

# A methodology for evacuation route planning inside buildings using geospatial technology

A. Ramón<sup>1</sup>, A. B. Rodríguez-Hidalgo<sup>2</sup>, J. T. Navarro-Carrión<sup>1</sup>  
& B. Zaragoza<sup>1</sup>

<sup>1</sup>*Instituto Interuniversitario de Geografía,  
Universidad de Alicante, Spain*

<sup>2</sup>*Departamento de Edificación y Urbanismo,  
Universidad de Alicante, Spain*

## Abstract

Evacuation route planning is a fundamental task for building engineering projects. Safety regulations are established so that all occupants are driven on time out of a building to a secure place when faced with an emergency situation. As an example, Spanish building code requires the planning of evacuation routes on large and, usually, public buildings. Engineers often plan these routes on single building projects, repeatedly assigning clusters of rooms to each emergency exit in a trial-and-error process. But problems may arise for a building complex where distribution and use changes make visual analysis cumbersome and sometimes unfeasible. This problem could be solved by using well-known spatial analysis techniques, implemented as a specialized software able to partially emulate engineer reasoning. In this paper we propose and test an easily reproducible methodology that makes use of free and open source software components for solving a case study. We ran a complete test on a building floor at the University of Alicante (Spain). This institution offers a web service (WFS) that allows retrieval of 2D geometries from any building within its campus. We demonstrate how geospatial technologies and computational geometry algorithms can be used for automating the creation and optimization of evacuation routes. In our case study, the engineers' task is to verify that the load capacity of each emergency exit does not exceed the standards specified by Spain's current regulations. Using Dijkstra's algorithm, we obtain the shortest paths from every room to the most appropriate emergency exit. Once these paths are calculated, engineers can run simulations and validate, based on path statistics, different cluster configurations. Techniques and tools applied in this research



would be helpful in the design and risk management phases of any complex building project.

*Keywords: emergency evacuation, route planning, building engineering, geospatial technology*

## **1 The importance of computational models for evacuation routing**

Around the world, architects and engineers are always concerned with how to demonstrate that their building designs are safe and evacuation routes will be efficient in an emergency situation [1]. Because of these requirements, many efforts have been made in developing two main groups of techniques: (1) evacuation demonstrations, and (2) the adherence to building codes.

Evacuation demonstration exercises are considered impractical and expensive. Results from these tests provide limited confidence in an emergency situation. This is why up-to-date approaches use simulation techniques for achieving an efficient evacuation. [2] divide the most common simulation methods applied to evacuation studies into seven different types (cellular automata, lattice gas, social force, fluid dynamics, agent-based, game theory and experiments with animals) with different strengths and weaknesses. While simulations do have their advantages, they also need to assess different technical problems related to spatial modelling [3, 4] or the amount of predefined conditions related to human physiology and psychology [2, 5, 6]. Evacuation simulations become more complex when dealing with large buildings, but the problem gets worse when the floor plan is intricate, some of the equipment is risky in nature and the number of users is high or registers at critical peaks at certain times. There are also cases in which evacuation may be hindered because, at some point in a building's life, architectural design and actual function got misaligned.

As mentioned above, simulation methods are widely explored in the scientific literature. These methods explore the environmental, behavioural and procedural aspects of the evacuation process. However, the configurational aspects of evacuation (building layout, exit configurations, travel distance, etc.) are those that building codes take into account [1, 7]. This means that configurational aspects must be considered in the design phase. Afterwards, experts may use simulation methods for testing other aspects of the evacuation plans.

According to the previous approach, in this work we assume the concept of evacuation route as defined by the DB-SI, which is Spain's base code for building safety in case of fire and the primary reference for engineers. This document states that an evacuation route is a path from an evacuation origin to either a floor exit or a building exit [8]. It also defines requirements, variables and procedures that will eventually determine the validity of any individual path or the suitability of an emergency exit for serving a given set of evacuation origins. The resulting standards cover construction elements as well as parameters directly involved in emergency planning. Regarding construction elements, there are rules

for calculating the required number of exits; for dimensioning doors, corridors, stairs and ramps; and, finally, for computing maximum evacuation capacity of each exit. These are rules that belong to the architectural project's design phase. The second category of standards drives the process of evacuation route planning and encompasses occupancy rates and distance constraints. In the near future, the DB-SI is meant to be completely integrated in a more general framework of building standards, namely The Eurocodes. However, there are some difficulties that may slow down adherence to the EU standards. The proper national implementation of the Eurocodes and their correct use by designers should facilitate the application of standards through training courses, handbooks and software, among other possibilities [9].

Related to the previously described need for methodology and tools [9], there are several software packages for performing dynamic fire simulations, which adjust the Eurocodes needs (e.g. <http://eurocodes.jrc.ec.europa.eu/doc/EurocodeSoftW.htm>). Our work focuses on a complementary methodology – we propose a tool for increasing efficiency in common engineering tasks during evacuation route planning. This way, our approach is somewhere between fire simulations and evacuation simulations.

In this paper we develop a testing environment which provides some useful methods for evacuation route planning: (1) automatic network generation (nodes and connected edges); (2) manual cluster configuration interface to assign a group of origin nodes to each exit node; (3) basic simulation capabilities to determine if a particular cluster configuration fits maximum evacuee figures imposed on each emergency exit by design standards.

## 2 A case study for the University of Alicante

We have chosen to work with an existing building at the University of Alicante (Spain). The Faculty of Sciences VI (CTQ) is a large building with teaching and research facilities. However, in its design, this building was meant to be a chemistry laboratories module, so the original evacuation plans had to be updated due to the higher risk caused by a high building occupancy increase. We also would like to emphasize that each step in our methodology is solved using publicly accessible data from the University of Alicante and Free and Open Source Software (FOSS), thus making our experiment reproducible and verifiable. For all of these reasons, this case study is an interesting test bed to create needed data structures and to examine the proposed operations sequence.

### 2.1 Input data and preprocessing

As for data acquisition, we propose the use of a geographic database through an open HTTP based protocol. In this kind of database, floor plans get represented as 2D vector polygon maps of rooms, passageways and partitions. The resulting data model is a cartographic synthesis of architectural CAD projects that preserves



area and distribution information while providing ‘smart’ data, that is to say, new geometry objects with built-in behaviour.

We have used SIGUA (Spanish acronym which stands for University of Alicante Geographical Information System) as our primary data source. SIGUA is a corporate system for building asset management. It can be viewed as a battery of services built upon a PostgreSQL/PostGIS spatial database [10]. Among these there is a WFS (Web Feature Service) which provides public access to the geographic data store. WFS is a fundamental interoperability specification published by the OGC (Open Geospatial Consortium) and operates as an HTTP-based protocol that allows complex vector data requests from a client [11]. Through this service, users can query and download enclosures from the data store. Each enclosure is a 2D polygon geometry representing a room, a passageway or a partition. Rooms are further classified into one of up to thirty use categories such as research and computer labs, lecture rooms or offices. Enclosures are the spatial data units of the geographic database. They are logically organized by floor levels and are hierarchically coded according to the building floor they belong to. Upon this basis, we can define a WFS GetFeature request to render a particular building floor plan using a plain URL. Two request schemas, which use different and mutually exclusive filtering mechanisms, can be used: a bounding box search and a spatial query – the latter being based on OGC’s normalized XML syntax called Filter Encoding. Two examples, one for each kind of request, are available at <https://gist.github.com/4573145> and <https://gist.github.com/4573212>. We have tested both to return the set of enclosures that comprises the first floor of the Faculty of Sciences VI (CTQ), a large building with teaching and research facilities, chosen for our case study. WFS response is given as GML (Geography Markup Language), which is OGC’s standardized XML-based language for geographic vector data exchange [12]. Today, all major desktop GIS applications allow users to perform these kind of requests transparently. We have chosen Quantum GIS, whose WFS client is seamlessly integrated within its vector data provider range. Once the service base URL is provided (<http://www.sigua.ua.es/cgi-bin/siguawms>) and the connection opens, requests are easily defined through a GUI which, presently, just supports bounding box filter definition. The response is further processed by the Quantum GIS internal GML parser and rendered as a polygon map showing the intended floor plan.

This map is topologically correct so that, given an enclosure, its adjoining ones can be obtained. This feature is central for calculating the transition polygon, a term used in this research context to name the union of all adjoining enclosures crossed by the evacuation route network within a building floor. Safety regulations should determine the transition polygon shape since they are expected to clearly define what an evacuation origin is. Having the DB-SI specification as our regulatory framework, the transition polygon is the aggregation of evacuee passageways and rooms likely to be at full-capacity according to their function. Enclosures in SIGUA hold a use category identifier that allows for functional classification of polygons on the floor plan following DB-SI criteria. This means running simple attribute queries which, in our test, involve retrieving hallways,



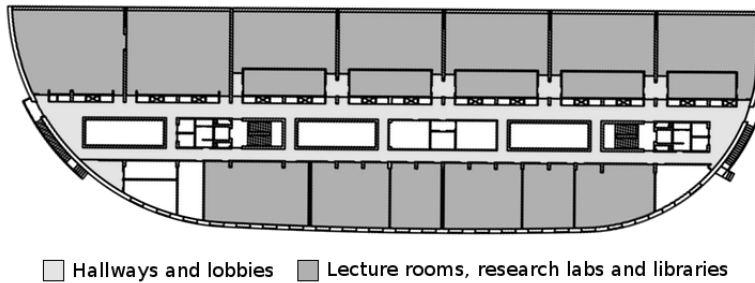


Figure 1: First Floor floor plan of CTQ building. Polygons are classified according to DB-SI specifications on passageways and rooms accommodating a large number of people.

lobbies, lecture rooms, research labs and libraries (Attribute query definitions can be browsed at <https://gist.github.com/4584740>). According to DB-SI, the last three use categories correspond to sources of potentially high evacuee flow, so path lengths inside of these kind of premises must be computed. Figure 1 shows the outcome of this classification as a thematic map rendered in Quantum GIS. The transition polygon is ultimately obtained by running a dissolve operation on the target enclosures, which basically means discarding shared edges in the input polygon set and getting a new aggregated geometry. The resulting transition polygon has been computed by the Quantum GIS geometry combination tool.

## 2.2 Node generation

In the context of a building floor plan, we consider any particular evacuation route as a directed network path. Its end node represents a building or floor exit, while its start node refers to an evacuation origin. All nodes in our testing floor plan have been manually digitized using OpenJUMP desktop GIS (see Figure 2). Nevertheless, since our data model supports the use of spatial analysis and computational geometry, we propose to automate node location. As a general rule, nodes will match access points. As mentioned above, we are dealing with a topologically consistent 2D polygon map. On such a map, an access can be defined as the one-dimensional intersection between neighbouring polygons which delimit enclosures whose category use identifiers will usually differ. Put simply, the segment shared between an office and a hallway can be considered an access to that office. Similarly the segment shared between a hallway and a staircase can represent a fire exit. Candidate nodes will lie on the center of each shared segment. We consider this a straightforward geoprocess where enclosures are visited in pairs and tested for access point calculation. A node type flag (returned by a use category 2D matrix), a relational predicate query, a geometry intersection and a midpoint calculation would be the building blocks of this node location

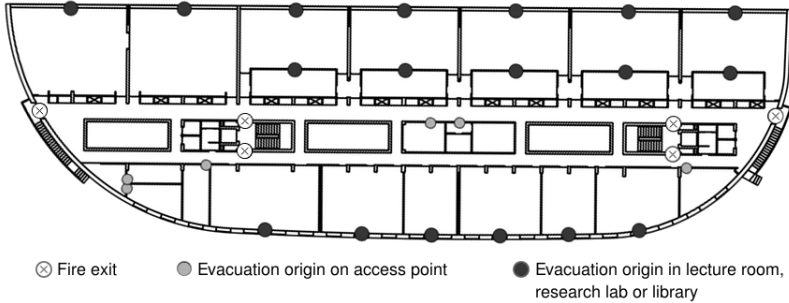


Figure 2: Manually digitized emergency exit and evacuation origin points.

algorithm. You can see a simplified proposal for a node location algorithm at <https://gist.github.com/4589233>.

However, DB-SI poses an exception to this rule: given an evacuation origin that triggers high outcomes of evacuees (a room likely to be at full-capacity, as said above), its node corresponds to the farthest trafficable location from the access point. These kind of point queries are better managed by spatial sorting methods such as Z-ordering, which is widely used to solve the nearest neighbour problem or to build hierarchical spatial indexes. The Z-order space-filling curve is a spatial hashing function: for each Cartesian coordinate an integer z-value is calculated by bit interleaving. Once ordered, the sequence of z-values preserves proximity relationships [13–15]. This means that z-values that are next to each other in the sequence most probably correspond to spatially close coordinates. We propose to use Z-ordering inversely to solve the farthest point query since, although results may be suboptimal, locations will be accurate enough and calculated in an efficient manner. We developed a simple Z-ordering tool which implements bit interleaving and generates z-values on a series of equidistant vertices along a polygon boundary (Source code used for this Z-ordering example is downloadable at <https://gist.github.com/4596403>). This program makes use of NetTopologySuite (an open source spatial analysis library) in order to compute geometries. Figure 3 illustrates this process on a polygon with 6 vertices from which 30 vertices (candidate node locations) are derived after a densify operation and labeled with their z-values. If we take vertex with z-value 1 as the access point, running a binary search on the sorted z-value list will return vertex with z-value 102 as the farthest candidate.

### 2.3 Network generation using medial axis transform

Once the main nodes of our network are generated, medial axis transform (also termed skeleton retraction or skeletonization) allows reducing 2D geometries to linear elements in order to achieve a suitable representation at smaller scales while preserving topology relationships. The idea behind this technique is to get a pattern

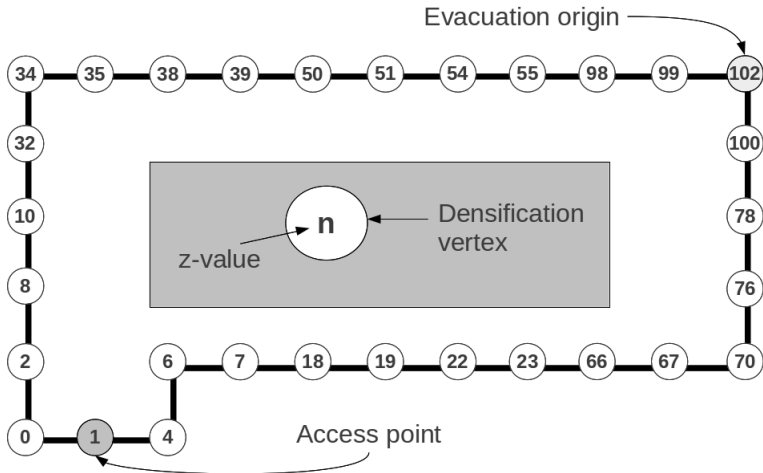


Figure 3: Z-order curve applied to a 2D polygon. Each vertex displays its z-value, obtained by bit interleaving x-coordinate and y-coordinate values. Given an access point, Z-order helps determine the evacuation origin point.

or skeleton that resembles the original in terms of continuity [16]. A fundamental approach for implementing this cartographic generalization technique is dealing with the medial axis as a by-product of Voronoi diagrams. This line of thinking can yield the skeleton of a polygon in a four-step process, as shown in Figure 4-a. Firstly, the polygon perimeter must be densified. The new geometry is a copy with a greater number of equidistant vertices that make up the point cloud from which Voronoi polygons are calculated. We are only interested in the section of the diagram which is covered by our target polygon. Hence, the polygon of interest must be overlapped with each polygon in the Voronoi diagram. Finally, the medial axis is retrieved by selecting those edges in the resulting polygon set which do not touch the perimeter of the target polygon. We wrote a small console application which implements this skeletonization algorithm using the Voronoi diagram builder provided by NetTopologySuite (Source code for the basic implementation of the skeletonization algorithm can be found at <https://gist.github.com/4605854>). Applying this technique on a building floor transition polygon, we could approximate its evacuation route network, but nodes would still be dangling and disconnected from predefined evacuation start and end nodes. Therefore, in order to obtain a comprehensive network, we resorted to Feature Skeletonizer, an OpenJUMP extension for cartographic generalization. This tool generates a Voronoi-based medial axis transform and links the derived skeleton to a set of input and outlet points which behave as connection nodes. Running Feature Skeletonizer on our transition polygon using the predefined nodes as input and outlet points yields a topologically correct network (see Figure 4-b) that makes evacuation routing and path length measurements feasible.

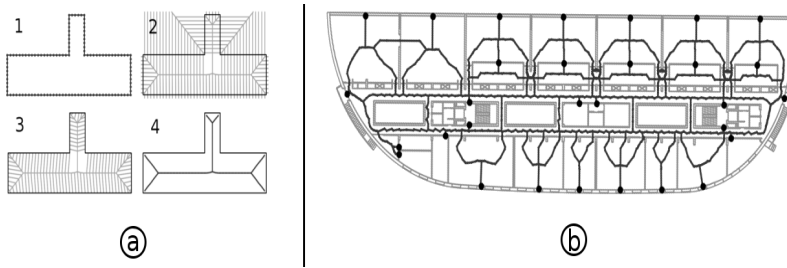


Figure 4: a) Step-by-step Voronoi based medial axis transform depicted graphically on a 2D polygon: (1) densification; (2) Voronoi diagram; (3) interior Voronoi polygons; (4) medial axis or skeleton b) Resulting network after applying medial axis transform on transition polygon and preserving edges connected to exit and evacuation origin nodes.

### 3 Route automation tests

As mentioned in the introduction, clustering of evacuation origins is a common practice among engineers when defining evacuation routes for a building floor. This is an heuristic process: different cluster configurations are tested in order to find one that best meets evacuee mandatory outcomes enforced on each emergency exit by design standards. All evacuation origins in a cluster are tried against the same emergency exit. The shortest path from each evacuation origin node to its exit node must be computed, along with the path length. If path length exceeds maximum distance allowed by prescriptive building codes, that node must be removed from the cluster. As for DB-SI, a density rate (expressed as number of people per area unit) applies to each evacuation origin according to the use category of the room it refers to. Its density rate and total room area determine expected evacuee outcome generated by a particular origin. Thus, the sum of all expected outcomes in a cluster must be less or equal to the maximum number of evacuees that its emergency exit supports.

When we focus on emergency exit load capacity, a simple state diagram such as the one in Figure 5 helps summarize this reasoning. This diagram suggests a state machine in which each route calculation triggers a transition, that is, a change in the accumulated load of the target exit. A persistence mechanism, so that states can be stored and retrieved at will, is also desirable.

An application prototype has been developed which provides a simple state machine that implements some useful functionality (source code at <https://gist.github.com/4633786>). A routing class is provided which takes advantage of QuickGraph, an open source graph computing library, for generating a non-directed graph from the set of connected linear geometries that make up our building floor skeleton. Upon this network structure, shortest paths between nodes are calculated using QuickGraph's implementation of Dijkstra's algorithm. This class also accepts a set of point geometries representing



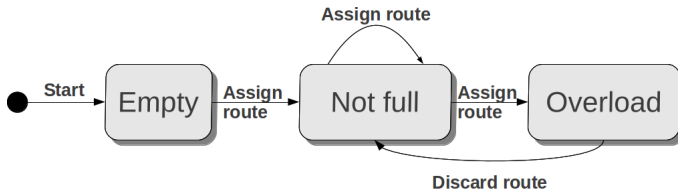


Figure 5: State diagram defining behaviour of a single emergency exit in the process of route assignment.

network nodes as input. Each point is automatically assigned an identifier which can be subsequently used to define a route query in the form  $start\_id, end\_id, z$ , where  $z$  is an integer value that equals the expected number of evacuees going from the evacuation origin  $start\_id$  to the emergency exit  $end\_id$ . Each single routing class instance is able to persist shortest paths and accumulated evacuee outcomes in memory, thus enabling a battery of working sessions to simulate one cluster configuration per session. In this sense, this can be viewed as an initial approach to an evacuation route automation programming interface.

In this example, we have 27 nodes in our test bed, three of them corresponding to fire exits (identifiers 0, 1 and 4). It involves configuring three clusters, thematically mapped in Figure 6, for which we have run a simulation. Clusters are configured following good practice criteria. These include prioritizing exits leading directly outdoors (identifiers 0 and 1) and defining compact distributions so that all evacuation origins in a cluster form a uniform cloud around their exit. From this single run, we have encountered two exceptions in the test results. One refers to path length between evacuation origin 15 and exit 1, which is greater than the 150 metres maximum allowed by DB-SI for this building. The other is related to total number of evacuees accumulated at exit 4 (159) exceeding the 156 maximum calculated under DB-SI standards. This information can be readily obtained by the engineer and used to gradually refine cluster configurations more efficiently.

## 4 Applicability for engineering projects design

Tests results show that the methodology applied and implemented in this research has advantages for evacuation planning as well as for earlier phases of building design, but there are also some issues that require further discussion.

Evacuation route layout according to fire safety regulations affects buildings' design. It is an issue that must already be addressed in preliminary plans, clearly stated in final working projects and updated in the event of functional and spatial rearrangements. Buildings change, and so do building codes and safety regulations in order to better protect their users. A fundamental point arising from this evidence is the need for fluently transferring architectural project data into an information system. In the case of corporations that manage large, complex and scattered building assets, we advocate data integration in

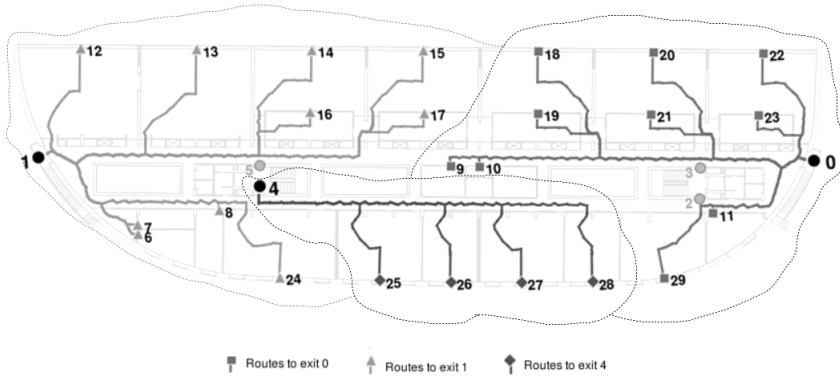


Figure 6: Test results on a particular evacuation origin cluster configuration. Clusters are configured for fire exits 0, 1 and 4. Exits 2, 3, and 5 are not considered.

a Geographical Information System (GIS). Not surprisingly, some intermediate tasks during project design which are to be discarded, such as enclosure outlining and use classification for area calculations, are indeed valuable GIS inputs. As we have demonstrated, an outstanding return of this integrated approach is that the information system handles buildings' geometry and function in a data model which supports high levels of automation and flexible development for evacuation route planning.

We consider this research to be a contribution for bridging the gap between Building Engineering and geographic databases. With regard to data acquisition, the usefulness of open interoperability protocols such as WFS has been proven. This is a key point from a user's perspective, since database implementation details remain in the background and practitioners obtain precisely what they need: hierarchically-coded polygon geometries with attribute data related to use category. This typical vector GIS data structure consequently becomes the foundation for solving a common engineering problem: evacuation route calculation. From a developer's perspective, the fact that the whole evacuation route network can be automatically generated from such a simple data structure must be highlighted. It is significant that innovation lies in standard geoprocessing rather than in an ad hoc algorithm. In this sense, all operations in the proposed sequence are well-known and widely used in cartography, spatial analysis or computational geometry. These can be summarized in classification queries, topological binary operators and relational predicates, spatial sorting using Z-order curve and Voronoi-based medial axis transform. The resulting network is suitable for shortest path calculation between pairs of predefined nodes, although the number of vertices should be reduced by means of a topology preserving simplify operation, thus rendering edges that are less sinuous and reducing distance distortion. Finally, we propose an automated testing environment to help

practitioners in the process of evacuation route planning. Tests are cluster-based and built upon the concept of the building floor being a state machine in which different groups of evacuation origins produce different distributions of evacuees in their assigned exits. These distributions may be accepted or rejected according to applicable safety regulations. Hence, the proposal focuses on finding a solution for each particular building floor level by simulating and gradually refining cluster configurations of evacuation origins.

This is obviously not a comprehensive solution since the testing environment would ideally treat the building as a whole. This is a work in progress and we definitely think there is room for future research, particularly regarding dynamic clustering so that globally acceptable configurations can be computed in advance instead of being manually defined.

## References

- [1] Gwynne, S., Galea, E., Owen, M., Lawrence, P. & Filippidis, L., A review of the methodologies used in the computer simulation of evacuation from the built environment. *Building and Environment*, **34(6)**, pp. 741–749, 1999.
- [2] Zheng, X., Zhong, T. & Liu, M., Modeling crowd evacuation of a building based on seven methodological approaches. *Building and Environment*, **44(3)**, pp. 437–445, 2009.
- [3] Zhang, L., Wang, Y., Shi, H. & Zhang, L., Modeling and analyzing 3D complex building interiors for effective evacuation simulations. *Fire Safety Journal*, **53**, pp. 1–12, 2012.
- [4] Vanclooster, A., Neutens, T., Fack, V., Van de Weghe, N. & De Maeyer, P., Measuring the exitability of buildings: A new perspective on indoor accessibility. *Applied Geography*, **34**, pp. 507–518, 2012.
- [5] Averill, J.D., Five grand challenges in Pedestrian and Evacuation Dynamics. *5th International Pedestrian and Evacuation Dynamics Conference*, eds. R.D. Peacock, E.D. Kuligowski & J.D. Averill, Springer US: Boston, MA, pp. 1–11, 2011.
- [6] Kuligowski, E.D., Modeling Human Behavior during Building Fires. Technical report, National Institute of Standards and Technology, Gaithersburg, MD, 2008.
- [7] Yuan, J., Fang, Z., Wang, Y., Lo, S. & Wang, P., Integrated network approach of evacuation simulation for large complex buildings. *Fire Safety Journal*, **44(2)**, pp. 266–275, 2009.
- [8] Dirección General de Arquitectura y Política de Vivienda, Documento Básico SI. Seguridad en caso de incendio. Technical report, Ministerio de Fomento, Madrid, 2010.
- [9] Dimova, S., Pinto, A., Oztas, A., Geradin, M. & Altinyollar, A., Identification of needs for improved fire protection by use of the eurocodes. Technical report, Joint Research Center, Ispra (VA), Italy, 2007.



- [10] Mira Martínez, J., Navarro Carrión, J. & Ramón Morte, A., SIGUA: SIG libre para la gestión del suelo de la Universidad de Alicante. *I Jornadas de SIG libre*, Universitat de Girona: Girona, p. 13, 2007.
- [11] OGC, Web Feature Service Implementation Specification. Technical report, Open Geospatial Consortium, Inc, Wayland, 2005.
- [12] OGC, Geography Markup Language (GML) Encoding Standard. Technical report, Open Geospatial Consortium, Inc, Wayland, 2007.
- [13] Gaede, V. & Gunther, O., Multidimensional Access Methods. *ACM Computing Surveys (CSUR)*, **30(2)**, pp. 170–231, 1998.
- [14] Lawder, J.K. & King, P.J.H., Using Space-filling Curves for Multi-dimensional Indexing. *Advances in Databases: proceedings of the 17th British National Conference on Databases (BNCOD 17)*, eds. B. Lings & K. Jeffery, Springer-Verlag: Berlin, pp. 20–35, 2000.
- [15] Connor, M. & Kumar, P., Fast construction of k-nearest neighbor graphs for point clouds. *IEEE transactions on visualization and computer graphics*, **16(4)**, pp. 599–608, 2010.
- [16] Gold, C. & Thibault, D., Map generalization by skeleton retraction. *Mapping the 21st century: 20th Int. Cartographic Conf.*, Int. Cartographic Assoc.: Beijing, pp. 2072–2081, 2001.

