

Article

Using Crowdsourced Indoor Geodata for the Creation of a Three-Dimensional Indoor Routing Web Application

Marcus Goetz

GIScience Research Group, Institute of Geography, University of Heidelberg, Berliner Straße 48, 69120 Heidelberg, Germany; E-Mail: m.goetz@uni-heidelberg.de; Tel.: +49-6221-54-5501; Fax: +49-6221-54-4529

Received: 5 April 2012; in revised form: 4 May 2012 / Accepted: 29 May 2012 /

Published: 6 June 2012

Abstract: Routing services for outdoor areas are omnipresent and also three-dimensional (3D) visualization is quite common within this area. Recent research efforts are now trying to adapt well known outdoor routing services to complex indoor environments. However, most of the current indoor routing systems only focus on two-dimensional visualization, thus only one level can be depicted. Especially multi-level routes therefore lack visualization. Also, most of the (few) existing 3D indoor routing services utilize proprietary software or plugins, thus a widespread accessibility for those services by using common computers or mobile devices is not feasible. Therefore this paper describes the development of a web-based 3D routing system based on a new HTML extension. The visualization of rooms as well as the computed routes is realized with XML3D. Since this emerging technology is based on WebGL and will likely be integrated into the HTML5 standard, the developed system is already compatible with most common browsers such as Google Chrome or Firefox. Another key difference of the approach presented in this paper is that all utilized data is actually crowdsourced geodata from OpenStreetMap (OSM). Such data is collaboratively collected by both amateurs and professionals and can be used at no charge under the Open Data Commons Open Database License (ODbL). Our research combines user-generated geo content of the Web 2.0 with future Internet technology for the provision of a ubiquitously accessible 3D indoor routing application.

Keywords: 3D-GIS; 3D routing; crowdsourced geodata; future internet; indoor LBS; indoor routing; OpenStreetMap (OSM); XML3D

1. Introduction

With the increasing size and complexity of public buildings or institutions, such as universities, hotels, airports or office buildings, there is also an increasing need for mature indoor location based services (Indoor LBS), such as indoor maps or indoor routing [1]. Both research and economy entities recently became aware of this trend. Global companies such as Google, Navteq or Bing started to extend their well-known outdoor map applications towards indoors [2–4]; however until now they have mainly focused on the two-dimensional visualization of building representation. For simple indoor map visualization this approach seems to be feasible. In contrast, the visualization of three-dimensional (3D) routes between several floors in a building with a two-dimensional representation can always only provide a brief overview, but not properly visualize vertical parts (*i.e.*, transferring from one floor to another) of a route. Essentially, 2D environments have revealed serious limitations when considering multi-level structures [5]. In contrast, a 3D map or model allows a better presentation of the provided information. It has been proven that 3D city models are advantageous for (mobile) navigation [6] and the actual need for 3D information has been elaborated [7].

Widespread outdoor applications and newly developed indoor applications are typically based on commercially provided data, which has been collected by professionals such as surveyors, architects or commercial data providers. However, in most cases this data is proprietary and licensed by the corresponding data provider, thus the development of own applications with such data is limited according to the individual licenses. Also, commercially collected data can be quite expensive to acquire. In contrast, within the last years a new source for geodata, namely crowdsourced geodata or volunteered geographic information (VGI), has evolved. Thereby, a community (normally aiming for some distinct goal) collaboratively collects geodata and provides it in the Internet. That is, VGI can be regarded as a spatially enriched type of user-generated content (UGC), a typical trend of the future Internet [8]. Similar to the UGC phenomenon of the Web 2.0, everybody is able to use VGI for different applications at no charge. Regarding the increasing availability of GPS-enabled devices, such as cameras or smart phones, there is an enormous potential arising from “*billions of humans acting as remote sensors*” [9]. Regarding the quality and quantity of the data, several evaluations of OpenStreetMap (OSM, one of the most prominent examples of VGI) have revealed that OSM is, especially in urban areas, able to compete against commercially collected geodata [10–12]. The usage of OSM data is free and open under the Open Data Commons Open Database License (ODbL). Trying to utilize the power of VGI, researchers have attempted to extend such communities to indoor spaces. Especially for OSM there is a very detailed *IndoorOSM* mapping proposal available [13], which initially originated from research on the demands and requirements of crowdsourced indoor geodata [14].

The main contribution of this paper is the development of a web-based 3D application, capable of both visualization of building interiors as well as computing and depicting (multi-level) routes inside a building. Essentially, the developed application utilizes existing (and future) HTML standards; the application runs on most modern browsers as well as on some mobile devices. A method for the automatic extraction of building interiors from crowdsourced geodata of OSM is presented, whereby this information can be used for automatically creating XML3D geometries. XML3D is an emerging technology based on the Extensible Markup Language (XML) for the declarative generation of 3D

content in the browser [15]. The route computation itself is based on the Dijkstra algorithm [16], whereas the Open Source implementation *pgrouting* for PostgreSQL databases has been utilized. A detailed route graph is created and all possible routes inside the building are pre-processed and stored as individual XML3D tiles on the server. The aim of the work presented within this paper can be summarized as: (1) to demonstrate the powerful application of crowdsourced geodata from OSM for the creation of (3D) indoor LBS; and (2) to present a web-based 3D routing system and its advantages in the context of urban environments by using future Internet technology.

The paper is organized in seven sections: after the introduction, Section 2 provides a brief overview about existing (3D) indoor applications and Section 3 then introduces the emerging technology of XML3D. Section 4 provides some general introduction of OSM, as well as detailed insight into crowdsourced indoor information with *IndoorOSM*. Thereafter, Section 5 describes the system architecture as well as the different algorithms that have been developed. Section 6 then describes the web front-end which allows an easy perception and understanding of indoor information as well as the functionality of indoor route computation. Section 7 finally concludes the conducted research and provides an outlook on future work.

2. Related Work

One of the first approaches towards an indoor navigation system was the so called *Cyberguide* system [17]. It is designed as a tool for guiding tourists through both outdoor and indoor environments. When entering a new room inside a building, the system indicates this by displaying an arrow on a two-dimensional map.

Similarly, the *MARS* System [18] provides information about buildings in the Columbia university campus to the users (e.g., visitors, staff, students, *etc.*). It is a collaborative system in which users can annotate obstacles or buildings and share this information via a desktop or augmented reality. A similar system for the *École Polytechnique Fédérale de Lausanne* (EPFL) is presented in [19], whereas a two-dimensional map is utilized for visualization.

The user-oriented development of a nomadic exhibition guide for trade fair visitors is described in [20]. Within the *SAiMotion* project, the authors developed a system which supports planning at home as well as mobile guidance on-site—both in a 2D map. A similar prototype for an exhibition show guiding application is also presented in [21].

One of the first 3D applications is described in [5]. As a prototypical system the authors utilized Oracle Spatial 10 g (with a mixed model of geometry and graph) as DBMS and the building of the Aerospace Faculty of TU Delft (with 13 floors) as a test building. The graph contains about 1200 elements.

An indoor navigation system, namely *iNAV*, is presented in [22]. It uses the A* algorithm for real-time routing and navigation. The Java client is suitable for laptops or proper PDA devices. The system consists of several distributed web services. An initial trial by the authors has revealed a lack of performance due to service communication. A similar approach with client-server architecture based on PHP (*i.e.*, programming language for dynamic web applications, namely Hypertext Preprocessor) web services and KML (a XML-based markup language for Geodata, namely Keyhole Markup

Language) models is also presented in [23]. Similar mobile solutions (mostly in 2D) are furthermore available [24–26].

In [27] the development of an indoor navigation system which considers users' special needs and preferences, is described. It aims at the provision of universal indoor LBS for everyone and therefore tries to support several types of users by exploiting multimodal interaction. Unfortunately the path algorithm used for route calculation “*can only be applied to connected two-dimensional digraphs, which means the route can be computed only if the O and D are located on a connected single floor*” [28]. Another 2D indoor navigation framework is also presented in [29,30].

Besides those research-motivated indoor routing systems, there are also some web-based indoor maps developed by global companies such as Google Indoor Maps [2], NAVTEQ Destination Map [4], or Bing Maps Venue maps [3], but these only utilize two-dimensional data for the visualization.

To conclude this brief review, it becomes apparent that both research and economy develop different solutions for indoor maps and routing. However, existing solutions are mainly two-dimensional, although it has been demonstrated that a 3D visualization is advantageous [6]. Furthermore, existing 3D applications require additional software or plugins, thus a broad application on standard browsers is not possible. That is, users have to download and install additional software, representing a barrier for the utilization of the service. However, regarding the future Internet, a ubiquitous accessibility of such standard-based services is desirable. Regarding the data of the above mentioned applications, it becomes obvious that they all utilize proprietary data. In most cases license fees have to be paid for those, because they have typically a cost recover model for the data acquisition. Essentially, as far as the author is aware, there is no 3D indoor map and routing application available, which utilizes crowdsourced geodata.

3. Creating 3D Web Applications with XML3D

Web technologies are omnipresent; not only for distributing digital information in the Internet, but also for forming the base of a ubiquitous application platform. Thereby, the content itself rapidly evolved from basic text data without any styling, over dynamic and attractive web pages including styling and images, to sophisticated web applications with heterogeneous data sources and multimedia elements such as videos or live communication features. Motivated by the fact that we are living in a 3D world and that more and more communication media is shifting towards 3D visualization, web developers always aimed at creating 3D content. However, the main ideas in this area mainly comprise the integration of additional 3D content which only can be visualized when additional software, such as Java3D or VRML viewers, are installed. A platform-independent and portable solution has not yet been developed so far, but in the last two years—trying to push 3D to the future Internet—different initiatives and ideas for creating a quasi-3D-standard for the web has been conducted.

One promising possibility for the creation of 3D web content is XML3D. It is an extension of HTML5 which allows for the creation of interactive 3D graphics. The specification of XML3D has been developed together by the German Research Center for Artificial Intelligence (DFKI), the Intel Visual Computing Institute (IVCI) and the Saarland University. These are furthermore the lead institutions driving the development of XML3D as well as the efforts for making XML3D an integral part of HTML5, thus the common standard for 3D web content.

XML3D is aiming at a maximum compatibility with HTML5 as well as XHTML. For the generation of graphical objects, XML3D mainly features triangles which can then be utilized for describing nearly any arbitrary shape [31]. Following an declarative approach “XML3D fully leverages existing web technologies including HTML, Cascading Style Sheets (CSS), the Document Object Model (DOM), and AJAX for dynamic content” [15]. That is, all 3D elements are part of the DOM, thus common DOM scripting and events, such as deletion of objects or on-demand integration of additional content, are fully supported. In particular, XML3D aims at enabling programmers to create 3D content without actually having any specific knowledge about certain 3D technologies [31]. That is, basically every web designer who is familiar with common World Wide Web Consortium (W3C) standards such as (X)HTML, CSS, *etc.* is able to create and integrate 3D elements into a web application. This is also regarded as one of the main differences between XML3D and other approaches like WebGL, O3D or X3DOM [31]. The possibilities and feasibility of XML3D have already been demonstrated by integrating XML3D support into WebKits and Mozilla browsers, as well as with a portable WebGL/JavaScript based version [15]. Most of the latest existing browsers such as Firefox or Chrome are already able to visualize XML3D content, thus applications which are based on this technology can already be utilized by the broad public. The Microsoft Internet Explorer does however not yet support any kind of 3D content. For more information on XML3D please refer to the publication [15] as well as the current specification version 0.4 [32].

4. OSM as a Source for Crowdsourced (Indoor) Geodata

Combining the idea of user generated content (UGC) in the Web 2.0 with the ubiquitous availability of GPS-enabled devices, such as smartphones or cameras, an enormous potential of crowdsourced geodata arises. Both amateurs and professionals collect not only content, but spatially references content, such as geo-tagged Flickr photos or street information (e.g., in FixMyStreet.com).

One of the most prominent and most varied source for crowdsourced geodata is OSM. Initially, OSM aimed at the creation of a free world map, but soon the project turned beyond that, becoming more like a free global database comprising different kinds of geodata. Essentially, everybody can contribute, change and improve the data in OSM and the project benefits from currently more than 500,000 registered users [33]. The users can contribute two different types of data: (1) two-dimensional geometries and (2) additional (semantic) information. For the provision of geometries, users basically provide geo-tagged points (*nodes* in OSM terminology). Currently there are about 1.42 billion *nodes* inside the database (by end of March, based on our internal OSM database). Additionally, several nodes can be combined into so called *ways*. These can be utilized for either describing linestring geometries or polygons whereas in the former case the way is not closed and in the latter case the way is closed (*i.e.*, the last node equals the first node). The current OSM database contains about 131 million *ways*. Furthermore, OSM provides so called *relations* (currently about 1.35 million) which can be utilized for mapping complex polygons with holes or for describing relationships between different OSM features (e.g., turn restrictions, routes, *etc.*).

Besides those geometries, users can furthermore tag them, thus provide additional information about them. OSM provides an open key-value pair methodology which allows the contributors to describe a wide range of information and attribution. The key describes a special kind of information

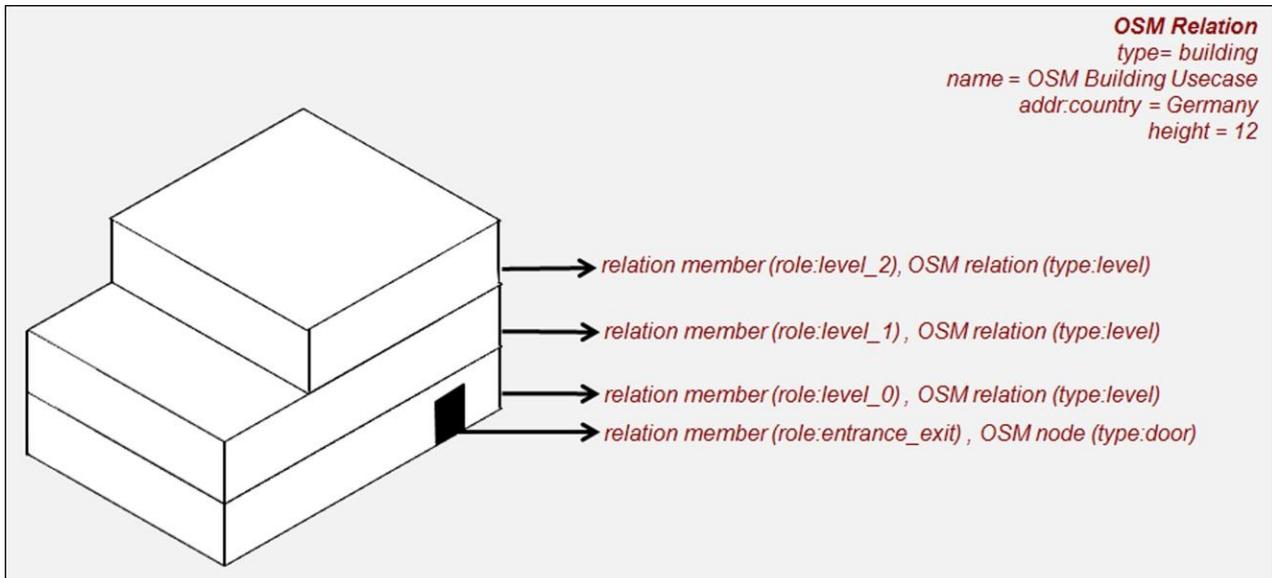
domain or characteristic of the map feature (e.g., *building, natural, etc.*) and the value refines this information (e.g., *university, forest, etc.*). There are some community accepted tags which are listed at Watchlist [34], but basically any kind of key/value can be utilized. A complete list of all currently used tags with some exemplary values is available on Tagwatch [35]. Also, the OSM wiki provides detailed descriptions about the different keys and values [36].

Convinced by the increasing demand for indoor data, OSM tries to provide information about indoor spaces, although there is no standard indoor mapping schema available yet. However, a promising and scientifically motivated mapping proposal, namely *IndoorOSM*, has been developed [14] and presented [13]. The application of collaboratively collected *IndoorOSM* data has already been demonstrated with a web-based two-dimensional indoor map with indoor routing capabilities [37]; however an application which utilizes 3D data of *IndoorOSM* has not been developed yet.

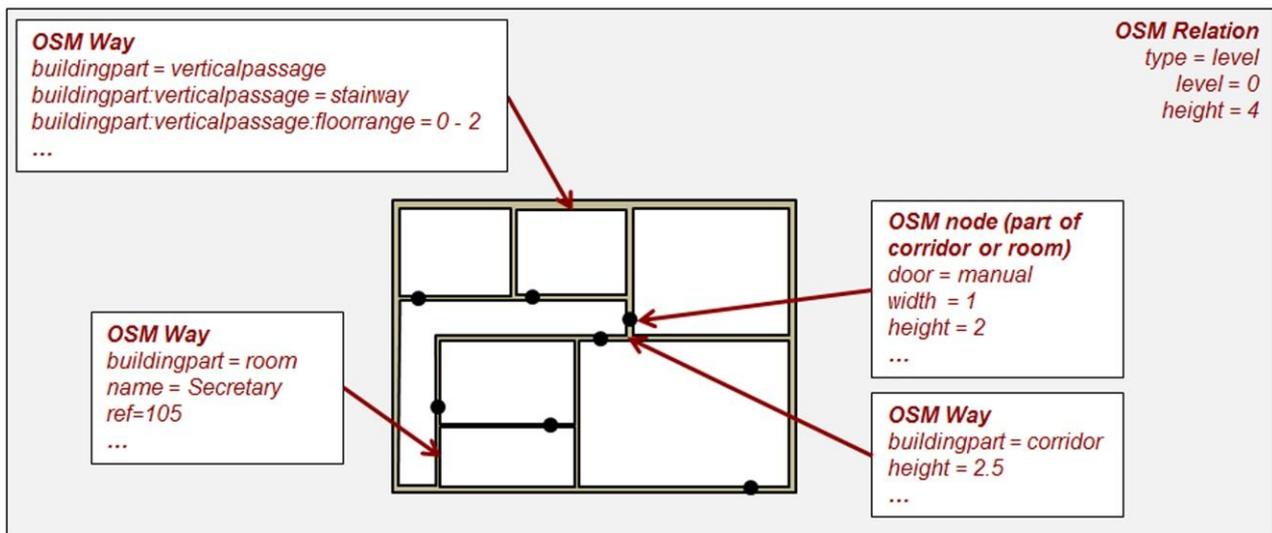
The *IndoorOSM* model can be described as follows: a building is considered as a hierarchically structured object which consists of several levels/floors. These floors are furthermore divided into elements such as rooms, corridors, stairways, doors or windows. Within OSM, a building is therefore mapped as a relation (main-relation). All the different floors are also mapped as *relations* (level-relation), whereby these are considered as relation members (children) of the main-relation. Furthermore, the different building parts (rooms, *etc.*) are also mapped as relation members of the corresponding level-relation. The *IndoorOSM* proposal aims at providing detailed floorplan information, thus all building parts are typically mapped as closed *ways*, representing the polygonal footprint of the corresponding building part. By tagging the different OSM-features with the key *height*, it is furthermore possible to provide 3D information, such as the height of a room. Information about doors is added by using OSM-*nodes* and the width or height of the door can be provided with the corresponding key-value pairs. The basic ideas of the *IndoorOSM* mapping schema are also depicted in Figure 1, whereby Figure 1(a) depicts the basic hierarchical principle of a building and Figure 1(b) shows the composition of a detailed floorplan for an exemplary building level (in this case level 0, *i.e.*, the ground level). Since it is only possible to map two-dimensional geometries in OSM, vertical connections, such as stairways, escalators or elevators, need to be expressed semantically with keys and values. In general, every vertical passage is mapped as a polygon (*i.e.*, a closed way) and tagged with *buildingpart=verticalpassage*. That is, for each elevator funnel, stairway area or escalator platform, there is a polygon on the corresponding floor. Furthermore, *IndoorOSM* provides the key *buildingpart:verticalpassage* for refining the corresponding type. Exemplary values are *stairway, escalator, elevator, ramp, etc.* The general floor range of a vertical passage is provided by using *buildingpart:verticalpassage:floorrange*, so a user can describe for example that an elevator ranges from the ground level to the second level (value 0–2). Since all of these vertical passage polygons are accessible via another building part (e.g., a corridor next to the elevator), the *IndoorOSM* model provides a corresponding door (or opening) node between the two adjacent elements. By adding the key *connector:ids* to these nodes, it is possible to semantically describe the connectivity between vertically connected elements. The value of this key is the OSM-ID of the element to which a vertical connection exists. For a one-way connector (e.g., an escalator), only the starting node is tagged with *connector:ids*, whereas for multi-way connectors (e.g., stairways or escalators), both adjacent *nodes* are tagged accordingly. However, the key *connector:ids* only contains information about the directly

adjacent neighbors. For more information on *IndoorOSM* please refer to the underlying research publication [14] or the *IndoorOSM* wiki page [13].

Figure 1. The basic hierarchical principle of a complete building (a) and the detailed floorplan information for an exemplary building floor/level (b) in *IndoorOSM*.



(a)



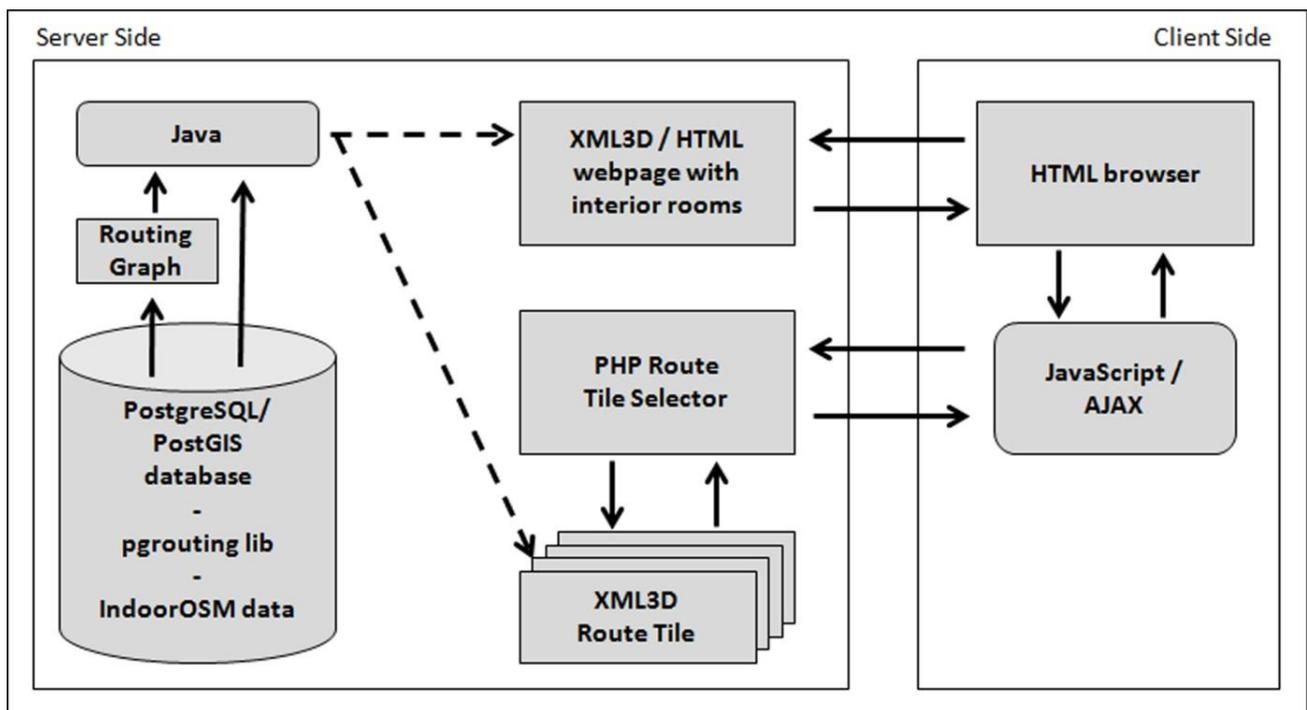
(b)

5. System Architecture for the Generation and Utilization of the 3D Indoor Routing Service

The system architecture of the web application is divided into the server side and the client side (depicted in Figure 2). On the server side, a PostgreSQL database with PostGIS extension serves as a data container for the *IndoorOSM* data. Therefore, the raw OSM data, which can be obtained from either the OSM web page or on several OSM data pages like Geofabrik [38], is imported by using the Open Source command line tool *OSMOSIS* [39]. Furthermore, the server side consists of the C++ *pgrouting* libraries. *Pgrouting* provides shortest route computation capabilities, whereas the computation is based on the wide-known Dijkstra algorithm [16]. Additionally, the system architecture consists of

some other algorithms for the automated creation of a XML3D/HTML webpage which visualizes the interior rooms of the desired buildings and additionally allows for the control of the visibility of different floors as well as the computation and visualization of routes. Those algorithms are based on Java. The different routes inside the desired building are pre-computed and stored as individual route tiles on the web server. On the client side, the web application can be visualized in any XML3D-capable web browsers. By using JavaScript and AJAX technology, it is possible to load and visualize the pre-computed route tiles. The server side of the developed application and the required algorithms are described in great detail in the following sub-sections, whereas the client application (from a user’s perspective) is elaborated in Section 6.

Figure 2. System architecture and processing workflow of the 3D indoor routing service.



5.1. Generating the Routing Graph

For the computation of meaningful shortest routes within the building, different building elements such as rooms, corridors and doors have to be modeled. For this purpose, a comprehensive route graph has been automatically created. Thereby, the methodology of the graph is based on the *Weighted Indoor Routing Graph* (WIRG) [40], thus all doors are represented as nodes in the graph and edges represent connections between adjacent doors. The doors of the building can be automatically retrieved from OSM, because they are mapped as OSM-nodes. Vertical connections, such as stairways or elevators, are represented as edges in the graph and can be automatically retrieved from different OSM keys, such as *connector:ids* or *buildingpart:verticalpassage*, etc. (cf. Section 4). By computing the centerline of a corridor and adding vertical edges from the doors to this centerline, it is possible to model corridors (cf. [40]). The complete algorithm for the automated routing graph generation is briefly described in Figure 3 as pseudo-code.

Figure 3. Pseudo-code algorithm for the automated generation of a WIRG from *IndoorOSM*.

Algorithm createWIRG(DB, bID)

Input: DB = Database connection to a database with IndoorOSM data

Input: bID = the unique identifier for the desired building

```

1: WIRG ← empty
2: building ← getBuilding(DB, bID)
3: num_levels ← getNumLevels(building)
4: levels[num_levels] ← getLevels(building)
5: for each levels as level
6:   num_doors ← getNumDoors(level)
7:   doors[num_doors] ← getDoors(level)
8:   for each doors as door
9:     addNode(WIRG, door)
10:  num_edges ← getNumEdgesOfAdjacentRooms(level)           // edges between adjacent rooms
11:  edges[num_edges] ← getEdgesOfAdjacentRooms(level)
12:  for each edges as edge
13:    weight ← computeWeight(edge)
14:    addEdge (WIRG, edge, weight)
15:  num_corridors ← getNumCorridors(level)
16:  corridors[num_corridors] ← getCorridors(level)
17:  for each corridors as corridor
18:    centerline ← computeCenterLine(corridor, doors)
19:    centerlineNodes ← getNodes(centerline)
20:    centerlineEdges ← geEdges(centerline)
21:    for each centerlineNodes as centerlineNode
22:      addNode(WIRG, centerlineNode)
23:    for each centerlineEdges as centerlineEdge
24:      weight ← computeWeight(edge)
25:      addEdge(WIRG, centerlineEdge, weight)
26: num_verticalConns ← getNumVerticalConns(building)
27: verticalConns[num_verticalConns] ← getVerticalConns(building)
28: for each verticalConns as verticalConn
29:   addEdge(WIRG, verticalConns)

```

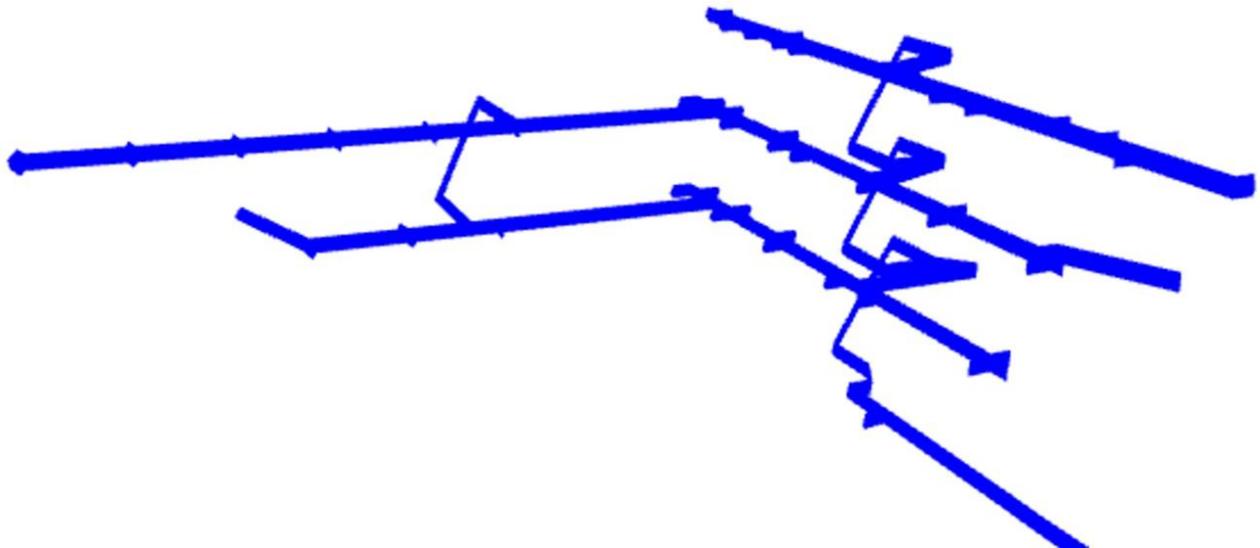
Output: Weighted Indoor Routing Graph (WIRG)

After the extraction and generation of the routing graph, it is transferred to the PostgreSQL/PostGIS database. The network is stored in one physical database table. It consists of link IDs, source and target of the corresponding edges, the 3D coordinates (both combined and separated into x, y, z) and the 3D length of the edge which is used as the cost parameter for Dijkstra. An excerpt of the route graph database table is provided in Figure 4(a) and the complete 3D route graph is visualized in Figure 4(b).

Figure 4. Route graph with x, y, z coordinates of the source and target nodes with connectivity and oneway information (a) and a complete 3D route graph for a sample building (b).

	id serial	source integer	target integer	cost double precis	reverse_cost double precis	oneway text	x1 double precis	y1 double precis	z1 dou	x2 double precis	y2 double precis	z2 dou	gid [PK]	the_geom geometry	length double precis
49	49	224311	68	6.947277037	6.947277037	N	8.6771015	49.4185436	-4	8.677101111	49.41855053	-4	49	010500002	6.947277037
50	50	68	224613	6.970561306	6.970561306	N	8.677101111	49.41855053	-4	8.6771008	49.4185575	-4	50	010500002	6.970561306
51	51	68	69	5.134246619	5.134246619	N	8.677101111	49.41855053	-4	8.677152453	49.41855043	-4	51	010500002	5.134246619
52	52	225563	69	6.878086266	6.878086266	N	8.6771532	49.4185436	-4	8.677152453	49.41855043	-4	52	010500002	6.878086266
53	53	69	228595	7.156125461	7.156125461	N	8.677152453	49.41855043	-4	8.6771513	49.4185575	-4	53	010500002	7.156125461
54	54	73	226822	2.094211873	2.094211873	N	8.676757347	49.41861475	-4	8.6767572	49.4186357	-4	54	010500002	2.094211873
55	55	224568	49	1.104752449	1.104752449	N	8.6767691	49.4187554	-4	8.676780147	49.41875531	-4	55	010500002	1.104752449
56	56	225818	226369	5.397795846	5.397795846	N	8.6767699	49.4187137	0	8.676767	49.4187676	0	56	010500002	5.397795846
57	57	226369	227132	4.874935897	4.874935897	N	8.676767	49.4187676	0	8.6767803	49.4187207	0	57	010500002	4.874935897
58	58	227132	35	6.936567049	6.936567049	N	8.6767803	49.4187207	0	8.676780345	49.41871376	0	58	010500002	6.936567049
59	59	225818	35	1.044522120	1.044522120	N	8.6767699	49.4187137	0	8.676780345	49.41871376	0	59	010500002	1.044522120
60	60	35	226756	1.186300263	1.186300263	N	8.676780345	49.41871376	0	8.6767922	49.4187142	0	60	010500002	1.186300263
61	61	35	36	5.497793566	5.497793566	N	8.676780345	49.41871376	0	8.676780419	49.41865878	0	61	010500002	5.497793566
62	62	36	227063	1.168812334	1.168812334	N	8.676780419	49.41865878	0	8.6767921	49.4186592	0	62	010500002	1.168812334
63	63	36	37	9.794131010	9.794131010	N	8.676780419	49.41865878	0	8.676780518	49.41864899	0	63	010500002	9.794131010
64	64	37	226685	1.411814968	1.411814968	N	8.676780518	49.41864899	0	8.6767664	49.418649	0	64	010500002	1.411814968

(a)



(b)

5.2. Automated Generation of an 3D Indoor Model for XML3D

Aiming at an easy and fast generation of 3D indoor models for different buildings from *IndoorOSM* information, an automatic generation is required. Therefore, information from OSM about the desired building (which is mapped according to the *IndoorOSM* mapping proposal) needs to be gathered. Thereby, the real world coordinates of OSM with latitude and longitude information must be transformed into some kind of local reference system (e.g., based on the building bounding box), because XML3D is currently not able to handle double precision coordinate values.

The algorithm is depicted as pseudo-code in Figure 5 and a human-readable explanation is given in the following: The algorithm sequentially retrieves all the different levels (floors) of the building. For each room on the corresponding floor, a XML3D geometry is created in that way that all corresponding walls are created as well as the floor and ceiling geometry. These faces are then triangulated, because XML3D is currently only capable of dealing with triangle geometries. The respective room elevation is based on the current level elevation, which can be calculated from all the other levels. For example,

when computing the level elevation of floor 2, the elevations of level 0 and level 1 are accumulated (based on the OSM key *height*). The elevation of the ceiling of a room is the level elevation plus the height of the corresponding room. After computing all the room geometries, they are added to the XML3D indoor model. Being able to hide different levels in the web application (cf. also next chapter), all rooms are grouped, according to their level, in the DOM of the web application. In doing so, it is possible to set the visibility of the different groups in the DOM via JavaScript and CSS on-demand during runtime.

Figure 5. Pseudo-code algorithm for the generation of a XML3D indoor model from *IndoorOSM*.

Algorithm createIndoorModel(DB, bID)

Input: DB = Database connection to a database with IndoorOSM data

Input: bID = the unique identifier for the desired building

```

1: 3DIM ← empty
2: addHeaderInfo(3DIM)
3: building ← getBuilding(DB, bID)
4: bounds ← computeBounds(building)
5: num_levels ← getNumLevels(building)
6: levels[num_levels] ← getLevels(building)
7: for each levels as level
8:   num_rooms ← getNumRooms(level)
9:   rooms[num_rooms] ← getRooms(level)
10:  level_height ← getLevelHeight(level)
11:  level_elevation ← getLevelElevation(level)
12:  for each rooms as room
13:    num_walls ← getNumWalls(room)
14:    walls[num_walls] ← getWalls(room)
15:    room_height ← getHeight(room)
16:    for each walls as wall
17:      wall ← computeAndTriangulateWall(wall, room_height)
18:      wall ← computeAndTriangulateFloor(room, room_elevation)
19:      ceiling ← computeAndTriangulateCeiling(room, room_elevation, room_height)
20:      transformCoordinatesForXML3D(walls, bounds)
21:      transformCoordinatesForXML3D (floor, bounds)
22:      transformCoordinatesForXML3D (ceiling, bounds)
23:      add(3DIM, walls)
24:      add(3DIM, floor)
25:      add(3DIM, ceiling)
26: addControls(3DIM)

```

Output: XML3D Indoor Model (3DIM)

6. The XML3D-based Web Application

To allow for the routing analysis and the 3D visualization of both the interior building parts as well as the routes, a web site which can be accessed from any XML3D-capable computer or mobile device,

is created (Figure 6). A predefined list enables the user to select the source and target of the desired route. The level overview on the right-hand side (cf. Figure 6) can be utilized for displaying or hiding the different floors. This enables the user to only visualize the desired floors, thus the visualization can be adapted to the user’s requirements. By clicking on *Computer Route*, the optimal path between the user defined entities is selected from the pre-computed XML3D route tiles and visualized in 3D in the routing application. Essentially, the pre-computation of all routes allows an efficient provision of any desired route in $O(1)$ on the user’s device. Figure 7 depicts an exemplary 3D route from the entrance of a building to an office on the second floor. For visualization purposes, the other floors (the basement and level 1) are hidden. By clicking on *Clear Route* the visualized route is removed from the DOM.

Figure 6. Indoor routing web application with XML3D scene graph.

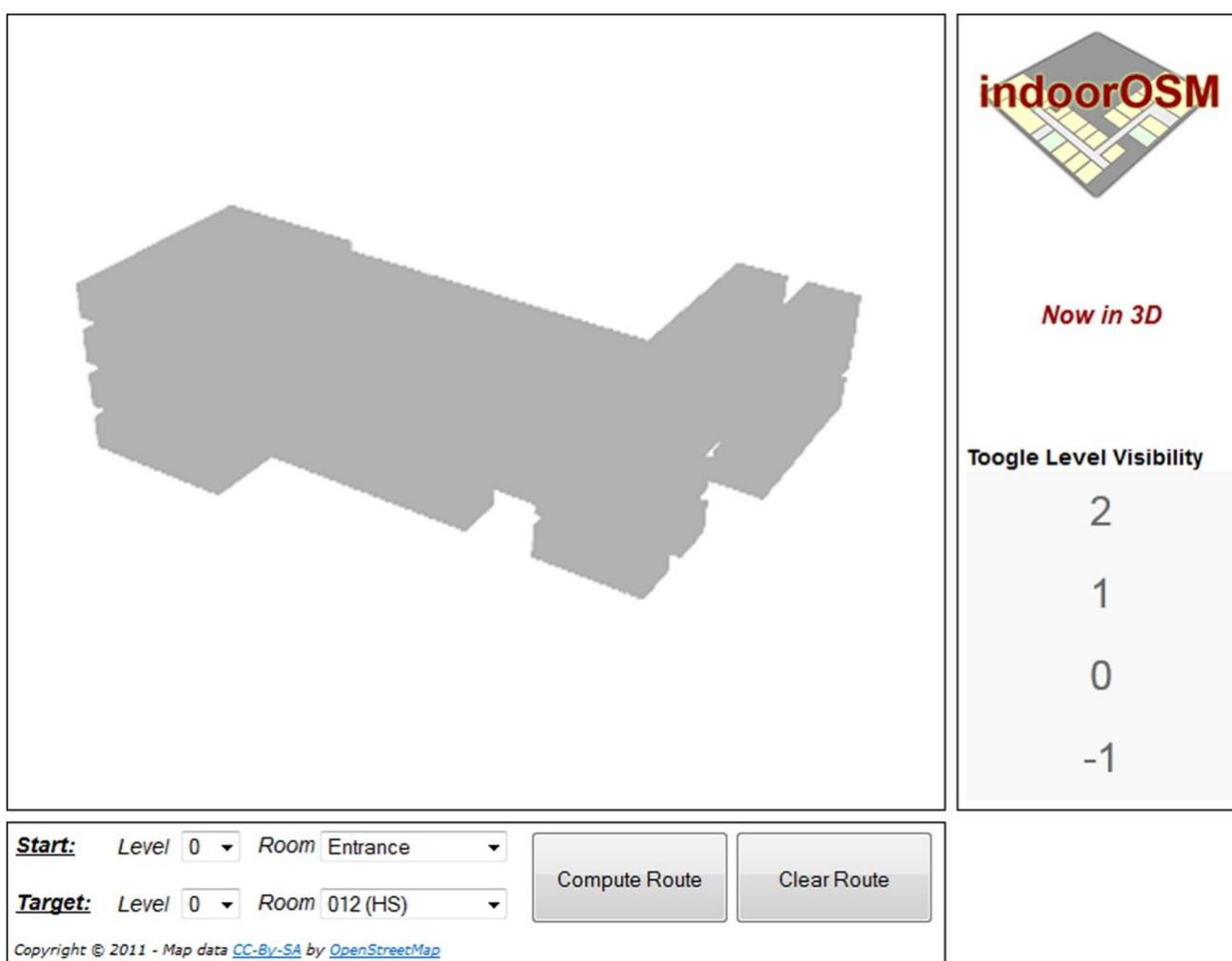
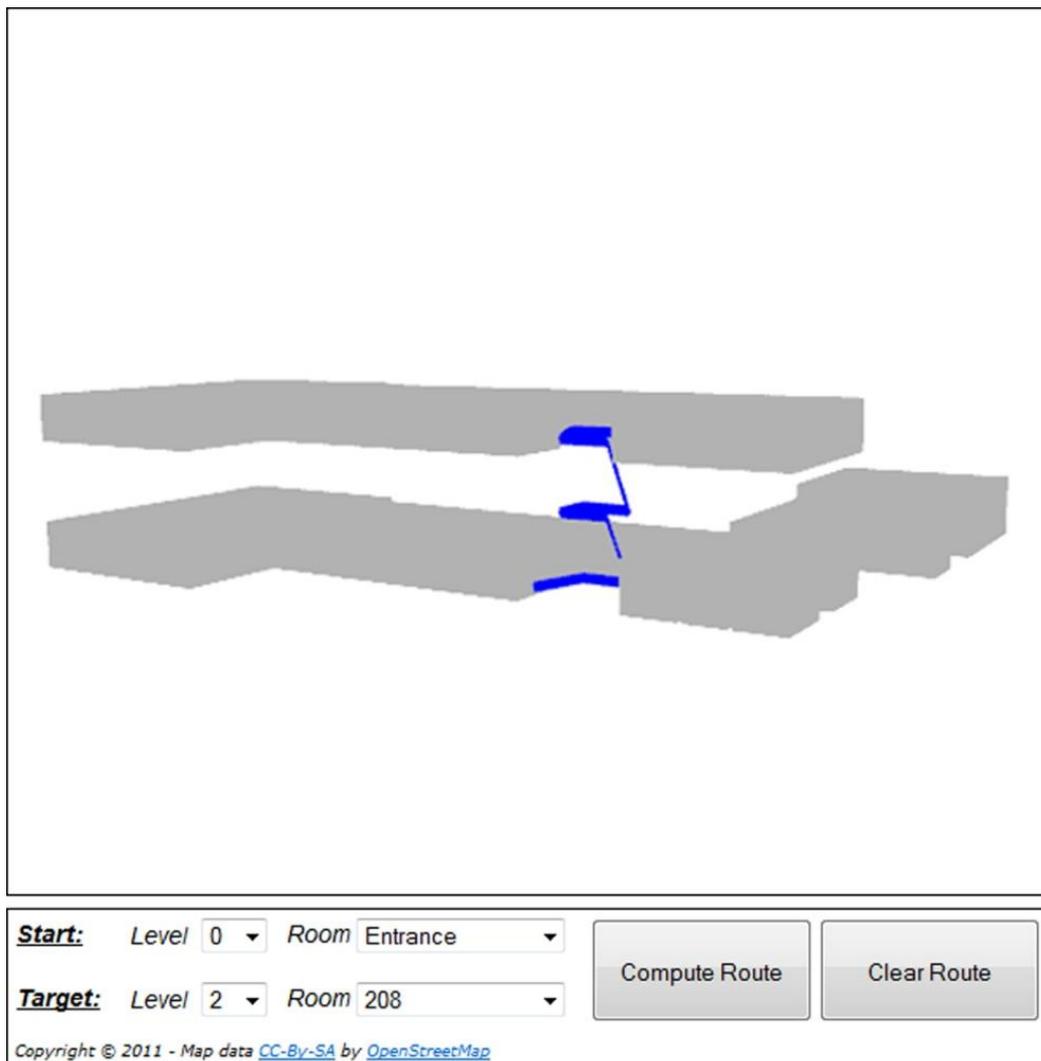


Figure 7. 3D visualization for a computed route between user-defined start and target inside the building.



For realizing the on-demand loading and removing of the corresponding route tile, Asynchronous JavaScript and XML (AJAX) technology is utilized (cf. Figure 2). When clicking the *Compute Route* button, an AJAX script retrieves the desired start and target node and sends them via HTTP-GET to a PHP application. This then checks if there is a shortest path between the desired nodes and if so, returns the XML3D code from the corresponding route tile file. The AJAX script parses the PHP response and, in the case of an existing route, appends the XML3D to the DOM. In contrast, when clicking the *Clear Route* button, a different AJAX script searches for every existing route tile in the scene graph and removes them from the DOM.

7. Conclusion and Future Work

Indoor LBS and especially indoor routing are gaining attraction in research efforts as well as commercial application. Contrary to existing approaches, a 3D visualization is desirable, because it allows a better perception and understanding of the provided information (e.g., the computed route) for the users. When developing 3D indoor applications, it is furthermore important that they are broadly

accessible, thus that common browsers can visualize 3D indoor information. That is, for reducing the barrier of using the application, no additional software or plugin should be required. Furthermore, crowdsourced indoor geodata seems to a promising data source for detailed information about indoor spaces.

Due to the pre-computation of routes, the approach presented here suits well to static scenarios, in which routes are always available and room settings do not change. However, in its current state, it cannot be directly adapted to scenarios which are rather dynamic, such as emergency situations or simulations, *etc.*, in which certain routings may quickly become invalid. That is, for the consideration of such dynamic scenarios, it is required to not only provide pre-computed routes, but also on-demand route computation functionality.

It can be concluded that it is already feasible to create 3D indoor routing applications which are purely based on crowdsourced indoor geodata from OSM. In particular, it has been proven that 3D indoor routing services can be implemented by using XML3D, thus that 3D indoor LBS can be used without any specific software or prerequisite (except the XML3D-enabled browser). Essentially, this allows the provision of 3D applications in the context of future Internet by utilizing cutting edge Internet technology. As described beforehand, the conducted research has been based on crowdsourced geodata. However, the general architecture as well as the client itself is also suitable for other data sources. In general, any other (proprietary) data source, such as CAD plans, evacuation plans, Industry Foundation Classes (IFC) models from the Building Information Modeling (BIM) domain or 3D modelling software (e.g., Cinema 4D or 3D Studio Max), can be utilized for the generation of the here presented XML3D based indoor routing application. In this research we have shown the potential of a crowdsourcing approach through OSM, because the authors believe that this is an important and rich source for open and free geodata.

Future work will include the incorporation of several buildings. This will allow queries like “*What is the shortest route between office 321 on the 3rd floor of building A to the lecture room in the basement of building B*”. Also, different usage modes such as examination or walk-around shall be included, because this is likely to increase the user experience. Additionally, the inclusion of access restrictions and different states of spaces and doors has to be considered in both the *IndoorOSM* model and the presented application. Furthermore, the XML3D technology needs to be developed further; essentially it is desirable that XML3D will become a part of HTML5, thus a standard which is accessible in every browser on any device. The utilization of the generated routing graph for complex analysis, such as emergency evacuation simulations, is also a desirable, but challenging task, and will be investigated in the near future. Regarding the data acquisition by OSM contributors, there is fertile ground for future research. Due to missing GPS signals inside buildings, other methodologies, techniques and devices must be utilized for indoor mapping activities. Some existing examples are step counters, images of publically accessible evacuation plans, or distinct indoor mapping apps, but easy to use and cost effective solutions are required. Research on indoor LBS applications, not necessarily 3D but also 2D, based on indoor VGI will increase the application presented here further.

Acknowledgments

The authors of this paper would like to express their gratitude to the anonymous reviewers. By providing their valuable comments on this paper they contributed towards the improvement of our work. Additionally we would like to thank Rainer Jochem for his support and help towards understanding and implementing the XML3D specification. This research has been partially funded by the Klaus-Tschira Foundation (KTS) Heidelberg.

References

1. Goetz, M.; Zipf, A. Open Issues in Bringing 3D to Location Based Services (LBS): A Review Focusing on 3D Data Streaming and 3D Indoor Navigation. In *Proceedings of the 5th 3D GeoInfo Conference*, Berlin, Germany, 3–4 November 2010; pp. 121–124.
2. Go indoors with Google Maps 6.0 for Android. Available online: <http://googlemobile.blogspot.de/2011/11/go-indoors-with-google-maps-60-for.html> (accessed on 31 March 2012).
3. Bing Maps venue maps now feature nine largest US malls, 148 total. Available online: http://www.bing.com/community/site_blogs/b/maps/archive/2011/03/22/bing-maps-venue-maps-now-feature-largest-nine-us-malls-148-total.aspx (accessed on 31 March 2012).
4. NAVTEQ Extends the Journey Beyond the ‘Front Door’. Available online: <http://press.navteq.com/index.php?s=4260&item=30551> (accessed on 31 March 2012).
5. Meijers, M.; Zlatanova, S.; Pfeifer, N. 3D Geo-Information Indoors: Structuring for Evacuation. In *Proceedings of the 1st international ISPRS/EuroSDR/DGPF-Workshop on Next Generation 3D City Models (EuroSDRBonn)*, Bonn, Germany, 21–22 June 2005; p. 6.
6. Coors, V.; Zipf, A. MoNa 3D—Mobile Navigation using 3D City Models. In *Proceedings of the 4th International Symposium on LBS and Telecartography 2007*, Hongkong, China, 8–10 November 2007; p. 10.
7. Zlatanova, S. SII for Emergency Response: The 3D Challenges. In *Proceedings of the XXI ISPRS Congress, Part B4-TYC IV*, Beijing, China, 3–11 July 2008; pp. 1631–1637.
8. Havlik, D.; Schade, S.; Sabeur, Z.A.; Mazzetti, P.; Watson, K.; Berre, A.J.; Mon, J.L. From sensor to observation web with environmental enablers in the future internet. *Sensors* **2011**, *11*, 3874–3907.
9. Goodchild, M.F. Citizens as voluntary sensors: Spatial data infrastructure in the world of web 2.0. *Int. J. Spat. Data Infrastruct. Res.* **2007**, *2*, 24–32.
10. Zielstra, D.; Zipf, A. A Comparative Study of Proprietary Geodata and Volunteered Geographic Information for Germany. In *Proceedings of the 13th AGILE International Conference on Geographic Information Science*, Guimarães, Portugal, 10–14 May 2010; pp. 1–15.
11. Haklay, M. How good is volunteered geographical information? A comparative study of OpenStreetMap and ordnance survey datasets. *Environ. Plan. B Plan. Des.* **2010**, *37*, 682–703.
12. Neis, P.; Zielstra, D.; Zipf, A. The street network evolution of crowdsourced maps: OpenStreetMap in Germany 2007–2011. *Future Internet* **2012**, *4*, 1–21.
13. IndoorOSM. Available online: <http://wiki.openstreetmap.org/wiki/IndoorOSM> (accessed on 31 March 2012).

14. Goetz, M.; Zipf, A. Extending OpenStreetMap to Indoor Environments: Bringing Volunteered Geographic Information to the Next Level. In *Proceedings of the Urban and Regional Data Management: Udms Annual 2011*, Delft, The Netherlands, 28–30 September 2011; pp. 47–58.
15. Sons, K.; Klein, F.; Ruinstein, D.; Byelozorov, S.; Slusallek, P. XML3D: Interactive 3D Graphics for the Web. In *Proceedings of the 15th International Conference on Web 3D Technology*, Los Angeles, CA, USA, 2010; pp. 175–184.
16. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 267–271.
17. Abowd, G.D.; Atkeson, C.G.; Hong, J.; Long, S.; Kooper, R.; Pinkerton, M. Cyberguide: A mobile context-aware tour guide. *Wirel. Netw.* **1997**, *3*, 421–433.
18. Höllerer, T.; Feiner, S.; Terauchi, T.; Rashid, G.; Hallaway, D. Exploring MARS: Developing indoor and outdoor user interfaces to a mobile augmented reality system. *Comput. Graph.* **1999**, *23*, 779–785.
19. Gilliéon, P.-Y.; Bertrand, M. Personal Navigation System for Indoor Applications. In *Proceedings of the 11th IAIN World Congress*, Berlin, Germany, 21–24 October 2003; p. 15.
20. Schmidt-Belz, B.; Hermann, F. User Validation of a Nomadic Exhibition Guide. In *Proceedings of the Human Computer Interaction with Mobile Devices and Services (Mobile HCI 2004)*, Glasgow, UK, 13–16 September 2004; pp. 86–97.
21. Pateli, A.G.; Giaglis, G.M.; Spinellis, D.D. Trial Evaluation of Wireless Info-communication and Indoor Location-based Services in Exhibition Shows. In *Proceedings of the Advances in Informatics*, Volas, Greece, 11–13 November 2005; pp. 199–210.
22. Kargl, F.; Geßler, S.; Flerlage, F. The iNAV Indoor Navigation System. In *Proceedings of the 4th International Symposium on Ubiquitous Computing Systems (UCS 2007)*, Tokyo, Japan, 25–28 November 2007; pp. 110–117.
23. Hijazi, I.; Ehlers, M. Web 3D Routing in and between Buildings. In *Proceedings of the Fourth National GIS Symposium*, Al-Khobar, Saudi Arabia, 4–6 May 2009; pp. 11–21.
24. Rehrl, K.; Göll, N.; Leitinger, S.; Bruntsch, S. Combined Indoor/Outdoor Smartphone Navigation for Public Transport Travellers. In *Proceedings of the 3rd Symposium on LBS & TeleCartography 2005*, Vienna, Austria, 28–30 November 2005; pp. 235–239.
25. Thunderhead Explorer: Indoor Routing for UC2009 on iPhone. Available online: <http://thunderheadxpler.blogspot.com/2009/07/indoor-routing-for-uc2009-on-iphone.html> (accessed on 31 March 2012).
26. Inoue, Y.; Ikeda, T.; Yamamoto, K.; Yamashita, T.; Sashima, A.; Kurumatani, K. Usability Study of Indoor Mobile Navigation System in Commercial Facilities. In *Proceedings of the 2nd International Workshop on Ubiquitous Systems Evaluation (USE '08)*, Seoul, Korea, 21 September 2008; pp. 1–6.
27. Papataxiarhis, V.; Riga, V.; Nomikos, V.; Sekkas, O.; Kolomvatsos, K.; Tsetsos, V.; Papageorgas, P.; Vourakis, S.; Xouris, V.; Hadjiefthymiades, S.; Kouroupetroglou, G. MNISIKLIS: Indoor Location Based Services for All. In *Location Based Services and TeleCartography: From Sensor Fusion to Ubiquitous LBS*; Springer: Berlin, Germany, 2008; Volume 2, pp. 263–282.
28. Karimi, H.A.; Ghafourian, M. Indoor routing for individuals with special needs and preferences. *Trans. GIS* **2010**, *14*, 299–329.

29. Lyardet, F.; Grimmer, J.; Muhlhauser, M. CoINS: Context Sensitive Indoor Navigation System. In *Proceedings of the 8th IEEE International Symposium on Multimedia (ISM'06)*, San Diego, CA, USA, 11–13 December 2006; pp. 209–218.
30. Lyardet, F.; Szeto, D.W.; Aitenbichler, E. Context-Aware Indoor Navigation. In *Proceedings of the European Conference on Ambient Intelligence (AML 08)*, Nürnberg, Germany, 19–22 November 2008; pp. 290–307.
31. XML3D—Based on W3C technologies. Available online: <http://www.xml3d.org/features/> (accessed on 23 May 2012).
32. XML3D 0.4 Specification. Available online: <http://www.xml3d.org/xml3d/specification/current/> (accessed on 23 May 2012).
33. Stats—OpenStreetMap Wiki. Available online: <http://wiki.openstreetmap.org/wiki/Statistics> (accessed on 31 March 2012).
34. Watchlist. Available online: <http://tagwatch.stoecker.eu/Planet-latest/En/watchlist.html> (accessed on 31 March 2012).
35. Tagwatch Planet-latest. Available online: <http://tagwatch.stoecker.eu/Planet-latest/En/tags.html> (accessed on 31 March 2012).
36. OpenStreetMap Wiki. Available online: <http://wiki.openstreetmap.org/> (accessed on 3 May 2012).
37. indoorOSM—Mapping the Indoor World. Available online: <http://indoorosm.uni-hd.de/> (accessed on 10 May 2012).
38. Download OpenStreetMap Extracts. Available online: <http://download.geofabrik.de/osm/> (accessed on 2 May 2012).
39. Osmosis—OpenStreetMap Wiki. Available online: <http://wiki.openstreetmap.org/wiki/Osmosis> (accessed on 2 May 2012).
40. Goetz, M.; Zipf, A. Formal definition of a user-adaptive and length-optimal routing graph for complex indoor environments. *Geo-Spat. Inf. Sci.* **2011**, *14*, 119–128.

© 2012 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).