSc) in Urban Studies and Territorial Management, 2012.
[5] P. K. Maghelal and C. J. Capp, "Walkability: a review of existing pedestrian indices," *URISA Journal*, vol. 23, no. 2, p. 5, 2011.
[6] S. Abley and S. Turner, *Predicting walkability*. NZ Transport Agency, 2011, no. 452.
[7] R. Ewing and S. Handy, "Measuring the unmeasurable: urban design qualities related to walkability," *Journal of Urban design*, vol. 14, no. 1, pp. 65–84, 2009.
[8] A. Forsyth and M. Southworth, "Cities AfootPedestrians, Walkability and Urban Design," 2008.
[9] M. de Almeida, "The Importance of Shade as a Strategy to foster Walking in Summer," in *Proceedings of. the 8th International Conference for Walking by Walk*, vol. 21, 2007.
[10] "Openstreetmap," http://www.openstreetmap.org.
[11] "Graph Integration Platform," http://www.gip.gv.at/gipat-en.html.
[12] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
[13] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
[14] D. Delling, P. Sanders, D. Schultes, and D. Wagner, "Engineering route planning algorithms," in *Algorithmics of large and complex networks*. Springer, 2009, pp. 117–139.
[15] W. Zeng and R. Church, "Finding shortest paths on real road networks: the case for A*," *International Journal of Geographical Information Science*, vol. 23, no. 4, pp. 531–543, 2009.
[16] A. Stentz, "Optimal and efficient path planning for partially known environments," in *Intelligent Unmanned Ground Vehicles*. Springer, 1997, pp. 203–220.
[17] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 354–363, 2005.
[18] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning A," *Artificial Intelligence*, vol. 155, no. 1, pp. 93–146, 2004.
[19] C. Dang, M. Iwai, K. Umeda, Y. Tobe, and K. Sezaki, "Navicomf: Navigate pedestrians for comfort using multi-modal environmental sensors," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2012, pp. 76–84.
[20] O. Czogalla and A. Herrmann, "Parameters determining route choice in pedestrian networks," in *TRB 90th annual meeting compendium of papers DVD. Washington, DC*, 2011, pp. 23–27.
[21] H. Monterde-i Bort, C. Johansson, L. Leden, and S. Basbas, "ITS and on-trip tasks while walking," *COST 358 PQN Final Report Part B1*, pp. 157–170, 2010.
[22] T. Heinroth and D. Buhler, "Arrigator evaluation of a speech-based pedestrian navigation system," in *Intelligent Environments, 2008 IET 4th International Conference on*. IET, 2008, pp. 1–4.
[23] A. Stark, M. Riebeck, and J. Kawalek, "How to design an advanced pedestrian navigation system: Field trial results," in *4th IEEE workshop on Intelligent data acquisition and advanced computing systems: Technology and applications, IDAACS*. IEEE, 2007, pp. 690–694.
[24] S. Schwarz, D. Sellitsch, M. Tscheligi, and C. Olaverri-Monreal, "Safety in pedestrian navigation: Road crossing habits and route quality needs," in *Future Active Safety Technology Towards zero traffic accidents, FAST-zero 2015 Symposium, Gothenburg, Sweden*, 2015, pp. 305–310.
[25] J. Wen, W. S. Helton, and M. Billinghurst, "Classifying users of mobile pedestrian navigation tools," in *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*. ACM, 2013, pp. 13–16.
[26] J. Gonçalves, J. S. Gonçalves, R. J. Rossetti, and C. Olaverri-Monreal, "Smartphone sensor platform to study traffic conditions and assess driving performance," in *IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 2596–2601.
[27] "Openstreetmap," http://wiki.openstreetmap.org/wiki/OsmfilterDownload.
[28] "Entfernungsberechnung," http://www.kompf.de/gps/distcalc.html.
[29] "Wiener Flächenwidmungs- und Bebauungsplan," https://www.wien.gv.at /flaechenwidmung/public/.
[30] "Die Zeitgleichung, eine einfache Formel zu Sonnenaufgang und Untergang," http://lexikon.astronomie.info/zeitgleichung/index.html.
[31] "Berechnung von Sonnenhöhe und Azimut mit Tabellenkalkulation," http://www.jgiesen.de/SME/tk/.
[32] "Wetter.com," http://at.wetter.com/apps_und_mehr/website/api/dokumentation.
[33] "Openweathermap," http://openweathermap.org/api.
[34] "Static maps api v2," http://developers.google.com/maps/documentation/staticmaps/?hl=en.
[35] G. Martin, M. Hogg, and D. Abrams, "Social cognition and attitudes," *Journal of psychology, Harlow: Pearson education limited, page*, no. 646-677, 2010.
[36] F. Seco, C. Prieto, J. Guevara *et al.*, "A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU," in *IEEE International Symposium on Intelligent Signal Processing (WISP)*. IEEE, 2009, pp. 37–42.
[37] K. Seifert and O. Camacho, "Implementing positioning algorithms using accelerometers," *Freescale Semiconductor*, 2007.
[38] "NEO-7 Series u-blox 7 GNSS-Module," https://www.u-blox.com/de/product/neo-7-series.
[39] "RTKLIB: An Open Source Program Package for GNSS Positioning," http://www.rtklib.com/.
[40] "NEO-7P u-blox 7 GNSS-Module with Precise Point Positioning," https://www.u-blox.com/de/product/neo-7p.
[41] P. W. Bradford, J. Spilker, and P. Enge, "Global positioning system: theory and applications," *AIAA Washington DC*, vol. 109, 1996.
[42] "AXIO-NET GmbH," http://www.axio-net.eu/.
[43] "Satellite Positioning Service of the German State Survey," http://www.sapos.de.
[44] J. Kouba and P. Héroux, "Precise point positioning using IGS orbit and clock products," *GPS solutions*, vol. 5, no. 2, pp. 12–28, 2001.

Cristina Olaverri Monreal graduated with a Masters degree in Computational Linguistics, Computer Science and Phonetics from the Ludwig-Maximilians University (LMU) in Munich 2002 and received her PhD 2006 in cooperation with BMW. After working several years in the US and different countries in Europe, both within the industry and academia, she currently leads the Competence Team Intelligent Technologies in Smart Cities at the University of Applied Sciences Technikum Wien, Austria. Her research aims to minimize the barrier between users and systems in complex, dynamic scenarios that are critical to decision making processes, such as driving a vehicle and innovative forms of mobile and ubiquitous interaction approaches to human mobility.

Matthias Pichler graduated with a Masters degree in Intelligent Transport Systems from the University of Applied Sciences Technikum Wien in Vienna 2015 in cooperation with the Austrian Institute of Technology. Currently he works for EBE Solutions GmbH in the field of developing railway systems.

Gerd Christian Krizek graduated in Theoretical Physics at the University of Vienna in the group of Caslav Brukner. He has been working since 1999 in the industry as engineer in the field of embedded systems and mechatronics. Since 2010 he is deputy program director of the Master degree program ”Intelligent Transport Systems” and lecturer at the University of Applied Science Technikum Wien.

Sebastian Naumann graduated in 2001 with a Diploma Degree in Computer Science at the Technical University of Ilmenau, Thuringia, Germany. Since 2002 he works as a researcher at the Institute f. Automation and Communication in Magdeburg, Saxony Anhalt, Germany, a non profit research organization which is engaged in applied sciences in the field of Transport & Assistance. He has been working on several mobility-related national and transnational research projects.