

# Web 3D Routing in and between Buildings

Prof Dr Eng. Manfred Ehlers, Eng Ihab Hijazi MS.c in GIS, PhD candidate

Institute for Geoinformatics and Remote Sensing, University of Osnabruck,

Osnabrück, Germany

{mehlers}{ihijazi}@[igf.uni-osnabrueck.de](mailto:igf.uni-osnabrueck.de)

Current routing systems in GIS software mostly provide routes that allow the users to navigate between source and destination points in 2 dimensions. This paper describes the development of a web-based 3D routing system for a university campus using Open Source Software (OSS) and Open Specifications (OS). The system uses the advantages of interoperability and allows the integration and extension of different system components. A data model is described and the process of creating the data model and the migration of the data stored in dxf architectural drawings to the data model are explained. The paper also discusses the architecture and the interaction of the different prototype components such as 3D viewer, database, and programming languages. Furthermore, it describes customized tools that were developed to provide the users a simple interface to interact with the system through a standard internet browser.

**Keywords:** 3D routing, 3D infrastructure, Open Source

## 1. Introduction

Institutions with large campuses such as universities, hospitals, or companies are in great need for routing systems that support pedestrians to find their routes within buildings. Most of the current GIS online mapping tools offer powerful route searching functions that target automobile travel and model only local one-dimensional street network (Musliman et al., 2006).

In the past few years the attention in modeling interiors has increased (Zlantanova, 2008). Considering the interior space in GIS requires extending GIS to the third dimension. Most of the existing GIS are based on 2D environments which reveal serious limitations when considering multi-level structure (Meijers et al., 2005). Current applications are unable to represent continuous indoor and outdoor spaces which are fundamental to a complex of buildings. Building interiors present the additional challenges of complex connectivity, changes in elevation, and movement through large open spaces (Whiting et al., 2007). Meijers et al. (2005) described a method for structure and modeling of interiors, to be used for an intelligent computation of evacuation routes. Whiting et al. (2007) described a method for storing a metric topological data model representing an extended indoor/outdoor multi-building environment.

In this paper we present a web based 3D GIS solution for routing in an urban environment, by using a 3D network analysis based on a modified Dijkstra algorithm. A method for semi-automatic extraction of different network elements from architectural CAD drawings is presented. The method is demonstrated using OSS and OS on data from University of Osnabruck campus, comprising of two buildings with more than 100 indoor and additional open spaces.

The aim of the work presented can be summarized into two points:

- To contribute to the knowledge base about OS and OSS, and how they can be used and extended to create a 3D web-based solution. The system implements geoprocessing functionalities that are not currently available as off-the-shelf solutions in commercial web-GIS software.
- To present a 3D routing system and its advantages for 3D GIS developments in the context of urban environments and management of building.

The paper is organized in 5 sections: After the introduction (section 1) section 2 defines OS, OSS, and interoperability and provides a brief background of the specific OS and OSS technologies used to create the 3D web-based routing spatial information and system architecture; section 3 presents the data model and the topology creation methodology, section 4 a demonstration for the developed interfaces and the network analysis; in section 5 we present conclusions and suggestions for future research and implementation.

## **2. OSS and OS Background**

OSS is defined as computer software whose license “permits users to use, change, and improve the software, and to redistribute it in modified or unmodified form. It is very often developed in a public, collaborative manner” (Wiki, 2009). They present a healthy competition with proprietary software and may lead to affordable pricing and increased access (Moreno-Sanchez et al., 2007).

Among the well known OSS projects are the Linux operating system and the Apache web server. A comprehensive list of GIS related OSS can be found at (<http://opensourcegis.org/>). As of January 2009, the open source GIS website has listed more than 247 of GIS related OSS projects (OpenGIS, 2009).

OS promotes interoperability through its public availability (OGC, 2009), it provide software developers with information about a given specification as well as specific programming rules and advice for implementing the interfaces and/or protocols. It allows the systems to work easily with each other and overcome tedious batch conservation tasks, import/export obstacles (Anderson & Moreno-Sanchez, 2003). Among OGC specifications are KML and CityGML, two standards for 3D GIS.

## 2.1 Design, Architecture, and Development

This section provides background information about the origin, relationships among the OS and OSS, and their relevance for the creation of the 3D geoprocessing web GIS.

Different aspects have been considered to select the appropriate OSS. We concentrated on mature OSS projects with clear identification core development teams and wide development and broad active users communities (particularly at the international level). We also sought projects with well-established records of success in developing simple prototype systems as well as large mission-critical applications. Other important aspect were the available resources such as books, user forums, user conferences, and online documentation (Moreno-Sanchez et al., 2007). After considering several options we decided to use the OSS described below.

The current version of the routing system is the result of a first development cycle. Figure 2 presents the system integration. The system is maintained and processed/operated on an Apache web server.

*Google earth plug-in* is a JavaScript API that allows embedding Google Earth into web pages and visualizes 3D models in standard web browsers. It provides powerful 3D web visualization functionalities; the viewer has movements and navigation aid tools including 'zoom in' and 'zoom out', he can easily control and change camera locations. It is able to visualize the KML standard. For the proposed system the viewer was customized using Java script and PHP to be integrated in the routing system website. The campus area included in this prototype and the results of the analysis were visualized using this viewer (Google, 2009a).

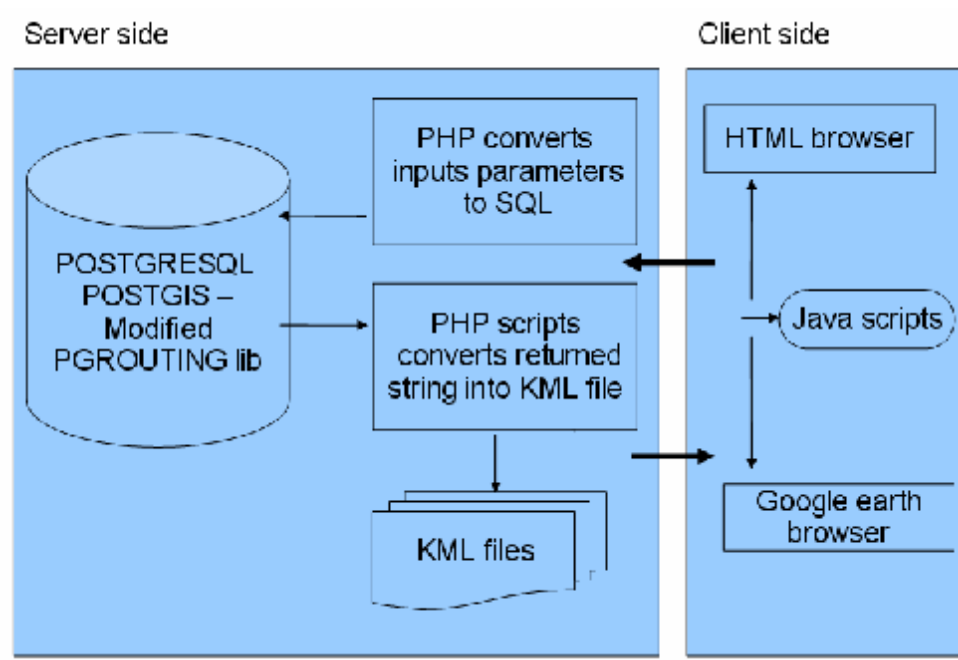


Figure 1: Software components that integrate the 3D web routing GIS system

*KML* Keyhole Markup Language: KML is an XML data format that has a tag-based structure with names and attributes to display geographic data in an Earth browser, such as Google Earth. KML allows the definition of styles to specify feature appearance, icons, and labels to identify locations. It can also display textured objects. The KML file has been generated on the fly to visualize the result of the suitable route queries and the 3D model of the campus building (Google, 2009b). Through java script code it is possible to adjust several visualization properties of these models such as 'hide texture' and 'change visibility'.

*PostgreSQL/PostGIS* is a powerful, OSS object-relational database system, supporting almost all SQL functions including sub selects, transactions, and user defined data types and functions (PostgreSQL, 2008). The PostGIS extension offers the capability to store spatial features with x,y,z coordinates and to perform more than 80 spatial and geoprocessing operations including limited 3D spatial analysis such as 3D BOX (PostGIS, 2008). It is used to process spatial and attributes queries and to store the network as spatial tables with its attributes.

*Pgrouting* is a set of C++ libraries that provide routing functionality to PostgreSQL/PostGIS. Its tools provide shortest route functionalities for real road networks including considering restrictions with heuristics. The tools are based on the well known Dijkstra shortest route algorithm (Pgrouting, 2009). Some modification were performed on the *Pgrouting* functions to be able to create a 3D topology (graph), and to perform the network analysis of the indoor and outdoor spaces. This considers the different level of space (3D) and the characteristics of the indoor environment such as door types, elevators and stairs.

*PHP* was used to create inter-application communication and to process the user's input and send it back to the web interface. An SQL query to the attributes and the network that are stored in PostGIS spatial tables is built using customized user interfaces such as buttons, checkboxes, or text fields. The query is then submitted to the server using an HTML form that captures the user's input. At the server, a PHP script converts the user's input into SQL statement that is submitted to PostgreSQL/PostGIS for processing. The geometry texts that are returned by PostgreSQL/PostGIS are converted to KML code format using PHP and then visualized in the Google Earth plug-in.

### **3. Conceptual Data Model**

To be able to define meaningful routes within and between buildings, different building elements such as rooms, door, stairs, walkways, etc have to be modeled. For this purpose the database structure has been divided into two main groups:

1. 3D reference model: a 3D representation of the building's interior and its surrounding environment (Figure 2a)
2. 3D network: topological relationships between the different elements of a building such as rooms or corridors (Figure 2b)

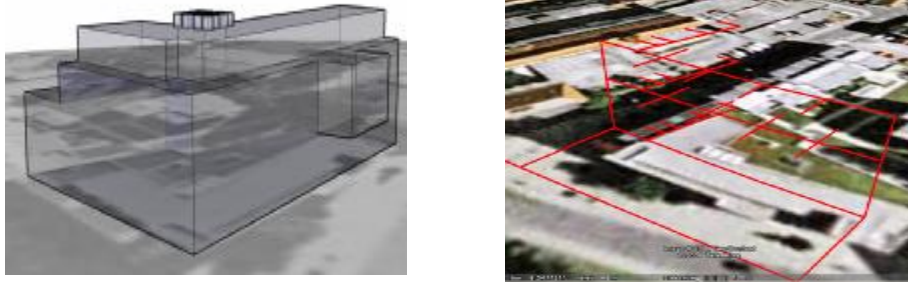


Figure 2: (a) building model. (b) Network model (Topology + Geometry)

The first part of the model is stored for 3D visualization and drawing purposes. It does not contain any topological information. The main objective is to provide a realistic representation that provides an orientation on the screen. The 3D network represents the path network in the building based on graph representation and consists of paths between two spaces; each node of the path represents one space. Modeling the different access types such as doors, elevators, stairs, emergency exits are also considered (Karas et al., 2006).

### 3.1 Model Implementation

The proposed model is generated from architectural drawings and the site plan of the university campus. The first part of the model that is described in the above section was used to generate the network part. The architectural drawings are stored in dxf (a vector file) format. There are different annotations (text) that are used in the architectural drawings as labels to define the room number and usage (Figure 3). These annotations texts are manipulated and re-used in the database as attributes for the different network elements. The model implementations are described in the following steps:



Figure 3: Architectural drawing with annotations

### *Step1: Generating the 3D model*

The 3D models of the building are being generated using Sketch-up a (freeware software) and stored in KML format. The texture of the building could be obtained by using photos that are captured using regular small-format cameras. After generating the 3D model, it is converted to KML format. This format can be visualized via internet browsers and Google Earth plug-in.

### *Step2: Generating the network*

The basic elements of the network model are spaces and portals. Each space represents continues a physical region such as room, corridor, sidewalk, or patch of the outdoor area. Each portal represents an adjacency such as a door, elevator, or a stairwell connecting two spaces. In this section we will describe a semi-automatic network generation process.

*Generating the spaces:* The spaces are represented using points, each space is represented by a closed polygon in the dxf files. The centroid function was used to determine the center of the spaces (Figure 4).

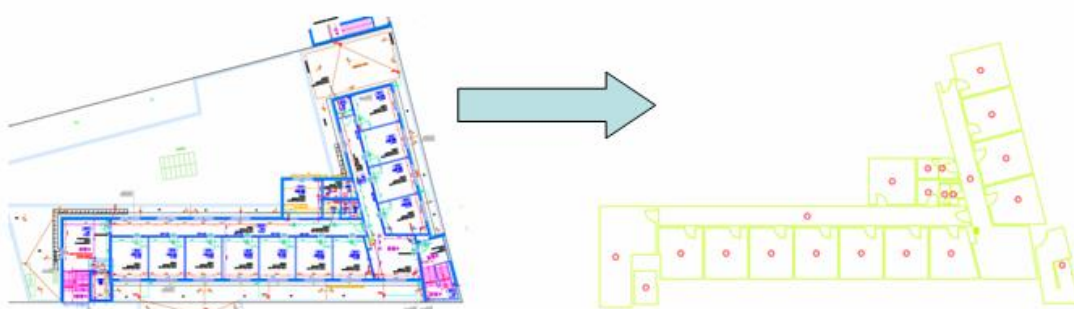


Figure 4: Automatic creation of spaces nodes

*Generating the portals:* Each portal is represented as a line that has a source space and a destination space. It can be classified as different types, horizontal portals that connect different spaces by the same floor or spaces in two different buildings, vertical portals that connect two spaces by different floors such as elevators, stairs, etc. Some of these portals can be generated automatically such as corridors (see Figure 6).

Portals may be explicit or implicit. Explicit portals represent a connection between spaces that are physically separated by a barrier (e.g. a doorway through a wall). Implicit portals represent connections between spaces with no physical barrier. For example, the photo in Figure 5 shows a large open area inside the university cafeteria with no wall between the corridor and the café tables. The two regions are defined separately in the floor plans to differentiate between circulation zones and food services. We identify implicit portals by an incident edge between two spaces similar to explicit portals (Whiting et al., 2007).



Figure 5: Implicit portals: no physical barriers exist although the corridor and the sitting area are modeled as separate places

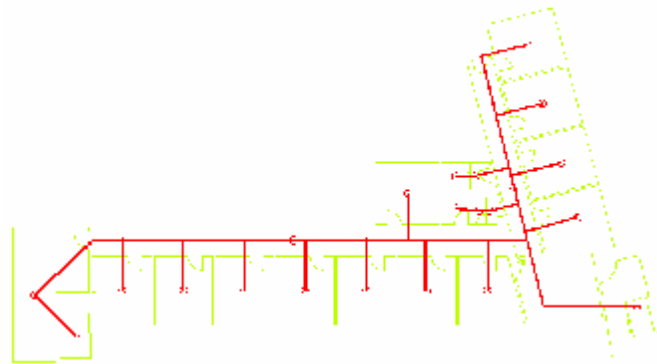


Figure 6: Creation of the portals

*Integrating spaces with portals:* The portals, as indicated above, connect two nodes. After locating the nodes that indicate rooms and corridors, users interactively point out which room nodes connect with which corridor nodes. Each corridor node is then connected to stairs or elevator points; two different floors are connected via stairs or elevators (Figure 7).

### *Step 3: Spatial database*

After generating the network (portals, spaces lines and points) the network model is transferred to the Postgres/PostGIS database. The network consists of two tables, one for the spaces as points, and the second for the portals as links. The spaces table (points) has two fields for the room use and number. The link (line) table consists of link IDs, source and target which are the start and end point of every link, the 3D length of the link which is used as cost parameter. Also stored is additional information such as closed portals, non-functional elevators, or restricted turns.

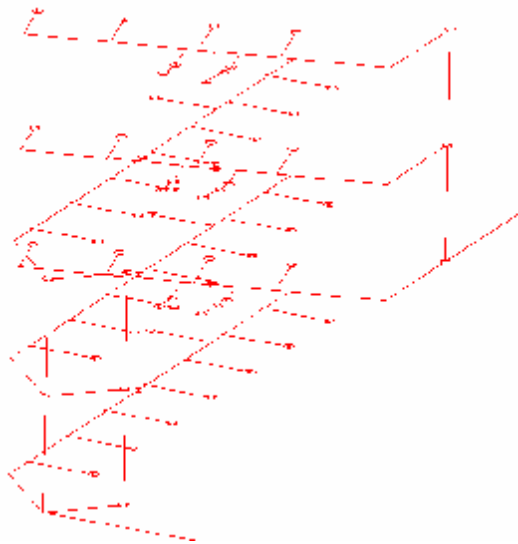


Figure 7: Integration of network elements

#### Step 4: Creating the topology

The topology (graph) is created using the Pgrouting functionalities. The functions are adjusted to create the graph in 3D, the network table is also stored in the database after creating the graph (Figure 8). Two fields are filled with source and target information which shows the link sequence between any two points in the network.

id	gid (PK serial)	id integer	line geometry	s1 double precision	s2 double precision	s3 double precision	s4 double precision	source integer	target integer	length double precision	name character(255)
1		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	1	1.41421356237	1-1020452134
2		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	55	1.41421356237	2-1020452134
3		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	22	1.41421356237	3-1020452134
4		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	8	1.41421356237	4-1020452134
5		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	-	1.41421356237	5-1020452134
6		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	5	1.41421356237	6-1020452134
7		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	6	1.41421356237	7-1020452134
8		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	7	1.41421356237	8-1020452134
9		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	8	1.41421356237	9-1020452134
10		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	9	1.41421356237	10-1020452134
11		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	10	1.41421356237	11-1020452134
12		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	11	1.41421356237	12-1020452134
13		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	12	1.41421356237	13-1020452134
14		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	13	1.41421356237	14-1020452134
15		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	14	1.41421356237	15-1020452134
16		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	15	1.41421356237	16-1020452134
17		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	16	1.41421356237	17-1020452134
18		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	17	1.41421356237	18-1020452134
19		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	18	1.41421356237	19-1020452134
20		0	LINESTRING(10 20 45 21 34 4)	10	45	21	34	1	-	1.41421356237	20-1020452134

Figure 8: Created graph with source and target fields

## 4. Network Analysis

To allow a system user to perform the routing analysis, a web site is customized which can be accessed from any client's PCs or mobile device. The user needs to enter the source and destination and click on the 'find route' option. An optimum path between the user defined entities is calculated and visualized in 3D (Figure 9).

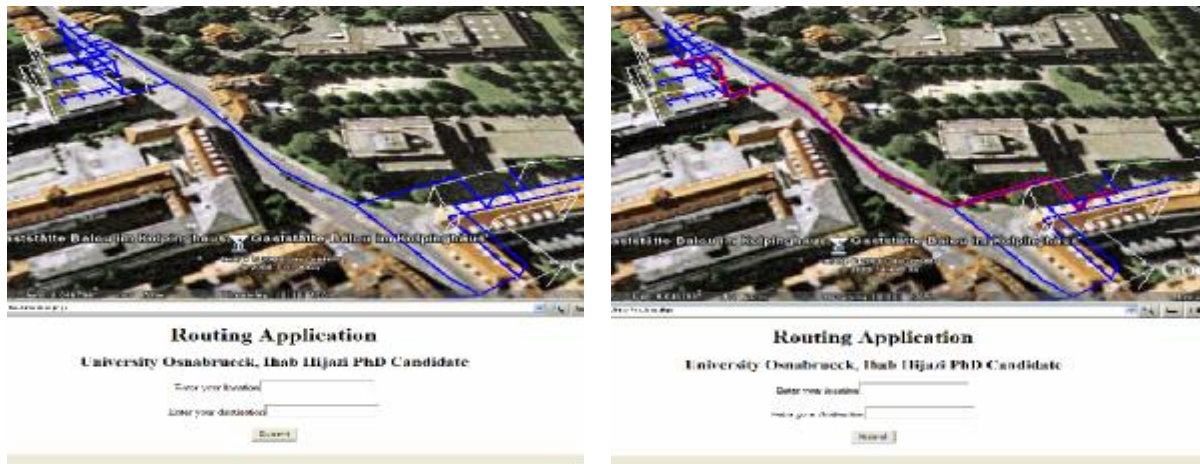


Figure 9: 3D visualization of the building and the network models (a) and 3D visualization of the optimum path between two entities (b).

## 5. Conclusion and Future Work

In this study, a new method for routing within building is presented based on a 3D web geoprocessing OSS solution. The prototype demonstrates the potential of using OS to provide highly interactive 3D GIS applications on the web. Postgres/PostGIS provides a robust database management system that offers a considerable and continuously increasing number of geoprocessing functions. Google Earth and KML standard proved to be an easy way to use 3D visualization tools with powerful functionalities. Pgrouting libraries provide easy to use function via PostgreSql/PostGIS. It is possible to modify these libraries to be used in 3D network (non-planar graph) analysis. This could be demonstrated by the feasibility of computing routes through graph theory and modified 3D Disjkstra algorithms.

From this experience we can conclude that for organizations with limited resources such as academic institutions sophisticated GIS solutions can be implemented through the WWW with software that is free of costs. In addition to building managers, architectural engineers could benefit from the current prototype. They can find answers to different practical questions that related to their field: What are the common navigation paths within their designed environment? Do they provide adequate circulation capacity? Does their design provide suitable facilities for access and navigation for people with disabilities? (Whiting et al., 2007)

Future work will include the incorporation of CityGML and IFC standards (OGC members). Also, more criteria in finding the routes for evacuation scenarios and the different state of the spaces and portals have to be considered.

## References

Anderson, G. and Moreno-Sanchez, R., 2003, Building web-based spatial information solutions around Open Specifications and Open Source Software. *Transactions in GIS*, 7, pp. 477-466.

Google, 2009a. About Google earth Plug-in. Retrieved January 5, 2009 from the World Wide Web: <http://code.google.com/apis/earth/>

Google, 2009b. About Google earth Plug-in. Retrieved January 5, 2009 from the World Wide Web: <http://code.google.com/apis/earth/>

Karas, I., Batuk F., Akay, A., and Baz, B., 2006, Automatically Extracting 3D Models and Network Analysis for Indoors. In: Abdul-Rahman, A., Zlatanova, S. and Coors V., (eds.) *Innovation in 3D-Geo Information System*, pp. 395-404, Springer Berlin Heidelberg.

Meijers, M., Zlatanova, S. and Pfeifer, N., 2005, 3D geo-information indoors: structuring for evacuation. In G Groeger & TH Kolbe (Eds.), *Proceedings of the First International Workshop on Next Generation 3D City Models*, pp. 11-16, Bonn: ISPRS/EuroSDR/DGPF/Universitaet Bonn.

Moreno-Sanchez, R., Anderson, G., Cruz, J. and Hayden M., 2007, The potential for the use of Open Source Software and Open Specifications in creating Web-based cross-border health spatial information systems. In: *International Journal of Geographic Information Science*, 21, pp. 1136-1163.

Musliman, I., Abdul Rahman, A. and Cross, V., 2006, 3D Navigation for 3D-GIS-Initial Requirements. In: Abdul-Rahman, A., Zlatanova, S. and Coors V., (eds.) *Innovation in 3D-Geo Information System*, pp. 259-268, Springer Berlin Heidelberg.

OGC, 2009, glossary, Retrieved October 25, 2008 from the World Wide Web: <http://www.opengeospatial.org/ogc/glossary/o>.

OpenGIS, 2009, about open GIS, Retrieved January 2, 2009 from the World Wide Web: <http://opensourcegis.org/>.

Pgrouting, 2009, About Pgrouting, Retrieved January 4, 2009 from the World Wide Web: <http://pgrouting.postlbs.org/>.

PostGIS, 2008, About PostGIS, Retrieved October 8, 2008 from the World Wide Web: <http://www.postgis.org/>.

PostgreSQL, 2008, About PostgreSQL, Retrieved October 10, 2008 from the World Wide Web: <http://www.postgreSQL.org/>

Whiting E., Battat J. and Teller S., 2007, *Topology of Urban Environments*, Retrieved January 5, 2009 from the World Wide Web: <http://www.mit.edu/~ewhiting/resources/pubs/WhitingCAADFutures07.pdf>.

Wiki, 2009, Open Source Software, Retrieved October 10, 2009 from the World Wide Web: [http://en.wikipedia.org/wiki/Open\\_Source\\_software](http://en.wikipedia.org/wiki/Open_Source_software).

Zlantanova S., 2008, SII for Emergency Response: The 3D Challenges, The international Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXVII, Part B4, Beijing 2008, pp. 1631-1637