

TWO-GRAPH BUILDING INTERIOR REPRESENTATION FOR EMERGENCY RESPONSE APPLICATIONS

P. Boguslawski^{a, *}, L. Mahdjoubi^a, V. Zverovich^a, F. Fadli^b

^a Department of Architecture and the Built Environment, Faculty of Environment and Technology, University of the West of England, Bristol, BS16 1QY United Kingdom - (Pawel.Boguslawski, Lamine.Mahdjoubi, Vadim.Zverovich)@uwe.ac.uk

^b Department of Architecture and Urban Planning, College of Engineering, Qatar University, Doha, Qatar – f.fadli@qu.edu.qa

Commission II, WG II/2

KEY WORDS: Indoor navigation, Emergency response, 3D modelling, Data models, Topological models, Spatial analysis

ABSTRACT:

Nowadays, in a rapidly developing urban environment with bigger and higher public buildings, disasters causing emergency situations and casualties are unavoidable. Preparedness and quick response are crucial issues saving human lives. Available information about an emergency scene, such as a building structure, helps for decision making and organizing rescue operations. Models supporting decision-making should be available in real, or near-real, time. Thus, good quality models that allow implementation of automated methods are highly desirable. This paper presents details of the recently developed method for automated generation of variable density navigable networks in a 3D indoor environment, including a full 3D topological model, which may be used not only for standard navigation but also for finding safe routes and simulating hazard and phenomena associated with disasters such as fire spread and heat transfer.

1. INTRODUCTION

Every day new buildings are constructed around the world. Demand for bigger public buildings is higher with rapid development of cities. Shopping centres, hospitals, airports are built to accommodate a big number of visitors. When they come to a building for the first time they usually do not know the exact location of a certain place and they need a piece of information how to get to: a particular shop in a shopping centre, a room in a hospital or a gate at the airport. For many it is not a big problem and they can spend some time to find the required location. However, in case of emergency situation everybody most likely would like to leave the building as soon as possible. It may be difficult if the evacuation routes and location of dangerous areas are unknown.

A similar situation applies to rescue teams, which come to the scene and need to get to people trapped in a building. Usually their knowledge about the threats and internal building structure is very limited, and they familiarize with the situation when they arrive to the scene. But the situation may change rapidly and even a small delay in the beginning of a rescue operation may cause death of people (DeCapua and Bhaduri, 2007).

Accurate information about the building structure, threats and critical issues delivered to rescue teams before they appear on the scene may save many human lives. An emergency response decision support system may deliver a preliminary solutions and help to plan the rescue operation.

The building structure for navigation purposes may be obtained explicitly from existing models, for example, CityGML (Gröger and Plümer, 2012) and Building Information Modelling (BIM) stored in the Industry Foundation Classes (IFC) format (ISO 16739:2013, 2013). It may be also reconstructed from Computer-Aided Design (CAD) (Lo et al., 2004) or scanned paper plans

(Atila et al., 2014; Boguslawski, 2011). Recently, spatial model reconstruction from point clouds has been widely investigated (Oesau et al., 2014; Volk et al., 2014). The resulting model is used to generate a navigable network for path finding applications. Another approach of indoor navigable routes reconstruction is based on tracking pedestrian trajectories using their smartphones (Alzantot and Youssef, 2012; Zhou et al., 2015a; Zhou et al., 2015b).

In this paper a gbXML model exported from BIM is used as input data for reconstruction of topology and navigable network. Some concepts implemented in the decision support system for emergency preparedness and response developed in this research are presented. The special focus is put on the representation of the internal building structure and navigable networks for hazard simulation and path-finding.

1.1 Navigable networks and path-finding

There are several models for indoor navigation developed recently. Kwan and Lee (2005) propose the Node-Relation structure. They apply the Straight Medial Axis Transformation for generation of a corridor skeleton and link adjacent rooms with the skeleton in order to create a network. Networks generated for each floor in a building are then connected into one navigable network by special links representing staircase connections. Liu and Zlatanova (2011) connect each pair doors within a corridor: directly by a link if the doors are in a direct visibility, or through concave corners otherwise. Lamarche and Donikian (2004) subdivide room floors using the Dalaunay triangulation. The edges of triangulation form a navigable network, which is used for animation. Another concept of floor partitioning into convex sub-regions was proposed by Stoffel et al. (2007). All these networks are used for indoor path-finding and navigation. However, only Kwan and Lee (2005) and Liu and Zlatanova (2011) focus on emergency response applications.

* Corresponding author

Navigable networks consist of nodes representing rooms or sub-regions of a room, and links connecting nodes. Links represent a navigable connection between two nodes. Usually, the shortest path-finding algorithms are applied on the network in order to find egress routes from a building. Crowd simulation may be performed to analyse evacuation bottlenecks (Choi and Lee, 2009). In case of an emergency event (e.g. fire, bomb explosion) nodes representing affected rooms are temporarily removed from the network and re-routing is performed (Lee and Zlatanova, 2008). However, the new path may not be a safe option, as the influence of the event on other rooms is not taken into consideration – there might be other, safer, paths possible to find, which are perhaps longer but avoiding dangerous areas. However, the aforementioned navigable networks are not suitable for analysis of the hazard influence and its spread in the building. There are missing non-navigable relations between adjacent rooms in a model, when they share a slab or wall without doors. Obviously, such connections are not used by pedestrians in a normal situation, but they are necessary to simulate some phenomena related to emergency. For example, a thin wall can delay the spread of fire, but finally fire will burn the wall and spread on adjacent rooms. Therefore, paths running through rooms in close proximity to hazard may not be the safest ones.

The following section describes research methodology supported by a real-life example of a building. Some aspects of a navigable network generation, hazard simulation and path finding are presented.

2. DECISION SUPPORT SYSTEM FOR EMERGENCY RESPONSE

In this paper a model of the Doha World Trade Centre (WTC) building in Qatar is used for construction of a navigable network. A typical floor (see Figure 1) is selected to show the details of the method. The most challenging aspect is to provide a reliable network within spaces, which are used intensively in indoor navigation, such as corridors. However, when the network of the whole building is generated, 3D model properties are taken in consideration and all topological connections between adjacent spaces are reconstructed: not only horizontal connections within one floor, but also connections between spaces at subsequent floors. They are reflected in a logical network of the building, which is accompanied by detailed networks within corridors and rooms having many doors and complex shape.

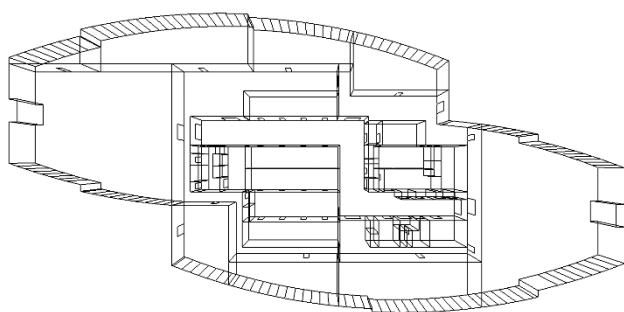


Figure 1. 3D model – one floor from Doha WTC building in Qatar.

A BIM model of the building is reconstructed from CAD plans and point clouds (Barki et al., 2015). It is then exported to the gbXML format using Autodesk Revit. The reason of conversion to gbXML is that the BIM model is too detailed, which makes spatial analysis inefficient due to geometrical complexity. Simplification of the model is therefore required, where some

geometry generalization algorithms may be applied. However, for the purpose of this research, gbXML is considered as a simplified model, which is automatically generated by Autodesk Revit. It may require additional validation to preserve consistency between the geometry and topology, such as, detection of missing faces or overlapping between adjacent rooms. Then, the model is represented with the dual half-edge (DHE) data structure (Boguslawski, 2011) in an in-house software application developed for testing purposes. DHE is suitable for simultaneous storage of the geometry and topology of a model represented as a cell complex, where rooms are cells (volumes) connected with the adjacent rooms by the dual structure – the graph of connections.

2.1 Variable Density Network

The process of network generation starts with reconstruction of a graph of connections between adjacent spaces. This graph is the dual to the primal graph, which represents the model geometry. Each space, e.g. room, corridor, is a cell in the primal structure associated with the unique dual node. Dual nodes for adjacent rooms are connected by an edge – dual edges are associated with primal faces bounding a cell. The dual graph reflects the topology of the building and is called a logical network (see Figure 2) and will be denoted by G .

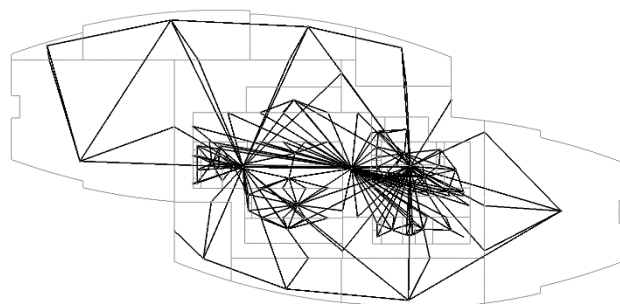


Figure 2. Building logical network.

In this research the logical network is used for simulating hazard events. However, it is not sufficient for path-finding purposes (Boguslawski et al., in press; Kwan and Lee, 2005). More detailed navigable network within spaces with many doors and complex geometry is required. The Variable Density Network (VDN) method proposed by Boguslawski et al. (in press) is applied. The method is based on Voronoi Diagram (VD) generation, which tessellates floors into a set of adjacent cells. Dual nodes associated with doors projected on the floor surface and concave corners from the primal structure are initial nucleation points for Voronoi cells, while additional points are introduced later in order to densify VDN. VD is represented as a graph, where the dual graph forms a network of navigable connections.

Density of the network is defined by two parameters: maximum length of network links ($threshold_1$) and maximum tessellation level (TL_{max}). Densification is an iterative process, where new nodes are added to VD at the middle of an existing link between adjacent cells if the link is longer than the threshold. The iteration is performed until there is no links to divide or the maximum number of iterations (TL_{max}) is reached. Denser tessellation may be performed locally in an area close to a hazard event and is defined by an additional parameter ($threshold_2$). The thresholds depend on application and may be specified independently for individual rooms. In the example shown in Figure 3, different values for $threshold_1$, $threshold_2$ and TL_{max} were tested in order

to validate the method. A denser VDN was generated for corridors ($threshold_1=2m, TL_{max}=10$) while within rooms it is sparse ($threshold_1=12m, TL_{max}=6$). Rooms with only one door are not processed and connections to these rooms defined in the original logical network are used for navigation. A hazard event is located in the most right room; thus, the network is locally densified in order to perform precise hazard simulation in the affected area ($threshold_2=0.7m$). The aforementioned parameters are assigned to the dual nodes, i.e. the nodes of G .

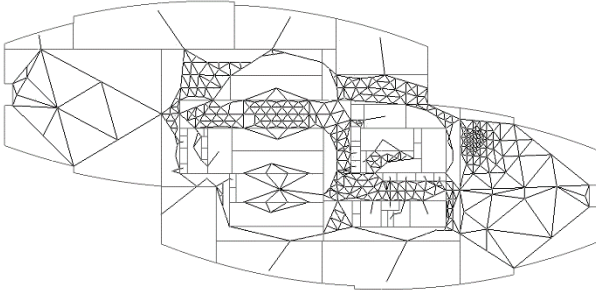


Figure 3. Navigable network

The graph G together with VDN generated for individual spaces will be denoted by G^+ . It should be noted that the navigable network G^+ consists of several graphs: G (logical network) and additional networks, $G_1 - G_n$, generated within some spaces. They do not form one connected graph: additional networks are linked with G by a reference to a root link of G_i , which is attached to a dual node representing i -space. There are points in G and G_i , which are paired, and thus can be considered as one node connecting these two graphs. These points are, in a simple case, door's dual nodes, which are used in VDN generation as initial nucleation points.

2.2 Hazard Simulation

Because all topological relations among spaces in a building are reflected in the dual of the logical network (G), it is possible to simulate phenomena associated with emergency events, such as fire spread, temperature distribution, etc. Navigable networks, which reflect only pedestrian routes have a disadvantage, that a number of phenomena, which can be simulated, is limited. For instance, smoke or gas is spread through similar routes like used by people, therefore it is possible to simulate them. However, fire can propagate through walls and slabs, and these routes are not reflected in those navigable networks.

Hazard simulation in this research is based on the idea proposed by Zverovich et al. (2016). Abstract hazard may be located in one or more indoor spaces. Its influence on the rest of a building is calculated based on distribution of hazard proximity numbers HP , which take into consideration a distance d and number of obstacles b (i.e. walls, doors and slabs) from the hazard epicentres. The highest HP is 100 at the hazard epicentre and is reduced with the distance from the epicentre and the number of obstacles between a current location and the epicentre. The number of obstacles can be easily calculated in G .

The procedure starts from a node representing a room with a hazard epicentre ($b=0$). For neighbours of the initial node we have $b=1$. In the iterative process, the next level of neighbours is calculated and b is increased by 1 for subsequent levels. The process stops when the parameter b is calculated for all nodes in G . The only exception are doors, where b is the smaller value of b from two adjacent rooms.

HP is calculated for each node in G^+ based on the formula:

$$HP_i = \frac{100}{\tau \sqrt{d_i \times b_i}},$$

where: HP_i – hazard proximity number calculated for i -node; τ – propagation coefficient; d_i – direct distance from the hazard epicentre to the i -node; b_i – number of obstacles from the hazard epicentre.

HP numbers are used to calculate link weights in G^+ , which reflect the degree of proximity to hazard:

$$HD_{uv} = 0.5 (HP_u + HP_v) \times D_{uv},$$

where HD_{uv} – a weight for a link between nodes u and v ; HP_u , HP_v – hazard proximity numbers respectively for nodes u and v ; D_{uv} – length of the link.

An algorithmic description of the above procedure is presented below.

Algorithm for Hazard Numbers: Calculation of hazard proximity numbers

Input: The graphs G and G^+ ; nodes $z_i, i=1,2,\dots,k$.

The propagation coefficient $\tau \geq 1$.

Output: Hazard proximity numbers $HD(e)$ for all links in G^+ .

```

foreach node  $z_i$  do
  foreach node  $v \in V(G^+)$  do
    calculate  $d(v, z_i)$  // This calculates the direct distance from
    //  $v$  to  $z_i$  in meters.
  end
  BreadthFirstSearchAlgorithm( $G, z_i$ )  $\mapsto b(v, z_i)$  for all nodes  $v$ 
  // The BFS algorithm is run in the graph  $G$  from node  $z_i$ .
  // It returns the number of links in shortest ( $v, z_i$ )-paths in  $G$ 
  // for all nodes  $v$ , i.e. the minimum number of obstructions
  //  $b(v, z_i)$  between  $v$  and  $z_i$ .
  foreach node  $v \in V(G)$  do
    if  $v$  represents a door do // Adjustment of  $b(v, z_i)$  for doors
      if  $v$  is an exit door do
         $b(v, z_i) = b(r, z_i)$  where the node  $r$  represents a room
        adjacent to  $v$ 
      else //  $v$  is a door, but not an exit door
         $b(v, z_i) = \min\{b(r_1, z_i), b(r_2, z_i)\}$  where  $r_1, r_2$  represent two
        rooms adjacent to  $v$ 
      endif
    endif
  end
end
foreach node  $v \in V(G^+) - V(G)$  do
   $b(v, z_i) = b(w, z_i)$  where  $w \in V(G)$  represents the cell whose
  tessellation contains  $v$ 
end
end

foreach  $i=1,2,\dots,k$  do
  foreach node  $v \in V(G^+)$  do
    calculate  $HP_i(v) = \frac{100}{\tau \sqrt{d(v, z_i) \times b(v, z_i)}}$ 
  end
end

foreach node  $v \in V(G^+)$  do
   $HP(v) = \max_{1 \leq i \leq k} HP_i(v)$ 
end

```

```

foreach link  $e=uv \in E(G^+)$  do // Calculation of hazard proximity
    // numbers for links
     $HP(e) = 0.5[HP(u) + HP(v)]$ 
     $HD(e) = HP(e) \times D(e)$  //  $D(e)$  is the length of  $e$  in meters.
end
    
```

The path-finding algorithm presented in the following section finds the safest path if these weights (HD) are used in step 11.a. It should be noted, that different paths may be obtained depending on propagation coefficient τ used in weight calculation. For $\tau = 1$, which is a minimum value, the safest path is the same as the shortest path, because $HP = 100$ for each node of the network. Then a binary search is performed in order to find higher values of τ causing the change of the path. The safest paths are shown in Section 3.

2.3 Path-finding

Any graph algorithm may be applied on the network presented in the previous section. In this research, for the sake of presentation clarity, we use a simple path-finding algorithm – Dijkstra’s algorithm – but other, more efficient algorithms can be readily implemented. The standard Dijkstra’s algorithm finds the shortest path in a graph based on distance weights associated with each link. However, in the presented research, except for the shortest paths, the safest paths are calculated using the same algorithms but with different weights. The weight used for the shortest paths is the distance between two nodes, which is attached to the link between these nodes. In case of the safest path the weight is calculated based on HP and the link length (Zverovich et al., 2016). It should be noted that the result is not the one safest path, but a set of safest paths depending on the τ parameter used in HP calculation, where the safest path calculated for $\tau = 1.0$ is the same as the shortest path, and the safest path runs further from the hazard location with higher τ .

Because the graphs G and G_i are stored separately, the path-finding algorithm requires modifications (see the path-finding algorithm below).

Path-finding algorithm (based on Dijkstra’s algorithm)

Input: *start*: dual source node representing a space
destinationName: name of the destination node (stopping criterion)

Output: path from *start* to *destination* node defined by the *destinationName* attribute

1. $destinationStop := nil$
2. Set attributes $start.distanceToSource := 0.0$ and $start.previousNode := nil$
3. Add *start* to *toVisit* list
4. If $start.G_i \neq nil$ then repeat Steps 2 and 3 for one of a selected nodes in the G_i (it may be an arbitrary node, e.g. closest to visited)
5. Take the first node from *toVisit*, denote it as *visited*, and remove it from the list
6. If $visited.G_i = nil$ then attribute $visited.visited := True$
7. If $visited = destinationStop$ then $destination := visited$ and goto 13
8. If $visited.Name = destinationName$ then
 - a. if $visited.G_i = nil$ then $destination := visited$ and goto 13
 - b. if $destinationStop \neq nil$ goto 5
 - c. $destinationStop :=$ select a node from $visited.G_i$ (it may be an arbitrary node, e.g. closest to visited)
 - d. if $destinationStop.visited = True$ then $destination := destinationStop$ and goto 13
 - e. goto 5

9. If $visited.G_i \neq nil$ then goto 5
 10. Add all neighbours of *visited* to the *neighbours* list; if *visited* is a node in G_i paired with a door node in G then add the door node to *neighbours*
 11. For each node *current* from *neighbours* which $current.visited = False$
 - a. Calculate *weight* of a link between *current* and *visited*
 - b. If $current.G_i \neq nil$ then
 - i. $current.distanceToSource := 0.0$, $current.previousNode := visited$, add *current* to *toVisit*
 - ii. $current :=$ get a node in G_i paired with a door represented by *visited*
 - iii. if $current.visited = True$ goto 11
 - c. if $current.distanceToSource = nil$ or $current.distanceToSource > weight + visited.distanceToSource$ then $current.distanceToSource := weight + visited.distanceToSource$
 - d. add *current* to *toVisit*
 12. Sort *toVisit* in ascending order by the *distanceToSource* attribute
 13. The path from *start* to *destination* is found
-

Paths generated by the above algorithms are shown in the following section. A hazard location and its spread in the building are taken into consideration in the path-finding process. Beside the shortest path, the safest paths avoiding dangerous areas are calculated.

3. RESULTS

Figure 4 and 5 show the result of path finding within the Doha WTC building in Qatar. There is a hazard event, e.g. fire, located on the ground floor. A mesh generated in the room with the hazard is densified locally around its epicentre. The shortest path from the affected room to the closest exit shown in Figure 4a runs very close to the hazard epicentre. The safer path ($\tau = 1.141$) shown in Figure 4b avoids the dangerous area – another door in the affected room is used. The path is slightly longer than the shortest one (23.6 m vs. 21.6 m), but after leaving the room the path runs in the area isolated from the hazard by a wall. For some hazard types, e.g. risk of bomb explosion, it may be too risky to use this path. The path shown in Figure 4c ($\tau = 1.354$) is safer, but longer (55.5 m). A different exit is used and the path runs much further from the hazard epicentre than the previous ones.

The density of the network within corridors and rooms, determined by $threshold_1=2$ m and 12 m respectively, is sufficient to perform path finding for pedestrians. Usually, there is no need to introduce a very dense network within rooms, because only a simple path from a defined location in a room to a door is calculated. However, in case of corridors, denser network is justified, because corridors are crucial ‘flow channels’ in a building with possible junctions and bends. With higher network density, the length of calculated paths is closer to actual paths used by pedestrians (Boguslawski et al., in press). However, local densification around the hazard epicentre does not have a significant influence on the resulting paths and was introduced in the model to show the flexibility of the proposed method. It is more important for better simulation of hazard propagation.

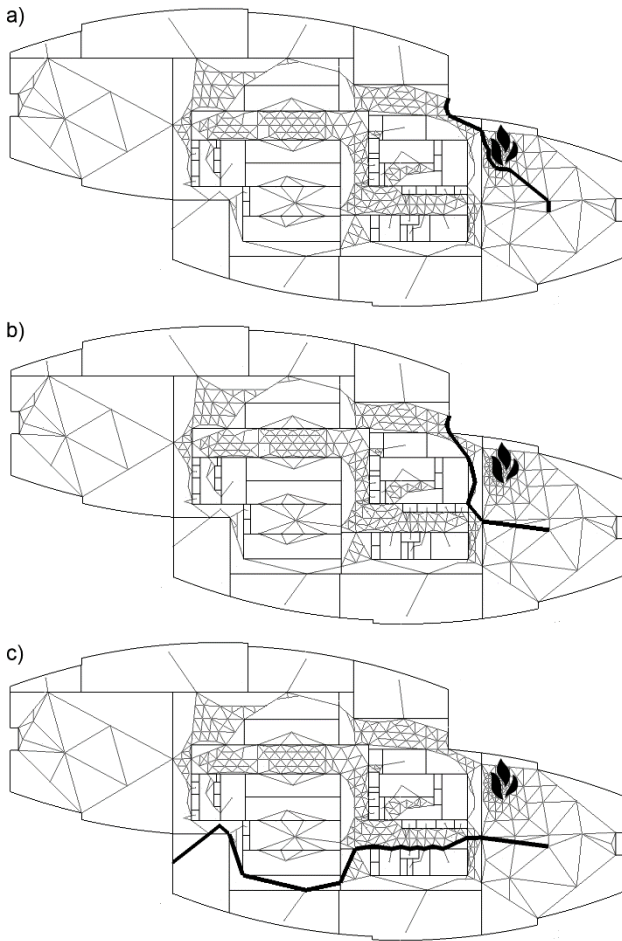


Figure 4. Paths from the selected room with the hazard event to an exit from the building: a) the shortest path ($\tau = 1.0$); b) short and safer path ($\tau = 1.141$); c) more safer but longer path ($\tau = 1.354$).

In the second scenario, there is another hazard epicentre introduced in the model. It is located close to the path shown in Figure 4c. A different path is then calculated ($\tau = 1.720$) as shown in Figure 5. The path is avoiding two dangerous areas and thus can be considered to be safer than other paths.

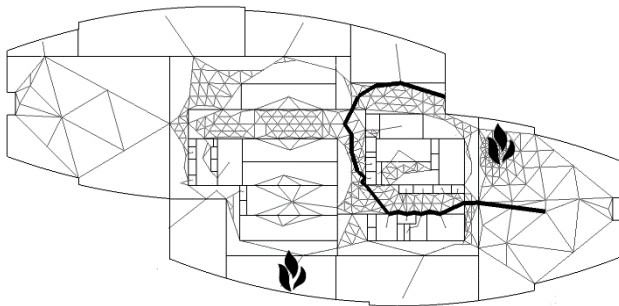


Figure 5. The safest path for two hazard epicentres ($\tau = 1.720$).

In the last example, three typical floors of the aforementioned building are used. There are two staircases in the building which allow for vertical navigation. One hazard epicentre is located on the ground floor. The shortest and safest paths are shown in Figure 6a and Figure 6b, respectively. The length of the shortest path is 63.6 m, while the latter one is 92.7 m long.

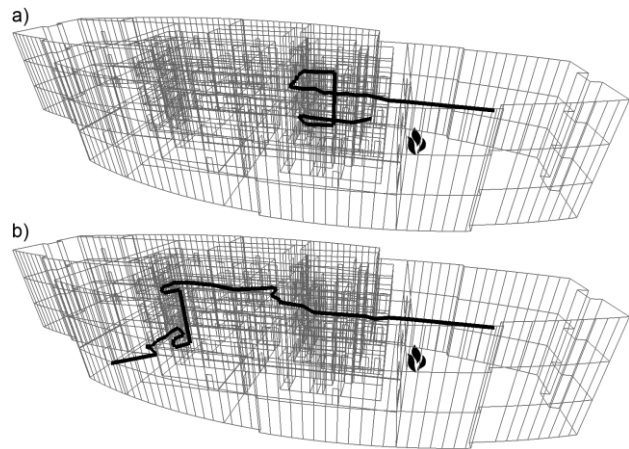


Figure 6. Paths from the selected room (3rd floor) to an exit from the building: a) the shortest path; b) the safest path.

4. CONCLUSIONS

In the paper, the details of the Variable Density Network for emergency response applications were presented, with a special focus put on the network representation. The resulting network G^+ is formed by several graphs: the logical network G and additional dense networks for selected indoor spaces G_i . The graphs G_i stored separately from G may be seen as a drawback reducing efficiency of path-finding algorithms. If the computation efficiency is crucial, e.g. in real-time applications, the graph G^+ may be exported from the model and stored as one unified graph. However, the aforementioned graphs are stored separately in the model, because G is an integral part of the spatial model represented with DHE, and it reflects the logical building structure, which is modified automatically with the change of the geometry. Also, G_i may be recalculated for a single room when required without updating the whole graph G^+ .

The length of calculated paths should be closer to the actual length of paths used by pedestrians when calculations are performed on a denser network. The density may be increased depending on application requirements, for instance, path planning for robot motion may require much denser networks than path-finding for pedestrians, where only rough estimation is usually required.

A path-finding algorithm proposed in this research is based on Dijkstra's algorithm and can be applied on G^+ for finding the shortest and safest paths. It was achieved by using different sets of weights reflecting searching criteria. In case of safest paths, no nodes affected by hazardous event were removed from the network to avoid navigation through them. The hazard was propagated in the model and the paths avoiding higher risk were calculated.

The future work will include calculation of the simplest paths, where complexity of the path is increased by staircases, doors and turning angles. Then, the multi-attribute decision-making technique will be developed in order to find 'the best path', which is reasonably short, safe and simple.

Another improvement anticipated in a near future is introduction of material fire rating of building elements, i.e. walls and slabs, as well as openings, i.e. doors and windows. It will be taken into consideration in calculation of hazard numbers.

ACKNOWLEDGEMENTS

This research/publication was made possible by a National Priority Research Program NPRP award [NPRP-06-1208-2-492] from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors.

REFERENCES

- Alzantot, M. and Youssef, M., 2012. CrowdInside: automatic construction of indoor floorplans, Proceedings of the 20th International Conference on Advances in Geographic Information Systems. ACM, Redondo Beach, California, pp. 99-108.
- Atila, U., Karas, I.R., Turan, M.K. and Rahman, A.A., 2014. Automatic Generation of 3D Networks in CityGML and Design of an Intelligent Individual Evacuation Model for Building Fires Within the Scope of 3D GIS. In: U. Isikdag (Editor), Innovations in 3D Geo-Information Sciences. Springer International Publishing, Cham, pp. 123-142.
- Barki, H., Fadli, F., Boguslawski, P., Mahdjoubi, L. and Shaat, A., 2015. 3D BIM models generation from 2D CAD drawings and 3D scans: a concrete case study and challenges analysis for AEC practitioners. In: L. Mahdjoubi, C.A. Brebbia and R. Laing (Editors), Building Information Modelling (BIM) in Design, Construction and Operations, BIM 2015. WIT Press, Bristol, United Kingdom, pp. 369-380.
- Boguslawski, P., 2011. Modelling and Analysing 3D Building Interiors with the Dual Half-Edge Data Structure. PhD Thesis, University of Glamorgan, Pontypridd, Wales, UK, 134 pp.
- Boguslawski, P., Mahdjoubi, L., Zverovich, V., Barki, H. and Fadli, F., in press. Automated Construction of Variable Density Navigable Networks in a 3D Indoor Environment for Emergency Response. *Automation in Construction*.
- Choi, J. and Lee, J., 2009. 3D Geo-Network for Agent-based Building Evacuation Simulation. In: J. Lee and S. Zlatanova (Editors), 3D Geo-Information Sciences. Springer, pp. 283-299.
- DeCapua, C. and Bhaduri, B., 2007. Applications of Geospatial Technology in International Disasters and During Hurricane Katrina, Available at the Project Site of Capturing Hurricane Katrina Data For Analysis and Lessons-Learned Research (www.gri.msstate.edu/research/katrinalessons/Documents/GeoSp_Tech_Applications.pdf).
- Gröger, G. and Plümer, L., 2012. CityGML – Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71: 12-33.
- ISO 16739:2013, 2013. Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries.
- Kwan, M.-P. and Lee, J., 2005. Emergency response after 9/11: the potential of real-time 3D GIS for quick emergency response in micro-spatial environments. *Computers, Environment and Urban Systems*, 29: 93-113.
- Lamarche, F. and Donikian, S., 2004. Crowd of Virtual Humans: a New Approach for Real Time Navigation in Complex and Structured Environments. *Computer Graphics Forum*, 23(3): 509-518.
- Lee, J. and Zlatanova, S., 2008. A 3D data model and topological analyses for emergency response in urban areas. In: S. Zlatanova and J. Lee (Editors), *Geospatial Information Technology for Emergency Response*. Taylor & Francis, pp. 143-168.
- Liu, L. and Zlatanova, S., 2011. A "door-to-door" path-finding approach for indoor navigation, *Gi4DM 2011: GeoInformation for Disaster Management*. International Society for Photogrammetry and Remote Sensing (ISPRS), Antalya, Turkey.
- Lo, S.M., Fang, Z., Lin, P. and Zhi, G.S., 2004. An evacuation model: the SGEM package. *Fire Safety Journal*, 39(3): 169-190.
- Oesau, S., Lafarge, F. and Alliez, P., 2014. Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. *ISPRS Journal of Photogrammetry and Remote Sensing*, 90(0): 68-82.
- Stoffel, E.P., Lorenz, B. and Ohlbach, H.J., 2007. Towards a Semantic Spatial Model for Pedestrian Indoor Navigation Advances in Conceptual Modeling – Foundations and Applications. *Lecture Notes in Computer Science*, pp. 328-337.
- Volk, R., Stengel, J. and Schultmann, F., 2014. Building Information Modeling (BIM) for existing buildings — Literature review and future needs. *Automation in Construction*, 38(0): 109-127.
- Zhou, B., Li, Q., Mao, Q., Tu, W. and Zhang, X., 2015a. Activity Sequence-Based Indoor Pedestrian Localization Using Smartphones. *IEEE Transactions on Human-Machine Systems*, 45(5): 562-574.
- Zhou, B., Li, Q., Mao, Q., Tu, W., Zhang, X. and Chen, L., 2015b. ALIMC: Activity Landmark-Based Indoor Mapping via Crowdsourcing. *IEEE Transactions on Intelligent Transportation Systems*, 16(5): 2774-2785.
- Zverovich, V., Mahdjoubi, L., Boguslawski, P., Fadli, F. and Barki, H., 2016. Emergency Response in Complex Buildings: Automated Selection of Safest and Balanced Routes. *Computer-Aided Civil and Infrastructure Engineering*.