robotics

MDPI

Article

Application of Path Planning for a Mobile Robot Assistance System Based on OpenStreetMap Data

Pascal Stahr *,†, Jochen Maaß † and Henner Gärtner

Faculty of Engineering and Computer Science, Hamburg University of Applied Sciences (HAW Hamburg), Berliner Tor 5, D-20099 Hamburg, Germany; jochen.maass2@haw-hamburg.de (J.M.); henner.gaertner@haw-hamburg.de (H.G.)

* Correspondence: pascal.stahr@haw-hamburg.de

† These authors contributed equally to this work.

Abstract: For visually impaired people, orientation and mobility are challenging and require a great deal of concentration. Especially unfamiliar routes are difficult to follow. This paper presents a navigation algorithm based on a geographic information system that can be optimally adapted to the needs of this user group. A three-stage process is being developed for this purpose. The first step is to check the map against ISO 19157, followed by map augmentation and the generation of a navigable graph. Finally, a new cost function for an A* algorithm is developed to make the best use of the modified map data and adapt it to the user's needs. It is shown that map data from the geographic information system OpenStreetMap is well suited to the task, although the map data needs to be verified and augmented with additional information before being used. Finally, we compare the presented solution with a standard A* algorithm.

Keywords: path planning; mobile assistance system; navigation; OpenStreetMap; mobile robot

check for updates

Citation: Stahr, P.; Maaß, J.; Gärtner, H. Application of Path Planning for a Mobile Robot Assistance System Based on OpenStreetMap Data. *Robotics* **2023**, *12*, 113. https://doi.org/10.3390/robotics12040113

Received: 8 May 2023
Revised: 19 July 2023
Accepted: 1 August 2023
Published: 4 August 2023

1. Introduction

In recent years, a strong increase in commercial and near-commercial autonomous vehicles, that operate on the sidewalk in urban environments, are seen (for example [1–4]) . Intuitively, the performance of those autonomously navigating vehicles depends on the quality of the map data as strongly as on the quality of the position-sensing equipment. The Shared Guide Dog 4.0 (SGD4.0) (see Figure 1) is an autonomous vehicle under development in cooperation with a non-profit foundation (Blinden- und Sehbehindertenverein Hamburg e.V. [5]) aiming at the guidance of visually impaired persons. It is beneficial to provide that sort of solution because orientation in unknown environments still poses a substantial challenge to visually impaired persons. Statistics show [6], that visual imparity strongly coincides with disabilities of the musculoskeletal system. Therefore, the Shared Guide Dog 4.0 resembles a rollator-walker, which can also be used to sit down and rest on it. The awareness of the demands of visually impaired persons in society and the economy has risen, and digital aides become available. Examples are OrCam, a small device capable of reading text, recognizing people, and identifying products [7] or Routago, a specialized smartphone navigation application that incorporates additional features to ease the orientation of disabled persons [8].

Even though geodata are available from various sources, such as the commercial map distributors Google, Microsoft, Nokia HERE [9], and TomTom [10], a generalized representation of the environment is still missing, and publicly available survey data are not represented with navigation in mind in the first place. High-precision survey data exists from various projects [11] but focuses on road traffic environment data, including traffic signs, parking lots, and electric vehicle charging points. Geodata collected specifically aiming at benefits for impaired persons are missing so far.

The OpenStreetMap project aims to create and maintain an open-source map of the world [12] and by March 2023 about 10 million registered users were contributing to it [13]. A guidance application for disabled persons can benefit substantially from a community of volunteers that adds, maintains, and complements features accordingly in a logically consistent way. OpenStreetMap's data have already proven useful in various research projects and is also used by commercial transportation companies such as Uber and Grab [14].

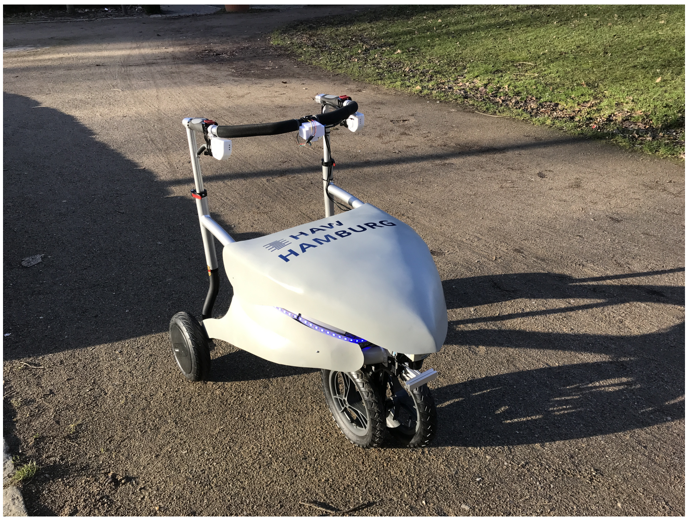

Figure 1. The Shared Guide Dog 4.0, an autonomous guidance vehicle resembling a rollator-walker.

With the advancing development of autonomous driving vehicles, the combination of local sensor data and a global environment representation by a given map is increasingly becoming the focus of research. A common method for recording an environment representation is to manually drive the mobile robot through an unknown environment and have the sensors record the environment. This approach has the disadvantage that the initial recording and processing of the map takes a lot of time and has to be repeated every time there is a change in the environment.

In this paper, we will present an approach that uses the OpenStreetMap as the database for creating a global environment map and computing an optimal global path for the mobile robot. We will look at the entire process from analyzing the data to using it on the Shared Guide Dog 4.0.

The focus is on the data model of the OpenStreetMap project and it is shown how features for disabled people have already been incorporated. Typical data inconsistencies are discussed, that impede the creation of a navigable graph required for the operation of an autonomous vehicle. An instruction set to augment the geodata and an algorithm to preprocess the results are derived to overcome those problems. Sidewalks and pedestrian zones, which are usually represented as polyhedral objects and are not navigable per se, are given special consideration and processing.

Based on the results of the preprocessor, a path planning algorithm is introduced that allows a trade-off between the pavement quality, a keep-right policy, and the distance to be covered. The results will be discussed to evaluate the suitability of the OpenStreetMap as a basis for sidewalk path planning and the applicability of the navigation algorithm.

The entire code of the Shared Guide Dog 4.0 can be found at https://github.com/PStahr-HAW/ros2-sgd4.0 (accessed on 17 July 2023), the later used JOSM plugin can be found at https://github.com/PStahr-HAW/JOSM-osm2x (accessed on 17 July 2023).

1.1. Related Work

In recent years, several projects investigated route finding for mobility-impaired people based on Volunteered Geographic Information (VGI). A major challenge is the completeness and accuracy of the data used, which has been studied in many articles. The focus is on the suitability of OpenStreetMap as a database and the navigation on the sidewalk.

1.1.1. OpenStreetMap

All of the studies examined conclude that OpenStreetMap provides the most complete map among the open-source maps, but before any use, the test area should be carefully checked.

In one paper[15], special focus is put on the completeness of sidewalk information based on three types: object, attribute, and value. The result is that the completeness in the presented German cities is mostly well below 20%. Before each use, the sidewalks have to be enriched with further attributes, especially because in many cases the sidewalk information is assigned to the streets and thus no conclusions can be drawn about the geometry and exact position. Ding et al. [16] studied the attributes for wheelchair users in England and concluded that about 98% of the nodes do not have wheelchair attributes. Although there is a specialized OpenStreetMap for wheelchair users (Wheelmap) the completeness is very low. Therefore, it is recommended to use additional data sources to provide greater coverage.

Despite the limitations in data quality, ref. [17] concludes that Open Data sources will be of great relevance in the future. OpenStreetMap is of particular interest as the most promising candidate due to its large and active community. Prior to use, it is recommended to compare the map data with ground truth information or at least to perform an intrinsic verification where only a logical estimation of the attributes takes place.

1.1.2. Sidewalk Navigation

In recent years, there have been several projects that studied route finding on sidewalks for user groups with different needs.

In one paper [18], a system was developed that uses OpenStreetMap for pedestrian navigation. In addition, a crowd sourcing system is implemented to collect obstacles and add them to the map. The pedestrian path generation is performed using an offset from the road. Based on this data, an optimal path, measured by a ‘difficulty criterion’, is then calculated.

In another paper [19], a service-oriented algorithm for a routing algorithm is presented. The map is transformed into a graph-like structure and a multicriterial algorithm is introduced for cost calculation. Attributes are assigned to nodes, which are weighted differently based on policies that users are free to choose. A graph wayfinding algorithm for cost minimization is then used to compute a suitable path according to the user preferences.

In one paper [20], a wayfinding tool for computing optimal paths for blind people is developed. The OpenStreetMap data are used to generate a weighted network graph and compute a path based on the criteria length, complexity, landmarks, and path type. The facts that preprocessing of the map is not considered and third-party software is used for path computation, which is why this solution is not suitable for the Shared Guide Dog 4.0.

Opach et al. [21] conducted a study on user behavior when selecting different routes to avoid potential flooding areas. Different routes were suggested to the users to reach the destination. These always include the shortest route and a route that avoids any flooding area. The study concludes that, on average, users are willing to take a route that is up to 10% longer if it avoids hazardous areas.

The project described in [1] focuses on processing OpenStreetMap data to prepare it for a delivery robot. The presented toolchain does not handle mapping errors and focuses on streets. Ways with a width of less than 3 m are not considered. In the presented approaches, the adaptations to the map are not sufficient for allowing a mobile robot to safely drive on footways.

According to our research, there is currently no fully comprehensive routing solution for mobility-impaired users on the sidewalk that allows for customization.

1.2. Paper Organization

The rest of the paper is organized as follows. The next section introduces the software architecture for navigation and then the OpenStreetMap data model. In Chapter 2, the steps needed to prepare the dataset to create a navigable dataset are covered. This is followed by developing a cost function for a path planning algorithm on the Shared Guide Dog 4.0. The paper concludes with an example and a comparison of the calculated route with a route calculated using a standard A* algorithm.

1.3. Architecture of the Shared Guide Dog 4.0

The control architecture of the Shared Guide Dog 4.0 is based on a layer model with three layers.

The top layer consists of the Global Planner, which calculates a path between the selected destination and the current position. At the start of navigation, the Global Planner calculates a route and then passes this route to the second layer, the Local Planner. The Local Planner calculates local avoidance strategies if the global path is blocked by obstacles and reacts to objects that are in the close-up range around the Shared Guide Dog 4.0. A LiDAR scanner is used to scan the environment and the local planner generates an alternative route if an obstacle is detected. If there are no obstacles in the vicinity of the Shared Guide Dog 4.0, the local path equals the global path. On the Shared Guide Dog 4.0, an implementation of the time elastic band planner (TEB Planner [22]) is used for local path planning. The bottom layer consists of the controller, which controls the movements to stay on the planned path. This paper discusses an adaptation and implementation of the global planner as a basis for the other layers as well as the example application on the Shared Guide Dog 4.0.

1.4. OpenStreetMap Data Model

In the OpenStreetMap project, a free and open-source map of the world is created that is accessible to everyone. The map is created by the community. In addition to the world map, many software solutions to edit or use the map have been created as part of the project.

The map data are stored in the XML file format and are built from nodes, ways, and relations [23]. A node represents a point on the Earth's surface which is defined through latitude and longitude. Nodes are the core element of OpenStreetMap and are referenced by every other element. The minimum requirements for a node are thus a unique ID u_N and the specification of the longitude λ and latitude φ. The longitude and latitude are specified as decimal numbers with seven digits of decimal precision. This ensures a positional accuracy of less than ±5.6 mm. In addition, nodes N can be provided with tags that can store further information.

$$N = \{u_N, \varphi, \lambda\} \tag{1}$$

The connection between nodes is established by ways W. They are defined as a collection of nodes and stored as a list of node IDs. An area is a way where the start node equals the end node.

$$W = \{u_W, u_{N,i}|_{i=1\dots k}\} \tag{2}$$

If paths and nodes are to be connected to form more complex structures, relations are used. A relation can contain nodes, ways, and other relations.

$$R = \{u_R, u_{N,i}|_{i=1\dots k}, u_{W,i}|_{i=1\dots l}, u_{R,i}|_{i=1\dots m}\} \tag{3}$$

Tags are used to describe properties of map elements represented by key-value pairs and can be added to nodes, paths, and relations. Both the key and value are free text fields but there are recommendations for which tags describe which objects. Examples of commonly used tags are highway = secondary or amenity = bench.

The following Listing 1 shows the definition of a node, a way, and a relation in OpenStreetMap data format. Each element has an additional tag to specify its type in the map.

Listing 1. Example of node, way, and relation represented in OpenStreetMap's XML data format: Ways and relations are formed by lists of references to IDs.

```
<node id = ‘253416079’ lat = ‘53.5571377’ lon = ‘10.0197035’>
    <tag k = ‘amenity’ vs. = ‘bench’ />
</node>
<way id = ‘5229522’>
    <nd ref = ‘253416079’ />
    <nd ref = ‘7156477920’ />
   <tag k = ‘highway’ vs. = ‘residential’ />
</way>
<relation id = ‘2872789’>
    <member type = ‘node’ ref = ‘6055658161’ />
    <member type = ‘way’ ref = ‘466451524’ />
    <member type = ‘way’ ref = ‘383422087’ />
</relation>
```

2. Path Planning

Path planning is divided into three steps: preprocessing of OpenStreetMap data, creation of a navigable graph, and route computation. While the preprocessing and graph generation is performed offline, the route computation is performed online on the Shared Guide Dog 4.0.

The presented toolchain is based on the JavaOSM (JOSM) editor [24]. It is an open-source editor for various OpenStreetMap data, which can be used to download the map, edit it with various tools, and export it in different formats. In addition, JOSM is extensible through a variety of plugins and user-specific plugins can be integrated.

2.1. Preparing OpenStreetMap Data

The quality of the OpenStreetMap varies widely around the world, depending on the activity of the OpenStreetMap community. But there is no metric to describe the mapping quality. The OpenStreetMap community claims that the "OpenStreetMap is often more up-to-date and of a higher quality than other commercial maps when dealing with New and Changed Ways" [25]. However, since the OpenStreetMap data is created by volunteers, it can be assumed that the population density leads to a higher data quality in urban areas in comparison with rural areas. Furthermore, the OpenStreetMap project states "Our aim is to be the best world map for accessibility" [26]. There are several projects, such as OpenStreetMap for the blind [27], which incorporate tactile pavings, acoustic signals at traffic lights, points of interest (POI) especially for blind people, accessibility by wheelchair, etc. by attributes. However, the coverage varies strongly and there is no all-in-one solution with global coverage.

In order to assess the accuracy and completeness of the OpenStreetMap, an analysis according to ISO 19157 [28] is performed. The map is evaluated according to the criteria completeness, thematic accuracy, logical consistency, temporal quality, positional accuracy, and usability. The results are presented in Table 1. The analysis is first performed by comparing the OpenStreetMap data with aerial photographs from the Land Registry of the City of Hamburg. This allows the positional accuracy to be determined. After the comparison of the aerial images, the locations are inspected in reality. Thus, the completeness, temporal quality and thematic accuracy can be evaluated.

The criteria of positional accuracy and completeness are evaluated again in detail in the following two sections. These two criteria are crucial for the use of the map and the quality of the routing solution.

Table 1. Evaluation of the OpenStreetMap data against ISO 19157.

Criterion	Description	Rating
Completeness	Completeness of features, attributes, and relations	good
Logical consistency	Rules of the data set were observed	good
Positional accuracy	Deviation from the real geographical position	very good
Thematic accuracy	Correctness of attributes and classification of features	good
Temporal quality	Representation of the current status	very good
Usability	Applicability for the project	very good

2.1.1. Positional Accuracy

On the left side of Figure 2 a city map with the scale 1:500 is used as the background, on the right side is an aerial photo with a resolution of 20 cm. Both images are published and georeferenced by the Land Registry of the City of Hamburg. Overlaid on the images are the OpenStreetMap data, which are shown in red. The alignment is based on the corner points of the basketball court in the middle of the images.

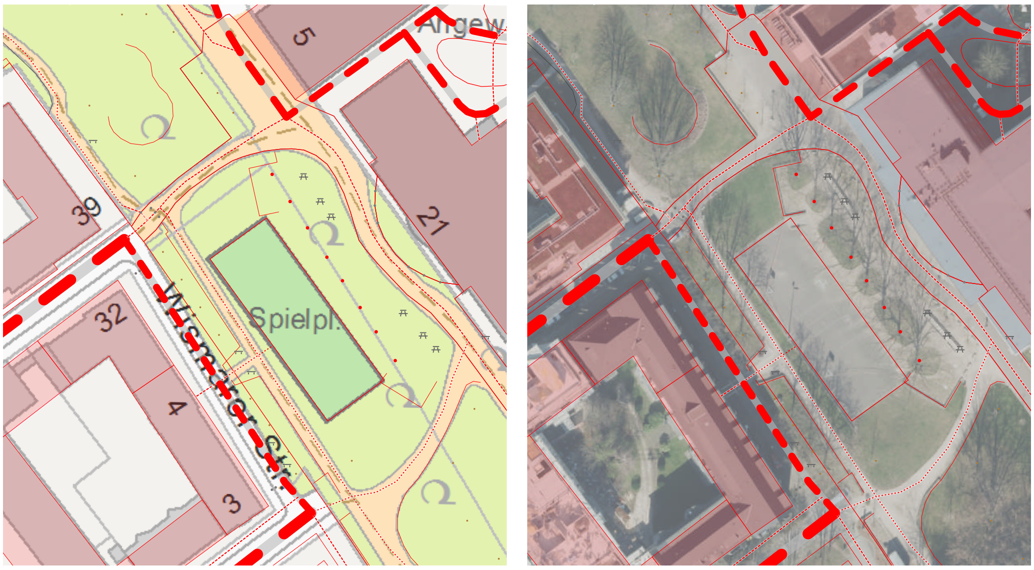

Figure 2. Evaluation of the positional accuracy of the OpenStreetMap data based on the data from the cadastral office of the city of Hamburg. The data from OpenStreetMap is shown in red.

The accuracy of the paths can be rated as very good based on the comparison, even though some paths are not exactly centered. The accuracy of objects such as benches, trees, and walls can also be rated as very good. No deviations can be found here. Only the observation of the buildings results in larger deviations. On the one hand, these are positional deviations, as in the case of the building at the top right. On the other hand, there are inaccuracies due to a too-coarse representation of the buildings, as can be seen below and in the center on the left.

2.1.2. Completeness/Temporal Quality/Thematic Accuracy

Completeness, temporal quality, and thematic accuracy are evaluated by walking the test area in reality. It turns out that trees, benches, barriers, and walls are almost completely drawn in. On the other hand, trash cans and boundary stones used to border the path are not drawn in at all. To ensure safety on any path, all these objects should be included in the route calculation. Some common errors in the OpenStreetMap data are shown in

Figure 3. In conclusion, the completeness analysis shows that many objects are already present, but verification and post-processing are required.

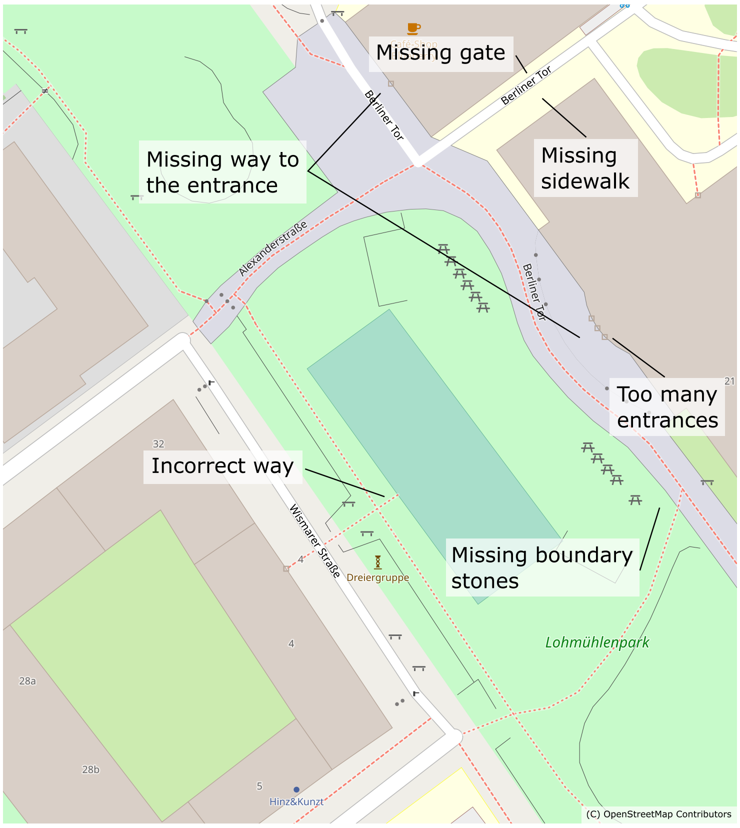

Figure 3. Completeness errors like missing and too many objects in the OpenStreetMap data in the test area.

2.2. Augmentation

In the previous section, it was shown that a review and augmentation of the data is necessary. As part of the OpenStreetMap project, quality assurance tools were developed, which allow both an automatic check of the map and manual reporting of mapping errors to be re-evaluated. In the quality assurance process for the Shared Guide Dog 4.0 project, the JOSM validator, a core functionality of the JOSM editor, is used. It is capable of finding mapping errors, such as intersecting paths without a common node, closely spaced nodes, and suspicious tag combinations. These errors, also referred to as logical inconsistencies of the data set, can be found and corrected very well by automated tools. However, the errors caused by logical inconsistency only make up a small part of the overall errors in OpenStreetMap.

The most common errors in the OpenStreetMap data processed for the Shared Guide Dog 4.0 project are missing ways, footways not separated from streets, missing barriers, and missing or wrong attributes. Since the OpenStreetMap is primarily created for street vehicles, completely missing separate footways are prevalent. In many cases, there is no distinction between a street and its sidewalk. The sidewalk is either added to the road as an attribute with no additional information about its properties or there may be not even any information about the sidewalk at all. These errors are classified as errors in completeness and errors in thematic accuracy. They cannot be detected by the JOSM validator and therefore manual processing must be performed.

In order to enable a standardized procedure, the following instruction set was developed. When processing the individual steps, the Shared-Guide-Dog-Plugin for the JOSM Editor can be helpful and is freely available at https://github.com/PStahr-HAW/JOSM-osm2x (accessed on 13 March 2023). The entire process of meshing and creating an obstacle map is supported. For example, the interpolation and generation of new nodes during meshing are simplified by a visualization tool for detecting errors.

1. Check whether the dataset is fully connected (every node is reachable from every other node)
2. Search for ways with unnamed dead ends
3. Add separate footways
4. Convert areas into ways
5. Check if all objects are presented in the map
6. Add the relevant tags to all ways

1: The first step is to check whether the data set in question is connected as one complete net or if there are multiple disconnected nets in the dataset. This step aims to have only one connected net in which every node is reachable from every other node.

2: Paths that end in a dead end should be given an address in order to make them navigable. Otherwise, the path cannot be traversed.

3: As mentioned before, the footways in OpenStreetMap are often added as attributes to streets. Therefore it needs to be verified that every street containing the tag footway has its own footway. Furthermore, it must be checked that all footpaths are present on the map. Figure 4 shows the necessary preparation of the data: Additional paths are added to represent the sidewalks including an attribute for the width of the sidewalk. Note, that the nodes next to the pedestrian crossing are members of 3 ways each and therefore are navigable. The old ways describing the streets are kept for reference but are not connected to the sidewalks.

4: Pedestrian areas are often represented as polygons, containing no navigable ways. Existing navigation algorithms based on the OpenStreetMap data route along the borders of the polygons result in an undesirable path as shown in Figure 5. Due to the complex structure and obstacles that can be found inside the pedestrian area, the polygon is augmented by manually inserting new ways and adding a suitable width attribute.

5: The second to last step is to check whether all existing obstacles are represented on the map. In particular, these obstacles include posts and barriers which may block the way for the Shared Guide Dog 4.0.

6: After step 5, the preparation of the path network for the Shared Guide Dog 4.0 is completed. The last step is to check if all paths feature the required attributes. The pathfinding algorithm presented in Section Path Planning requires the tags highway, surface and sgd_width. Other attributes, such as hazard, are optional and can be added as needed. The OpenStreetMap data model has an official attribute for the width, but its use is optional. Due to missing information on the measurement method, for example whether curbs were taken into account or not, the value should not be used without verification. To avoid misidentification with the width attribute, a new attribute sgd_width is introduced. This attribute represents the drivable width of the way.

Currently, these preparations must be performed manually since they require knowledge of the traffic situation. However, it is conceivable that such verification could be carried out by machine learning algorithms in the near future.

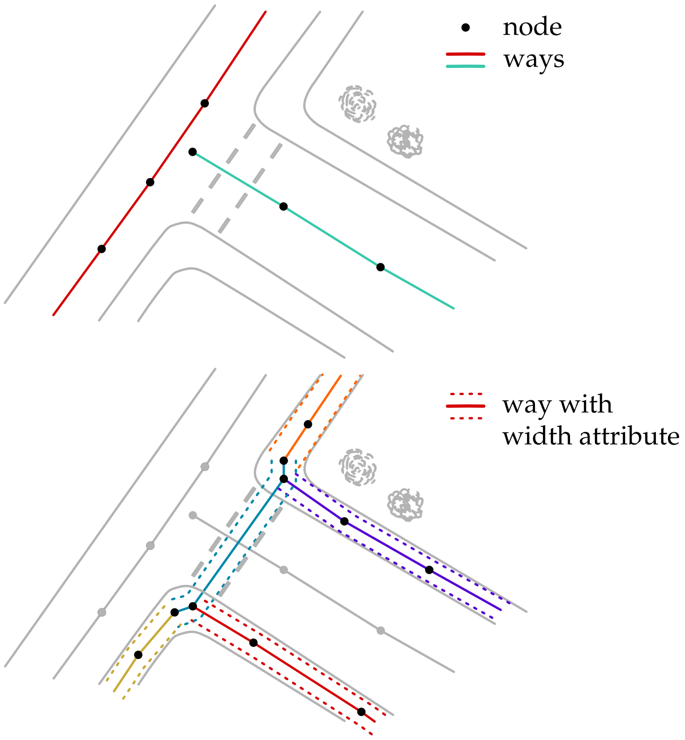

Figure 4. Typical sparse way information (**top**) and the result from the augmentation: A connected graph with width attributes (**bottom**).

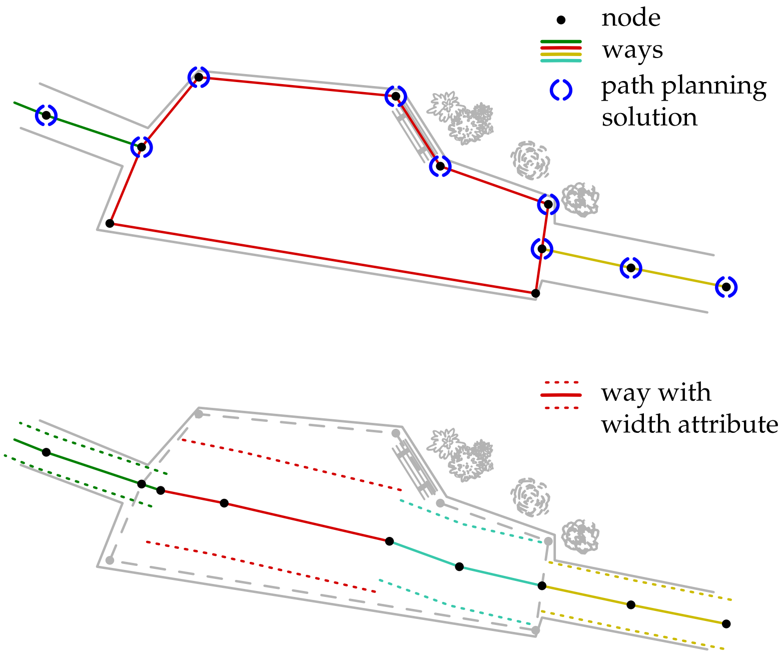

Figure 5. Pedestrian area represented by a polygonal way ((**top**), red) resulting in an inappropriate path planning solution and the same area after augmentation of the geodata (**bottom**).

2.3. Path Generation

The OpenStreetMap data cannot be used directly for navigation in its native data format. Nodes are connected by ways, but only connections along the current way are visible. Ways to further neighboring nodes cannot be accessed for navigation. In order to create a navigable graph, additional further processing must be carried out. These operations are divided into the three steps of interpolation, creation of a path network, and connection of the nodes, as shown in Figure 6.

The goal of meshing nodes is to create a navigable connected network of ways that span across all drivable surfaces. We chose to augment each prepared way by two additional ways which are connected in a mesh pattern to each other. These three lanes represent one lane per direction and one lane for passing. With this method, it is possible to significantly increase safety using a right-hand walking rule. Even though there is no requirement to walk on the right side in the test area, it is natural for people to do so because they know it from road traffic. Another advantage is that paths that are partially blocked can still be considered during path planning and a path around the obstacle can already be planned at the start. It leaves less work to the Local Planner which would detect an obstacle by vehicle sensors at a later stage. For example, a long-term construction site blocking only half of the path can already be considered in an easy way during path planning so that the Shared Guide Dog 4.0 can be navigated to the free side of the path.

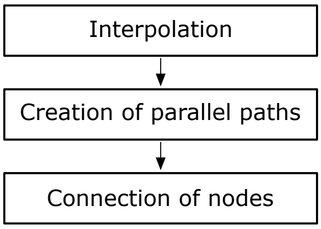

Figure 6. The meshing algorithm is divided into three steps: interpolation, creation of parallel paths, and connection of nodes.

Figure 7 shows three original OpenStreetMap ways with a width attribute as an example input data set to showcase the meshing algorithm's mode of operation.

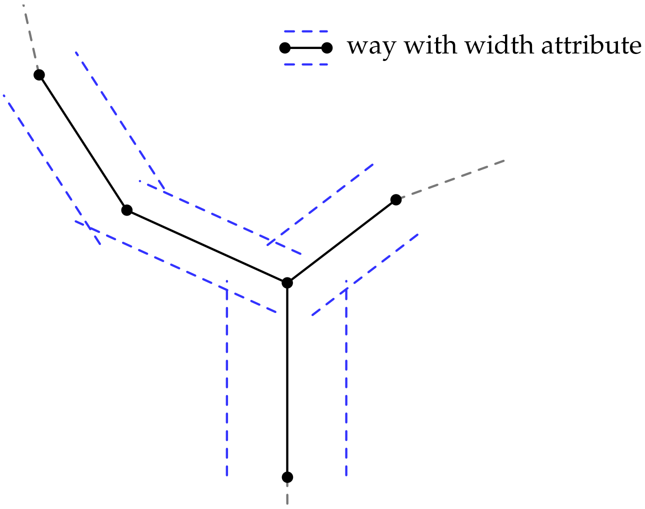

Figure 7. Example paths from the OpenStreetMap dataset with an augmented width attribute.

2.4. Interpolation

To allow changing the path in short intervals within the navigable connected network, ways with a length greater than a threshold a are linearly interpolated. During some tests with blind people, a distance of 5 m has been found to be a suitable interpolation distance. This allows the lane change for short distances, but at the same time does not form large angles during the lane change.

For each point, the latitude φ and longitude λ are specified. The presented calculations are based on a Mercator projection with WGS84 as the reference system. To take into account the declining projection length of the longitudes toward the poles, a scale factor $\varphi_1\varphi_2 = \cos\big((\varphi_1 + \varphi_2) \cdot \frac{\pi}{360^\circ}\big)$ is defined. The approximate distance d between two points x

and y is calculated as the Euclidian distance by the following equation. R is the rounded median earth radius as defined by the WGS84 reference system as $R = 6{,}371{,}000\,\text{m}$.

$$d = \sqrt{x^2 + y^2} \tag{4}$$

$$x = (\lambda_1 - \lambda_0) \cdot R \cdot \frac{\pi}{180^\circ} \cdot \varphi_0 \varphi_1 \tag{5}$$

$$y = (\varphi_1 - \varphi_0) \cdot R \cdot \frac{\pi}{180^\circ} \tag{6}$$

The distance between the two nodes is then divided by the threshold value for the interpolation a to calculate the number of new nodes.

$$n = \left\lceil \frac{d}{a} \right\rceil \tag{7}$$

The number of nodes to insert is then used to create new nodes

$$N\left\{u_{N,i}, \varphi + i \cdot \frac{\Delta\varphi}{n}, \lambda + i \cdot \frac{\Delta\lambda}{n}\right\} \tag{8}$$

with $i = 1 \ldots (n-1)$. To complete the interpolation, the created nodes are inserted into the existing ways and connected to each other as shown in Figure 8.

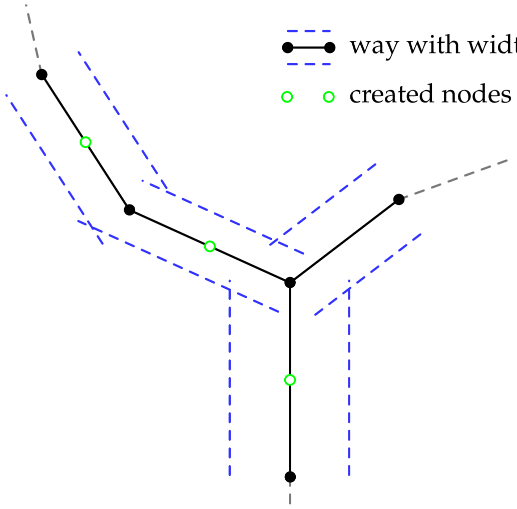

Figure 8. Interpolated ways and newly added nodes before meshing.

2.5. Creating the Set of Parallel Paths

The calculation of new nodes for parallel paths is based on the paths to the neighboring nodes. A direction of and a distance to the original node are required for determining the position of a new node. The bisector between the paths is used to calculate the direction. Figure 9 shows the naming convention used for the calculation of new points.

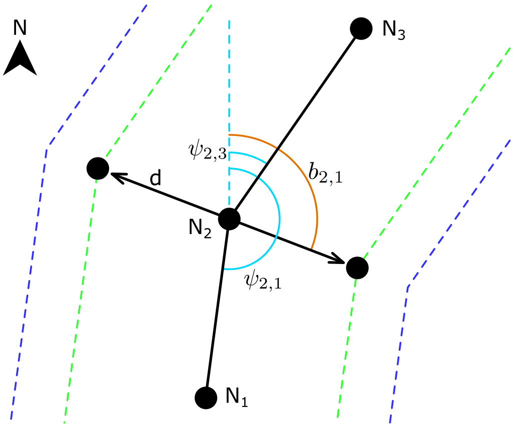

Figure 9. Nomenclature of the angles when calculating new points for path meshing.

New nodes are inserted based on the following equations equation with $N_1\{u_{N,1}, \varphi_1, \lambda_1\}$ being the predecessor node, $N_2\{u_{N,2}, \varphi_2, \lambda_2\}$ being the current node and $N_3\{u_{N,3}, \varphi_3, \lambda_3\}$ being the next node on the way.

$$\varphi_{new} = \varphi_2 + \sin(b) \cdot \frac{d \cdot 180^\circ}{R \cdot \pi} \quad (9)$$

$$\lambda_{new} = \lambda_2 + \cos(b) \cdot \frac{d \cdot 180^\circ}{R \cdot \pi \cdot \varphi_2 \varphi_{new}} \quad (10)$$

b is the direction in which to insert the new node and d is the distance to the new node. To calculate the direction to the newly added node, the bisector is calculated using Equation (11). The second node is added in the opposite direction. The signum function is used in order to ensure that an angle is within the interval $[-\pi, \pi]$.

$$b_{2,1} = \frac{\psi_{2,1} + \psi_{2,3}}{2} \quad (11)$$

$$b_{2,2} = b_{2,1} - b_{1,2} \cdot \pi \quad (12)$$

$\psi_{2,1}$ and $\psi_{2,3}$ are the angles of the paths from N_2 to N_1 and from N_2 to N_3.

$$\psi_{2,1} = \arctan 2((\lambda_1 - \lambda_2) \cdot \varphi_1 \varphi_2, \varphi_1 - \varphi_2) \quad (13)$$

$$\psi_{2,3} = \arctan 2((\lambda_3 - \lambda_2) \cdot \varphi_3 \varphi_2, \varphi_3 - \varphi_2) \quad (14)$$

The width is based on the width of the two paths between which the node is to be inserted. The distance between the original node and a newly added path is calculated according to Equation (15). w_1 and w_2 are the width of the two adjacent paths and w_{sgd} is the width of the Shared Guide Dog 4.0. The additional parameter ϵ is an estimation of the positioning accuracy. With a width of the SGD of 0.7 m and an estimated positioning accuracy of 0.3 m, this results in a constant distance to the edge of the way of 0.5 m.

$$d = \frac{w_1 + w_2}{4} - \frac{w_{SGD} + \epsilon}{2} = \frac{w_1 + w_2}{4} - 0.5\,\text{m} \quad (15)$$

The two new nodes are added to the node set with a reference to the original node.

2.6. Meshing the Grid

Once the new paths left and right of the original path have been created, these three parallel paths must be connected. Therefore additional paths shall connect each node of one path with other nodes of the neighboring path. It resembles a set of railroad switches on a shunting yard. An algorithm shall achieve the described interconnection by inserting additional paths according to the following rules:

1. Nodes shall not be connected with those nodes from which they were generated
2. Nodes at the end of ways (e.g., entrances) do not generate new nodes
3. Ways with a width less than a threshold are not represented by three ways

The algorithm following these rules shall mesh the grid of nodes. The result is shown in Figure 10. After meshing, the path network is exported as a navigable graph and can be used by the path planning algorithm. The navigable graph requires that all neighboring nodes are connected and that the cost of navigating a connection between two nodes is also known.

The following Listing 2 shows an example of two nodes with their neighbors. Each neighbor node is specified by the unique ID and the path to it is defined by the included attributes.

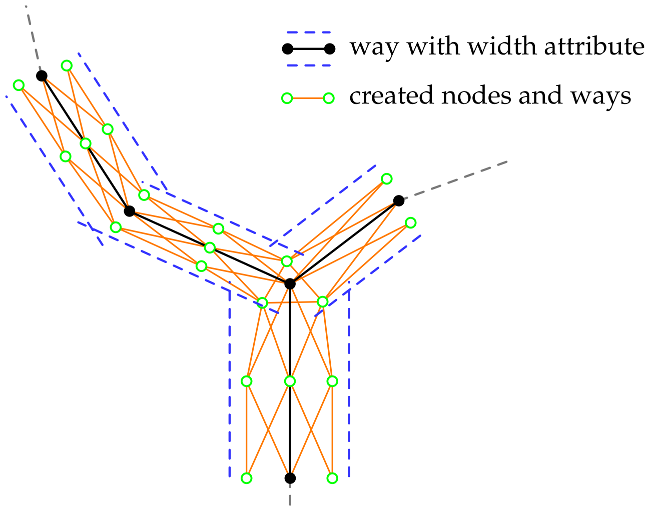

Figure 10. Result of path meshing: A fully connected set of ways.

Listing 2. Ouput data format.

```
<nodelist name = ‘lohmuehlenpark’ version = ‘0.1’>
    <node id = ‘8596122144’ lat = ‘53.5571622’ lon=’10.0207811’>
        <curb>0.1</curb>
        <nd ref = ‘8596122143’>
            <highway>footway</highway>
            <surface>paving_stones</surface>
            <sgd_width>3.0</sgd_width>
        </nd>
        <nd ref = ‘8596122145’>
            <highway>footway</highway>
            <surface>paving_stones</surface>
            <sgd_width>2.5</sgd_width>
        </nd>
    </node>
    <node id = ‘8596122143’ lat = ‘53.5570256’ lon = ‘10.0207442’>
        <barrier>bollard</barrier>
        <nd ref = ‘8596122144’>
            <highway>footway</highway>
            <surface>paving_stones</surface>
            <sgd_width>3.0</sgd_width>
        </nd>
    </node>
</nodelist>
```

3. Path Planning on the Autonomous Vehicle

Graph-based algorithms aim at minimizing the cost of a route. Often the length is used as a criterion. In this section, a multi-criteria cost function is introduced, which is used to optimize the routing solution.

The A* algorithm is a widely used algorithm, for example in mobile robots and unmanned vehicles as well as in video games. In its original form, it is used to calculate a cost-optimal path. It is based on Djikstra's algorithm and extends it with a term that estimates the cost from the current node to the destination node, the so-called heuristic. The algorithm consists of three different terms, a cost function that takes into account the cost up to the current node $g(N_i)$, the cost from the current to the next node $c(N_i, N_{i+1})$, and the estimated cost from the next node to the destination $h(N_{i+1}, N_d)$. These costs are summed up to calculate the cost of the current node.

$$f(N_i) = g(N_i) + c(N_i, N_{i+1}) + h(N_{i+1}) \tag{16}$$

Besides the original form, there are many approaches to how the A* algorithm can be improved; for example, regarding the computation time and the computed path. This adaptability of the algorithm will be used in the following to optimize the computed path for sidewalk navigation.

3.1. Cost to Current Node and Heuristic

The costs to the current node are the summed-up costs of all preceeding nodes up to the current node, where p denotes the current position.

$$g(N_p) = \sum_{i=0}^{i=p} c(N_i, N_{i+1}) \tag{17}$$

The heuristic is calculated as the Euclidan distance d to the destination node. This function was introduced in Section 2.4.

$$d = \sqrt{x^2 + y^2} = h(N_{p+1}) \tag{18}$$

$$x = (\lambda_d - \lambda_{p+1}) \cdot R \cdot \frac{\pi}{180^\circ} \cdot \varphi_{p+1}\varphi_d \tag{19}$$

$$y = (\varphi_d - \varphi_{p+1}) \cdot R \cdot \frac{\pi}{180^\circ} \tag{20}$$

3.2. Development of a Cost Function

To meet the requirements of the Shared Guide Dog 4.0, a new function is developed for calculating costs from the current node to the subsequent node $c(N_i, N_{i+1})$.

The function should satisfy the requirements that the path characteristics are taken into account and the Shared Guide Dog 4.0 is kept on the right side of the road. This results in a function consisting of two terms. One term takes into account the characteristics of the path. This means that a path with an unfavorable surface will have a higher cost and the Shared Guide Dog 4.0 will more likely choose an alternative path. Only if the detour is unreasonably long, the expensive and potentially problematic path is used.

The other term keeps the Shared Guide Dog 4.0 on the right side of the path. Without the second term, the algorithm would calculate the shortest path and cross the footway multiple times or drive on the left side of the footway. This would pose a safety risk in right-hand traffic and should be avoided. The first term is referred to as the user function c_u and the second term is called the direction function c_r. The total cost is calculated by multiplying both individual costs with the path length.

$$c(N_i, N_{i+1}) = d \cdot c_u \cdot c_r \tag{21}$$

To incorporate different user profiles, a default user is defined first, which serves as the basis for all other user profiles. Then, further users can be created that overwrite some or all parameters of the standard user. If a parameter is not overwritten, the cost function reverts to the standard user. If the path has an attribute that is not specified in the cost table, a value of 10^6 is used and thus the path is avoided if possible.

All parameters are multiplied to calculate the path function.

$$c_u(N_i, N_{i+1}) = \prod_{i=1}^{4} v_i \tag{22}$$

To determine the user profile parameters, the OpenStreetMap data must be enriched with information. This information is then used to assign costs to the paths and calculate an optimal path. In the unprocessed OpenStreetMap data, some attributes to ways often are already present, such as the type of road and the surface. In addition to these two attributes, further attributes are added for gradients/slopes and for general hazards. For nodes, however, only the barrier attribute is permitted. It represents various barriers, such as lift gates and bollards.

For the development of the path function, the road surface (asphalt, gravel, sand, . . .), the type of road (footway, residential road, highway, . . .), and other hazards are considered. Based on the attributes, an evaluation takes place. The result is a table that maps the non-numerical characteristics to factors that can be used in the cost function as shown in Listing 3. For numeric values, no mapping takes place. Instead, the absolute value is calculated to obtain non-negative values and these are used. In addition to the attributes, each path is assigned a direction value of 0.9, 1.0, or 1.1 when the network is generated. Paths in the middle always receive the value 1.0, paths on the right side the value 0.9, and paths on the left side the value 1.1. Depending on the direction of movement, a path can therefore assume different direction values. This value is used to make a movement on the right side more favorable and to support the direction function.

In right-hand traffic, the direction function must meet the requirement that a path on the right side of the way has a lower cost than a path on the left side. In addition, the function must not become zero or less than zero and the function should not unduly penalize left turns, which is defined as changing direction by more than 30 °. The coordinate system is defined so that counter-clockwise is the positive direction of rotation.

In order to fulfill the requirement for the cost increase of small angles, an exponential function with an angle-dependent exponent is chosen. In addition, it has been shown that the displacement function must not be the same for angles larger than 30 ° and angles smaller than -30 °, since otherwise small angles to the right are preferred until a large angle to the left can be driven. This behavior is undesirable and is prevented by a tangent hyperbolic function. This function ensures that even with large angles, turning to the left has a slightly higher cost than turning to the right.

Listing 3. Mapping of OpenStreetMap attributes to costs.

```
<users>
  <user name = 'default'>
    <surface>
      <asphalt>1</asphalt>
      <paving_stones>1.2</paving_stones>
      <compacted>1.5</compacted>
      <gravel>2.0</gravel>
      <cobblestone>4.0</cobblestone>
      <sand>5.0</sand>
    </surface>
    <way>
      <footway>1.0</footway>
      <service>1.5</service>
      <living_street>2.0</living_street>
      <residential>5.0</residential>
      <steps>1e6</steps>
    </way>
    <slope>1.0</slope>
  </user>
</users>
```

To ensure maximum flexibility the factors a, b, and c are inserted and can be defined by the maintainer. In general, quite small values are sufficient to achieve the desired effect. Figure 11 shows the resulting functions for different user factors.

$$c_r(N_1, N_2) = -a\psi \cdot e^{-(b\psi)^2} + c \cdot \tanh(5\psi + 1) \tag{23}$$

with

$$\psi_{1,2} = \arctan2((\lambda_2 - \lambda_1) \cdot \varphi_1 \varphi_2, \varphi_2 - \varphi_1)$$

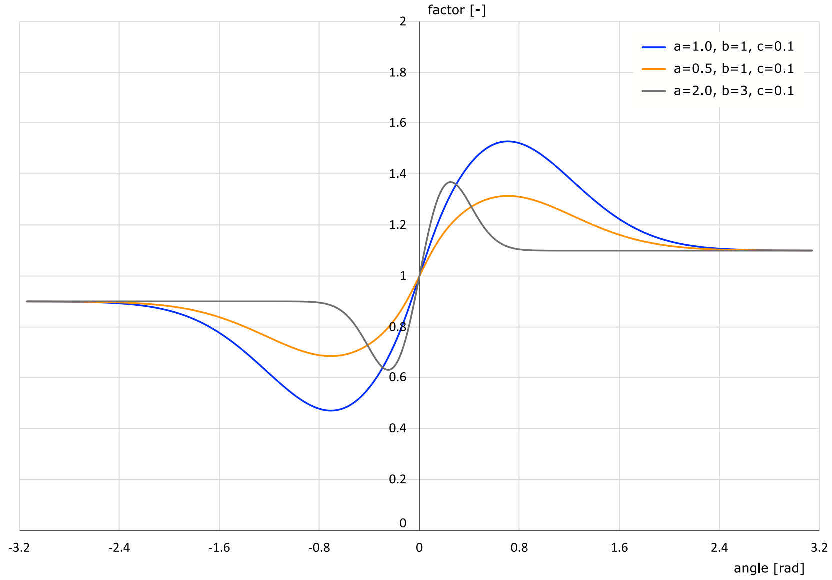

Figure 11. Visualization of different factors for the direction function.

To calculate the cost function from the current to the next node, the path function and the direction function are multiplied. Optimization of the cost function is always dependent on the user. For people who cannot walk well, for example, paths with a sandy surface can be avoided by setting the factor to 10 or even higher. The simplest way to optimize the result is to make paths that are to be avoided successively more expensive until the desired effect is achieved.

4. Example and Evaluation

A part of the Lohmühlenpark in Hamburg is considered to be an evaluation area. The park serves as a test area for the Shared Guide Dog 4.0 and has proven an ideal location because it has both small and large, street-like paths. Conveniently, motorized vehicles are prohibited.

4.1. Augmentation, Interpolation and Meshing

Figure 12 shows the map on the right after augmentation, interpolation, and meshing. Routes with the attribute highway=footway are shown in green, while routes with the attribute highway=service are shown in blue. The black line represents a wall. Clearly visible are the different widths of the paths and that a particular footway is represented by only one path, because it does not exceed the required width for meshing. The virtual deletion of occupied nodes is shown on the left. If obstacles are already known during route planning, the nodes can be occupied with very large costs and thus a bypass can be demanded. The procedure presented in the previous subsection is performed for all paths of the map. Afterward, the map is transformed into a navigable graph, exported, and downloaded to the Shared Guide Dog 4.0. The subsequent calculations for path planning and for obstacle avoidance are performed at runtime.

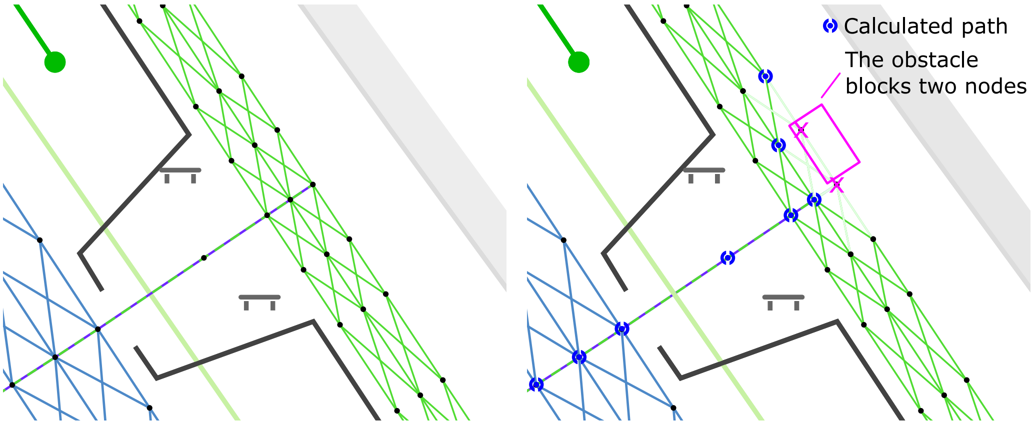

Figure 12. (**Left**): A detailed view of an intersection after augmentation, interpolation and meshing. One way is not meshed due to its small width. (**Right**): Long-standing obstacles are avoided by virtually deleting nodes.

4.2. Cost Function and Direction Function

The calculation of the cost function and the direction function for path planning is to be shown by means of an example. For this purpose, a path with the attributes as shown in Table 2 is considered.

Table 2. Attributes and user values of example path segment.

Attribute	Value
`paving_stones`	1.2
`service`	1.5
`hazard`	1
`length`	6.0

For the mapping of the values, the default user just presented is used. This results in a value of 1.2 for the surface attribute and a value of 1.5 for the way attribute. The value for hazard is 1. In order to obtain the total costs of the path segment, the values are multiplied and again multiplied by the length of the path segment.

Before multiplication, the user value is 1.8, which means that a detour up to 1.8 times the length of the current path is used to bypass the current path, provided that the other path has a user value of 1.

4.3. Path Finding and Obstacle Avoidance

For the path calculation, an A*-algorithm is used as described in Section 3. By adjusting the individual user values, a path can be found with the optimal ratio of length to difficulty.

Permanent obstacles, such as parked cars, are taken into account during path planning by a special sgd_obstacle attribute. If this is set to a high value, the path section is avoided with a very high possibility. The advantage of this obstacle avoidance is the reliability of the pathfinding. If there is no alternative route for an obstacle, a path is still planned and it can be checked locally, whether or not the obstacle is still present. And if required, a local strategy for obstacle avoidance can be used.

4.4. Comparison

To verify the route planning, a comparison is to be made with a standard A* algorithm that minimizes the path length. It is considered whether dangerous points are avoided and how much longer the route gets as a result. The calculation time is not considered in this comparison because the route only has to be calculated once when the planning on the Shared Guide Dog 4.0 is started. The following Figure 13 shows the starting point in blue and four different end points marked in red. The number in the marker indicates the route. This number is used later in the comparison and discussion to identify the route.

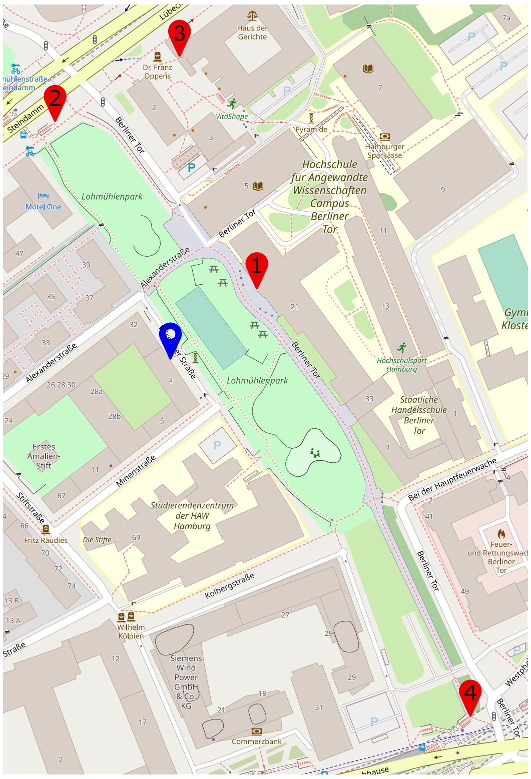

Figure 13. An overview of the four routes used for the comparison. The starting point is marked in blue, the end points in red.

The results for the four compared routes are shown in Table 3. The lowest percentage difference is 0.7% while the highest difference is 26.9%. The major difference is in the route used. For Route 3, the shortest route is over a road classified as 'residential'. With a used cost factor of 8.0, a detour is taken to avoid this road, as happened here. The small deviation of 0.7% shows that the algorithm uses the closest route, provided it meets the user's needs. The average percentage deviation is about 8.9%.

Table 3. Comparison of modified route planning to standard A*.

Route	Standard A* Length	Modified A* Length	Difference
Route 1	139.658	146.196	4.7%
Route 2	175.713	176.897	0.7%
Route 3	229.871	291.801	26.9%
Route 4	313.745	323.633	3.2%

Figure 14 shows all routes compared to the standard A*. Basically, it can be seen for all routes that a route on the right side of the path is selected, whereas the standard A* algorithm uses roads and crosses them.

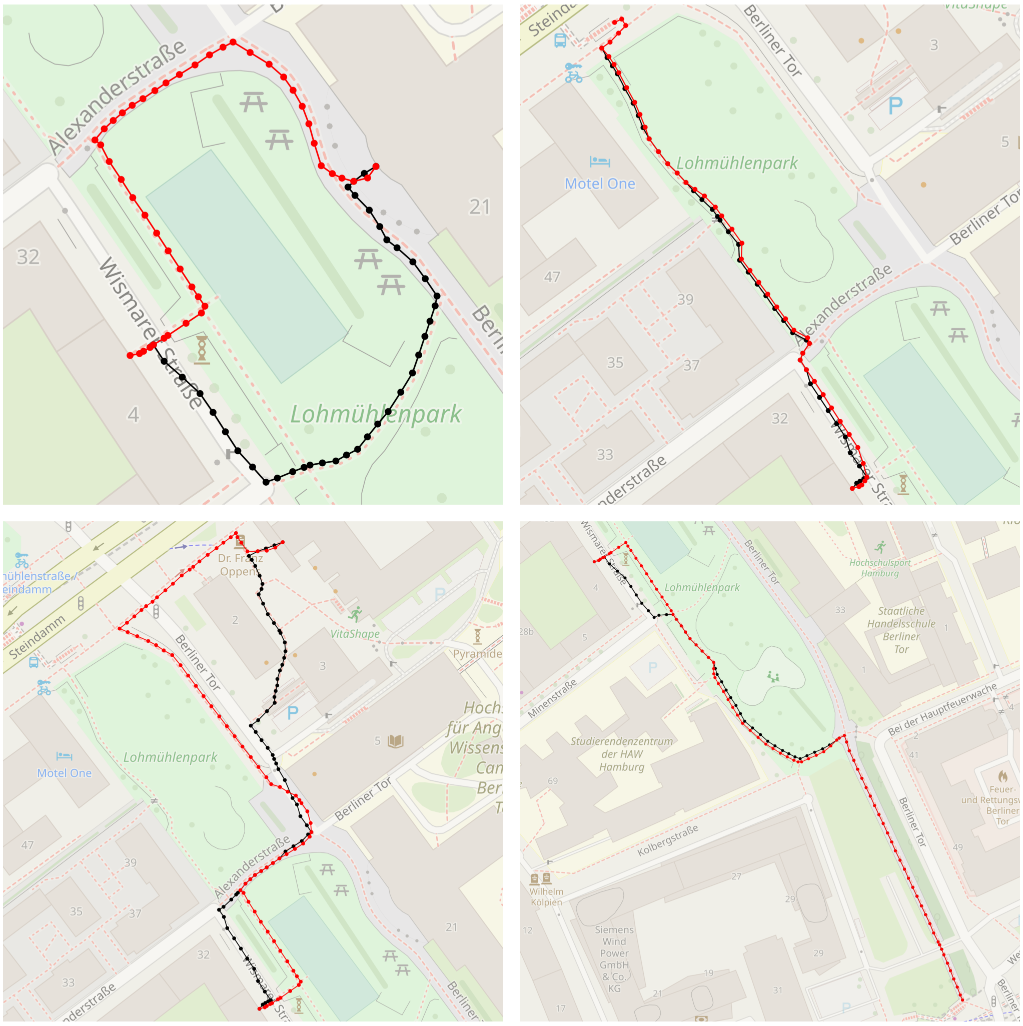

Figure 14. The result of the route calculation: Routes 1 to 4 from top left to bottom right. Routes in red are calculated with the presented algorithm.

5. Discussion

The comparison results from the previous section are critically reviewed in this section. The basic behavior of the presented solution is that a route will always be chosen on the right-hand side, is very reliable, and works well. However, the detailed consideration of a crossing, as in Figure 15, also shows a weakness. If there is enough space, the route is planned in a large arc at junctions. This route can be followed very well by the Shared Guide Dog 4.0. However, it is more difficult for blind people to follow in their minds than tight curves. Further adjustments to the cost function will be necessary in the future to fully meet the needs of users.

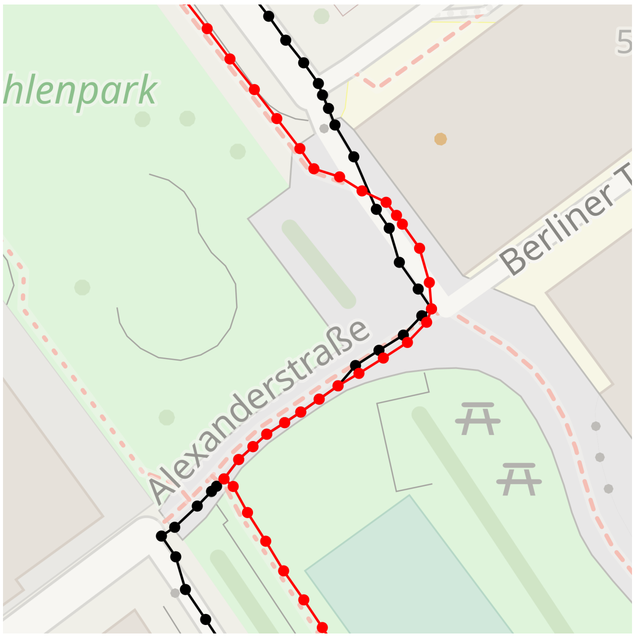

Figure 15. A detailed view of a junction with routes of standard A* (black) and the presented solution (red)

In the previous section, the path lengths between a standard A* and the solution presented in this paper were compared. The deviation was less than 10% in three of the cases studied. According to Opach et al. [21], the extension of a route for obstacle avoidance should be at most 10% to be perceived as a real alternative. One calculated route does not meet this criterion as it avoids the road. Here it should be further investigated whether short distances on the road would be a better solution than a too long detour.

6. Conclusions

This paper proposes a solution to safely navigate a mobile robot assistance system for the blind. Based on the information from OpenStreetMap, a cost function for an A* algorithm is derived, which can take into account pedestrian conditions, such as path and surface type, and different user requirements. Due to the flexible OpenStreetMap data format with freely assignable attributes and the adaptability of the costs for the attributes, the algorithm can be easily adapted to further requirements.

The tests showed that the presented path planning algorithm, consisting of the user function and the direction function, fulfills the requirements. The user function can be used very flexibly. For example, if the values for the surface are significantly higher than the values for the road type, they will be weighted more heavily. Setting the parameters for the direction function requires considerably more experience to achieve the desired results. For the Lohmühlenpark test area of the Shared Guide Dog 4.0, the parameter set presented in this article proved to be appropriate. In other areas, however, different values may give better results.

OpenStreetMap has proven to be a good basis for the development of a route-planning algorithm. Urban areas in particular have a high degree of completeness and accuracy. For sidewalk navigation, the OpenStreetMap data must be manually revised and augmented. In particular, the representation of places as areas and the missing sidewalks along many streets have to be corrected manually. In the future, however, machine learning algorithms are expected to provide increasing assistance.

Further Work

The presented algorithm works well in the test area Lohmühlenpark in Hamburg. The presented steps and algorithms are transferable to other areas. Therefore, it can be assumed that additional areas can be developed quickly. However, in order to ensure the safety and reliability of the route, an evaluation and possible adjustment of the cost factors is necessary.

The open system of OpenStreetMap and the cost allocations associated with the attributes allow for the inclusion of additional criteria in the route calculation in the future. This is also recommended to improve the routing solution and further adapt it to the needs of different user groups. In particular, curbs should be taken into account as they can be a major obstacle for autonomous vehicles.

The computation time of the algorithm was not considered in the previous comparison, but for a final implementation on the Shared Guide Dog 4.0 or any other navigation solution, it should also be included in the comparison. The test area considered is very limited in size and the longest path is not very long at 323 m. It needs to be checked whether the algorithm can compute a reasonable solution for longer paths and whether the computation time is acceptable.

During the development, close cooperation with blind people took place and the routing solution was developed based on the needs of this specific user group. The applicability of the solution to other user groups or vehicles should be investigated in the future. Due to the free assignment of factors to attributes, quick adaptability is basically given.

Author Contributions: Conceptualization, J.M. and P.S.; methodology, P.S. and J.M.; software, P.S.; validation, P.S.; resources, H.G.; writing—original draft preparation, P.S. and J.M.; writing—review and editing, J.M. and H.G.; visualization, J.M. and P.S.; supervision, H.G.; project administration, H.G.; funding acquisition, H.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the authors.

Acknowledgments: The authors thank the Faculty of Engineering and Computer Science of the Hamburg University of Applied Sciences for funding and the administrative support of the project Shared Guide Dog 4.0.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the writing of the manuscript or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

OSM	OpenStreetMap
SGD	Shared guide dog
WGS84	World geodetic system 1984

References

1. Bashkanov, O.; Seidel, M.; Yakymets, M.; Daupayev, N.; Sharonov, Y.; Assmann, T.; Schmidt, S.; Zug, S. Exploiting OpenStreetMap-Data for Outdoor Robotic Applications. In Proceedings of the 2019 IEEE International Symposium on Robotic and Sensors Environments (ROSE), Ottawa, ON, Canada, 17–18 June 2019 ; IEEE: Piscataway, NJ, USA, 2019; pp. 1–7. [CrossRef]
2. Floros, G.; van der Zander, B.; Leibe, B. OpenStreetSLAM: Global Vehicle Localization Using OpenStreetMaps. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013.
3. Hentschel, M.; Wagner, B. Autonomous robot navigation based on OpenStreetMap geodata. In Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Portugal, 19–22 September 2010; pp. 1645–1650. [CrossRef]
4. Suger, B.; Burgard, W. Global outer-urban navigation with OpenStreetMap. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1417–1422. [CrossRef]

5. Blinden- und Sehbehindertenverein Hamburg e.V. Home Page. Available online: https://www.bsvh.org/ (accessed on 13 April 2023).
6. de Verdier, K.; Ulla, E.; Löfgren, S.; Fernell, E. Children with blindness – major causes, developmental outcomes and implications for habilitation and educational support: A two-decade, Swedish population-based study. *Acta Ophthalmol.* **2018**, *96*, 295–300. [CrossRef] [PubMed]
7. Help People Who Are Blind or Partially Sighted—OrCam Home Page. 2022. Available online: https://www.orcam.com/en/ (accessed on 3 November 2022).
8. Routago Assist Home Page. Available online: https://www.routago.de/ (accessed on 3 November 2022).
9. HERE Technologies | The world's #1 location platform Home Page. Available online: https://www.here.com/ (accessed on 13 March 2023).
10. TomTom GPS-Traffic Alerts, Maps, & Apps Home Page. Available online: https://www.tomtom.com/en_us/navigation/ (accessed on 13 March 2023).
11. Hamburger Hochbahn AG—The HEAT Project. Available online: https://www.hochbahn.de/en/projects/the-heat-project (accessed on 13 April 2023).
12. Main Page—OpenStreetMap Wiki. Available online: https://wiki.openstreetmap.org/w/index.php?title=Main_Page&oldid=2301410 (accessed on 13 March 2023).
13. OpenStreetMap Stats. Available online: https://planet.openstreetmap.org/statistics/data_stats.html (accessed on 13 March 2023).
14. Organised Editing Teams—OpenStreetMap Wiki. Available online: https://wiki.openstreetmap.org/w/index.php?title=Category:Organised_Editing_Teams&oldid=2455286 (accessed on 13 March 2023).
15. Mobasheri, A.; Sun, Y.; Loos, L.; Ali, A.L. Are Crowdsourced Datasets Suitable for Specialized Routing Services? Case Study of OpenStreetMap for Routing of People with Limited Mobility. *Sustainability* **2017**, *9*, 997. [CrossRef]
16. Ding, C.; Wald, M.; Wills, G. A Survey of Open Accessibility Data. In Proceedings of the 11th Web for All Conference (W4A'14), New York, NY, USA, 7 April 2014. [CrossRef]
17. Neis, P.; Zielstra, D. Recent Developments and Future Trends in Volunteered Geographic Information Research: The Case of OpenStreetMap. *Future Internet* **2014**, *6*, 76–106. [CrossRef]
18. Fahnenschreiber, S.; Gündling, F.; Weihe, K. *Per Pedes Routing*; Technical Report; Technische Universität Darmstadt: Darmstadt, Germany, 2019.
19. Damian, I.; Ionita, A.D.; Anton, S.O. Community- and Data-Driven Services for Multi-Policy Pedestrian Routing. *Sensors* **2022**, *22*, 4515. [CrossRef] [PubMed]
20. Cohen, A.; Dalyot, S. Route planning for blind pedestrians using OpenStreetMap. *Environ. Plan. B Urban Anal. City Sci.* **2021**, *48*, 1511–1526. [CrossRef]
21. Opach, T.; Navarra, C.; Rød, J.K.; Neset, T.S. Pedestrian Routing and Perspectives: WayFinder's Route down the Lane—Come on with the Rain. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 365. [CrossRef]
22. Rösmann, C.; Hoffmann, F.; Bertram, T. Integrated online trajectory planning and optimization in distinctive topologies. *Robot. Auton. Syst.* **2017**, *88*, 142–153. .: 10.1016/j.robot.2016.11.007. [CrossRef]
23. Elements—OpenStreetMap Wiki. Available online: https://wiki.openstreetmap.org/w/index.php?title=Elements&oldid=2459386 (accessed on 13 March 2023).
24. JOSM—OpenStreetMap Wiki. Available online: https://wiki.openstreetmap.org/w/index.php?title=JOSM&oldid=2486812 (accessed on 13 March 2023).
25. Quality Assurance—OpenStreetMap Wiki. Available online: https://wiki.openstreetmap.org/w/index.php?title=Quality_assurance&oldid=2492612 (accessed on 13 April 2023).
26. OpenStreetMap Blog—The Best World Map for Accessibility. Available online: https://blog.openstreetmap.org/2020/12/02/the-best-world-map-for-accessibility/ (accessed on 13 April 2023).
27. OSM for the Blind. Available online: https://wiki.openstreetmap.org/wiki/OSM_for_the_blind (accessed on 3 November 2022).
28. *19157:2013*; Geographic Information—Data Quality. International Organization for Standardization: Geneva, Switzerland, 2013.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.