Integrating Sensing and Routing for Indoor Evacuation

Jing Wang[1,2], Stephan Winter[2], Daniel Langerenken[3,2], and Haifeng Zhao[2]

[1] School of Mathematics and Computer Science, Wuhan Polytechnic University, China
[2] Department of Infrastructure Engineering, The University of Melbourne, Australia
[3] Cognitive Systems Group, University of Bremen, Germany

Abstract. Indoor evacuation systems are needed for rescue and safety management. A particular challenge is real-time evacuation route planning for the trapped people. In this paper, an integrated model is proposed for indoor evacuation used on mobile phones. With the purpose of employing real-time sensor data as references for evacuation route calculation, this paper makes an attempt to convert sensor systems to sensor graphs and associate these sensor graphs with route graph. Based on the integration of sensing and routing, sensor tracking and risk aware evacuation routes are generated dynamically for evacuees. Experiments of the proposed model are illustrated in the paper. The benefit of the integrated model could extend to hastily and secure indoor evacuation and it potentially presents an approach to correlate environmental information to geospatial information for indoor application.

1 Introduction

Applying geographic information systems (GIS) for indoor environments has gained attention in way-finding contexts and in building facility management. The challenges for indoor GIS include the representation of indoor space for routing tasks and the integration of relevant building facilities. There is no standard representation for routing networks in indoor space until now because indoor space has multiple functional purposes which lead to different proper representations for the same indoor environment. Furthermore, people move more freely in indoor space than vehicles move in outdoor space [1]. Many data structures realize in a unique way the representation of the navigable space in an indoor environment, with varying advantages and disadvantages [2–5].

The prior research on indoor space representation especially focused on utilization of the routing capabilities of GIS in response to indoor disasters, such as fire, gas leakage or terror attacks. This research was also used for simulation of navigation in evacuation scenarios considering building structure and emergency propagation characters [6–12]. Generally, evacuation routes in these systems are built on route graphs which are originally derived from building floor plans and then dynamically updated in references to the predicted result of emergency simulation models or risk assessment.

However, in modern buildings different kinds of sensors obtain environmental parameters continuously, such that prediction can be replaced by real-time sensing. Some other evacuation applications take already advantage of wired or wireless building sensor systems to monitor a current environmental state for dynamic evacuation [13–15].

M. Duckham et al. (Eds.): GIScience 2014, LNCS 8728, pp. 268–283, 2014.

In contrast to open (outdoor) environments the detecting area of a sensor inside a building is limited by the building structure. For example, near sensors can be independent because of a wall in between. Thus, linking sensor readings to routes requires complex analysis. Additionally, wireless building sensor systems (WSN) rely on the ad-hoc communication network of the sensor vertices, and special routing protocols [16] are required to guarantee a near-real-time updating of evacuation routes.

The above review implies the need to identify an integrated data model that not only represents the routing properties of indoor environments, but also the observations of building sensor systems in their impact on routing. This data model should combine geospatial information such as route graphs and sensor graphs, and environmental information such as sensor readings to achieve real-time risk-aware evacuation planning in emergency situations. In this study we introduce such a data model for indoor evacuation. The suggested data model, implemented in a personalized evacuation system, will allow generating an evacuation route dynamically for each evacuee, depending on current sensor readings, and taking risk assessments into account. The integrated model enables indoor evacuation applications valuable for both accurate emergency localization and reliable evacuation guidance. The potential benefits of the model can also extend to offer an approach to correlate route network to sensor network for a wide range of indoor applications.

The paper is structured as follows. Section 2 presents related work. Section 3.1 introduces the method of formulating route graphs, and Section 3.2 propose an algorithm to create sensor graphs with completely coverage of all area in building floor plans. Section 3.3 explains the integration of route graphs and sensor graphs. Section 4 provides a sensor tracking and risk aware evacuation plan using this novel data model. Experiments of the model are discussed in Section 5. Section 6 concludes the current work and explores future work.

2 Related Work

Since many outdoor concepts of routing networks are not suitable for indoor environments, indoor routing models have been presented, such as a representation in *Indoor Geography Markup Language* (IndoorGML) [2], and the *Node-Relation-Structure* (NRS) [3]. In [6] a 3D geometric network model based on NRS is introduced to represent an indoor environment, before an ant colony optimization algorithm is applied to find the shortest path during evacuation. Pre-simulations deploying the *Fire Dynamic Simulation* (FDS) provided feedback of smoke spread in the system.

Han et al. have proposed an integrated real-time evacuation route planning method for high-rise building fires. The sensor data acquisition is also pre-simulated in FDS, and the data transmission is based on a wireless network. An interpolation method is applied to quantitatively assess a *casual risk* for each evacuee. Casual risk is based on the distance to the nearest exits [9].

Nguyen et al. proposed a fire evacuation model under the assumption that evacuees follow the boundaries of obstacles or walls to find the exits when their visibility is limited by smoke. In their agent-based model, they do not discuss the method to generate an evacuation route on account of smoke [17].

In [11] a *multi-node hierarchical data model* is proposed to extract route graphs by topological relationships among indoor units of a building. A hierarchical route planning algorithm dynamically schedules evacuation routes considering distance, risk level and waiting time for evacuees. In their approach route planning is risk aware, but no environmental features are involved.

In [14] a WSN-based safe route identification algorithm has been introduced which depends on a dense sensor network and wireless communication technologies, and which routes along the sensor network. A location-based routing protocol is applied to dynamically generate an evacuation route in which the building structure and risk prediction are ignored.

Stahl et al. developed YAMAMOTO (*Yet Another Map Modeling Toolkit*) to generate route graphs automatically from floor plans for multi-level buildings by adding route vertices at key points on the basis of the outlines of spatial regions [18–21]. The toolkit provides an approach for finding navigable routes that also allows for modeling environmental monitoring devices, such as sensors or actuators. In this paper, the integration of sensing and routing, and then also the simulated evacuation experiments are built on top of YAMAMOTO.

3 The Proposed Model

The model consists of three main parts, presenting a route graph for the indoor environment, sensor graphs for sensors in this environment, and the integration of the route graph with all sensor graphs.

3.1 Route Graph

The main purpose of route graph is to associate evacuation routes with the building structure to ensure the feasibility of the generated evacuation route. While in principle any of the suggested route graphs of Section 2 can be applied, the following is specifically based on the route graph of YAMAMOTO [21], which is specified formally here for the first time. For this formal specifications, some definitions are introduced before generating a route graph.

Definition 1. *A region R is a planar and accessible area with a unique identifier in the model, denoted as R_i.*

Thus a region in this paper refers to a two-dimensional bounded area with at least one entrance that could provide access for pedestrians, such as a room, corridor, or lobby. Entrances form virtual parts of the boundary of a region. Every two regions R_1 and R_2 must not overlap, but may neighbor, in which case they share at least one boundary line. Overlapping areas P and Q are separated into regions, either by $\{P,\ (P \cup Q) \cap \overline{P}\}$, or by $\{Q,\ (P \cup Q) \cap \overline{Q}\}$.

Every region is represented by a polygon. If a part of the boundaries of a region is curved, this curved part is substituted by a polyline, for example, using the successive bisection algorithm of [22] adapting to a chosen level of accuracy. For convex regions, each pair of vertices inside or on a boundary of the region can be linked together by a

straight line edge without piercing any boundary. These edges are *passable*. However, for concave regions the links between some pairs of vertices are not passable. For those pairs of vertices there exists always a set of passable edges between them, including a sequence of concave boundary vertices [23].

There are three types of vertices in a route graph, called *inner vertex*, *access vertex* and *connecting vertex*.

Definition 2. *An* inner vertex *is a shifted concave boundary vertex of concave regions.*

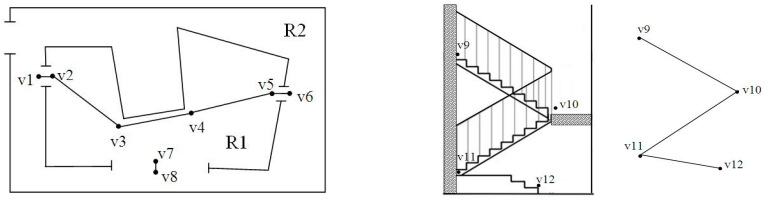
Inner vertices are used to form a passable edges of route graphs. They are placed one meter away from walls, boundaries or corners in order to keep a natural distance for pedestrians.

Definition 3. Access vertices *are placed pair-wise on both sides of the virtual boundary of an entrance.*

The edge between a pair of access vertices represents the accessibility of two adjoining regions in a route graph. The two vertices belong to different regions respectively. Also access vertices are placed one meter away from the virtual boundary of the entrance in order to keep a natural distance for pedestrians.

Definition 4. Connecting vertices *are placed at the beginning, end, and turning points of staircases.*

The edges between a sequence of connecting vertices are route graph elements connecting two different floors.



(a) Inner vertices and access vertices (b) Connecting vertices

Fig. 1. Three types of vertices in route graphs

In Figure 1(a), there are three pairs of access vertices between R_1 and R_2 among which v_2, v_5, and v_7 belong to R_1. A straight link between v_2 to v_5 is not passable in the concave region R_1, but one of the passable routes is $v_2 - v_3 - v_4 - v_5$ with v_3 and v_4 being inner vertices. Figure 1(b) shows four connecting vertices; v_{10} and v_{11} are placed at turning points of a staircase. Elevators, while connecting floors as well, are not represented in the route graph because they cannot be used for evacuation.

Definition 5. *A route graph $G = (V, E)$ consists of a set V of inner vertices, access vertices, and connecting vertices, together with a set E of edges. The set E consists of the edges between each pair of access vertices, the edges between every pair of vertices in the same region if the edge is passable, and the edges between every two consecutive connecting vertices along the same staircase.*

As an example, Figure 2 shows a floor plan of a building. The five regions are named as R_1 to R_5, and R_5 is a concave region. The corresponding route graph G is shown in Figure 3.

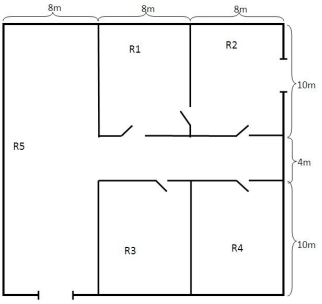


Fig. 2. Sample floor plan

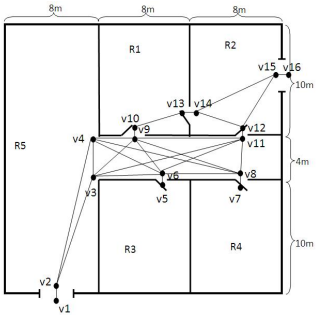


Fig. 3. Route graph

3.2 Sensor Graphs

In contrast to route graphs, the objective of sensor graphs is to give timely feedback of environmental factors in order to predict the impact on evacuation plans, including a risk assessment. The definition of a sensor graph is also based on some concepts that need to be defined first.

Definition 6. *A* sensor detecting area S_i *of sensor* s_i *is the collection of all points where sensor data can be obtained by sensor* s_i*.* S_i *is represented by a polygon with a clockwise ordered set of corner vertices* $S_i := \{p_1, \cdots, p_m\}$*,* $m \geq 3$*. Each vertex* p_k*,* $1 \leq k \leq m$*, is denoted by its position* (x_k, y_k, z_k)*.*

The typical detection range of most types of sensors is a circular area of radius r_s of which the center is the point of installation of the sensor. However, there are significant exceptions. For example, walls shield the reception of sensors [24]. Hence, the actual detection area of each sensor is the intersection between the uninhibited sensor detection area and the boundary of the region in which the sensor is installed. For example, in Figure 4 S_3 is the sensor detecting area of s_3 in R_1, thus, $S_3 := \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\}$.

Definition 7. S_i *of sensor* s_i *and* S_j *of sensor* s_j *is considered intersecting if (a)* s_i *and* s_j *are in the same region and* $S_i \cap S_j \neq \emptyset$*, or (b)* s_i *and* s_j *are in different regions and* $S_i \cap S_j$ *covers at least one entrance between the regions, or (c)* s_i *and* s_j *are at the opposite entrances of a staircase (i.e., on neighboring floors).*

Definition 8. *A sensor graph* $G' = (V', E')$ *consists of a set* V' *of sensor vertices* s_i *representing the location of the sensors, together with a set* E' *of edges if two sensors' detection areas are intersecting.*

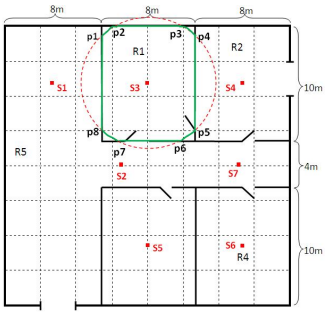


Fig. 4. A sensor detection area

When forming a sensor graph, there is a complete coverage problem of sensor vertices. According to the objective of sensor graphs, the sensor network should cover all areas of a floor plan, or otherwise events may be not detected, and also the event simulation becomes incoherent. Therefore, a sensor graph according to Definition 8 is completed by adopting an algorithm suggested originally in [25].

All together, the following steps describe the sensor graph generation.

Step 1: A floor plan is partitioned by a grid T, with equally spaced tiles $t_{p,q}$ (p is the row number and q is the column number). For those tiles that are divided by borders between regions, each part of the tile will be treated as separated tiles in different regions.

Step 2: The sensor vertices S are added to G' for each actual sensor in the environment.

Step 3: For a region R_i, a matrix $A(R_i) = [a_{p,q}]_{n \times m} (1 \leq p \leq n, 1 \leq q \leq m)$ is formed corresponding to the $n \times m$ tiles that just cover R_i and initialized by 0s. If there are some tiles in this matrix that are not part of the region the corresponding entries are set to infinite.

Step 4: For each tile $t_{p,q}$, if $t_{p,q} \cap S_j \neq \emptyset \, (1 \leq j \leq |V'|)$ in R_i the entry $a_{p,q}$ in $A(R_i)$ will be set to 1.

Step 5: If $A(R_i)$ contains one or more zero elements, i.e., tiles inside the region that are not observed by a sensor, a virtual sensor is placed at the center of the first zero tile encountered in row order, and added to V'. Then go to Step 4. Otherwise, go to Step 6.

Step 6: If all regions in the floor plan have been operated go to Step 7. Otherwise, go to Step 3.

Step 7, connecting floors: For each staircase connecting two adjacent floors, two virtual sensor vertices are added to V', which are placed in the middle of the entrances of the same staircase.

Step 8: For all pairs s_i, s_j, if S_i and S_j are intersecting add an edge between them to E'.

Step 9: If all floor plans in the building have been observed, stop. Otherwise, go to Step 1.

In Figure 5, the floor plan is firstly partitioned into tiles with an equal spacing, here of $r_s/2$ ($r_s = 6m$). The origin of the grid is at the top left corner. There are originally seven smoke sensors s_1 to s_7 in the sensor system, denoted by red squares.

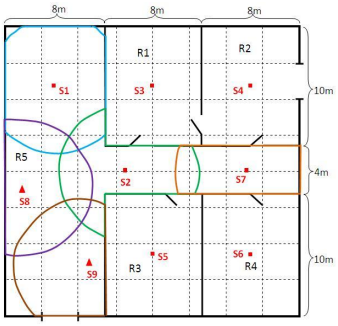


Fig. 5. Sensor detection areas in R_5

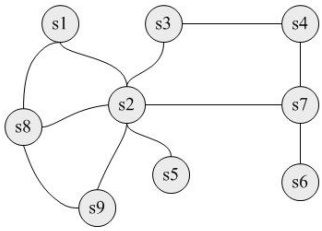


Fig. 6. Sensor graph

For R_1, the matrix $A(R_1)$ is 4×4 although some of the tiles in R_1 are only partly inside. Since s_3 is the only actual sensor in R_1 and all grids in R_1 intersect with S_3 (Figure 4), all entries in $A(R_1)$ equal to 1 and no virtual sensor vertex need to be added in R_1. R_2, R_3, and R_4 are similar, and S_4, S_5 and S_6 are added to V'.

For R_5, however, Figure 5 shows some gaps in the sensor coverage. There are three actual sensors s_1, s_2 and s_7 whose detection areas are denoted with lines of different colors. Corresponding $A(R_5)$ is

$$A(R_5) = \begin{bmatrix} 1 & 1 & 1 & \infty & \infty & \infty & \infty & \infty \\ 1 & 1 & 1 & \infty & \infty & \infty & \infty & \infty \\ 1 & 1 & 1 & \infty & \infty & \infty & \infty & \infty \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & \infty & \infty & \infty & \infty & \infty \\ 0 & 0 & 0 & \infty & \infty & \infty & \infty & \infty \\ 0 & 0 & 0 & \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

containing some zero values. Since the first zero value (in row order) refers to $t_{5,1}$ a virtual sensor s_8 is placed at its center, denoted by a red triangle, and S_8 is added, denoted by a purple line. After another iteration,

$$A(R_5) = \begin{bmatrix} 1 & 1 & 1 & \infty & \infty & \infty & \infty & \infty \\ 1 & 1 & 1 & \infty & \infty & \infty & \infty & \infty \\ 1 & 1 & 1 & \infty & \infty & \infty & \infty & \infty \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & \infty & \infty & \infty & \infty & \infty \\ 1 & 1 & 0 & \infty & \infty & \infty & \infty & \infty \\ 0 & 0 & 0 & \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

still contain some zero values. $t_{7,3}$ is the first zero value in search order, and a virtual sensor s_9 is added to V'. In the next iteration, $A(R_5)$ does no longer contain a zero value. The algorithm proceeds to Step 8 to generate edges E'. The final result of G' is shown in Figure 6.

This sensor graph generation algorithm is easy to program, but not optimal because the number of virtual sensors is not the minimum. For the purpose of this paper the algorithm is sufficient. In the future, another algorithm can be plugged in covering the whole area with a minimal number of virtual sensors.

Generally, when a building has different types of sensors simultaneously, separate sensor graphs are formed in the model.

3.3 Integration of Sensor Graph and Route Graph

The integration of sensor graph and route graph is actually a mapping from sensor network observations to route network elements. The integration facilitates the representation of *affectedness* of route graph vertices and edges, for passive and active sensors as well as for sensors at risk according to some prediction model. The following method integrates *one* sensor graph with a route graph. With other sensor graphs the same method can be applied.

If a vertex v_i in route graph G is inside S_k of some sensor s_k, vertex v_i is one of the affected vertices of s_k. Also, if an edge e_j in G intersects with S_k, edge e_j is one of the affected edges of s_k. The latter provides a mechanism to represent that an edge is blocked although the endpoints may not. The relationship between route graph vertices and sensors is $n : m$ since route graph vertices can be inside none, one or multiple S_i, and there may be zero, one or more route graph vertices in S_i. The same applies for relationships between route graph edges and sensors because a straight edge may go through more than one S_i and the range of a sensor may intersect with more than one route graph edge.

The integration of the sample floor plan is shown in Table 1. There are no affected route graph vertices or edges in S_1 which suggests that S_1 will not be crossed during evacuation. However, when s_8 is activated in an event, two route graph edges will be blocked (but no vertices are affected).

4 Sensor Tracking and Risk Aware Evacuation Planning

This section introduces evacuation planning based on the integrated route and sensor graph data model. It considers sensor states of being activated by an event, or at risk according to some prediction model. In emergency, evacuees who are familiar with the building tend to choose their habitual routes for evacuation although it may lead them to blocked or risky areas. Other evacuees who are unacquainted with the building may follow the crowd or the signed evacuation routes. Also these strategies may be in conflict with the event itself. Evacuation based on outdated information is *blind* evacuation. Recently, mobile application systems have emerged able to provide assistance for evacuees. Some of these applications are also blind, based on by the event outdated data. Their evacuation route planning is *static*, it does not adapt dynamically to the situation.

Table 1. Integration of sensor graph and route graph

Sensor	Affected vertices	Affected edges
s_1	$\emptyset$	$\emptyset$
s_2	v_3, v_4, v_6, v_9	all edges start or end on v_3, v_4, v_6, v_9
s_3	v_{10}, v_{13}	all edges start or end on v_{10}, v_{13}
s_4	v_{12}, v_{14}, v_{15}	all edges start or end on v_{12}, v_{14}, v_{15}
s_5	v_5	all edges start or end on v_5
s_6	v_7	all edges start or end on v_7
s_7	v_8, v_{11}	all edges start or end on v_8, v_{11}
s_8	$\emptyset$	v_2-v_4, v_2-v_3
s_9	v_2	all edges start or end on v_2

Even if these applications receive the initial location of the emergency (updating their data initially) they can not generate dynamically escape routes.

Few indoor evacuation systems are tracking sensors and risk aware. A *sensor tracking evacuation* is obtaining timely the sensor data from building sensor systems to generate real-time evacuation routes for evacuees. In addition, a *risk aware evacuation* also avoids or reduces risks after adopting some strategies to predict near-future states of the event.

The proposed model in this paper can be applied to realize sensor tracking and risk aware evacuation planning. There are two reasons. Firstly, sensor data is continuously read out by a building sensor system. When unusual situations are reported at a time (i.e., some sensors get activated), the identifiers of these sensors localize the event. If these identifiers are forwarded to an evacuation application, the affected route graph vertices and edges are directly accessible in the integrated model, and are marked as blocked when generating a real-time evacuation routes for individuals. Consequently, this application meets the requirements of sensor tracking evacuation planning. The application is also capable of risk awareness. When building sensor systems report an event (i.e., some sensors are activated), the connectivity in the sensor network can be used to predict the near-future spread of the event. Reporting risky sensors to the evacuation application allows to mark the affected route graph vertices and edges being at risk, e.g., by a risk rank value, which may be considered by some weighting in the computation of individual evacuation routes. Table 2 illustrates the differences among these three evacuation plans in which ST&RA refers to applying a sensor tracking and risk aware evacuation plan, and the symbol $\surd$ means the corresponding leftmost item is aware by the evacuation plan.

Both sensor tracking and risk awareness requires a standing communication between building sensor systems and evacuation planning applications. However, even only occasional updates and iterative evacuation planning must be superior to blind or static evacuation planning. This is tested in the following section.

Table 2. Difference among evacuation plans

	ST&RA	Static	Blind
Emergency location	√	√	×
Original route graph	√	√	Evacuee's experience
Real-time sensor reading	√	×	×
Risk analysis	√	×	×
Dynamic evacuation route generation	√	×	×

5 Experiments

This section demonstrates a mobile evacuation system application, implementing the proposed model, in an office building. The objective of the experiments is to compare sensor tracking and risk aware evacuation plans with blind and static evacuation plans. In this experiment a fire scenario is assumed.

5.1 System Implementation

The system implementation is built on a framework shown in Figure 7. There are six separately modules in the system. The building sensor system gathers continuously the environment data, and reports an alarm when it detects some unusually situation. The sensor graphs, the route graph, and their integration have been pre-calculated and stored in a server. On the user side, four modules cooperate with each other to generate a safe and short evacuation route for each evacuee. The location module gives the current position of each user. The risk assessment module evaluates the emergency situation and renews the risk rank value of affected vertices and edges in route graph. Path generation module sets an optimal evacuation path dynamically by considering both the distance and the risk value. The building model provides a platform to the user to follow the evacuation route and related instructions. The thick lines in Figure 7 denote the dynamic data and the filaments denote the pre-generated information.

The entire evacuation system is developed by expanding YAMAMOTO, in C#. Sensor graphs, route graph and the integration are all produced in a suited XML format, from planar floor plans. The current location of each evacuee is obtained by QR-code scanning which obviously can be substituted by any other positioning mechanism. The risk assessment module is accomplished by a series of risk assessment strategies synthesized from Dow's Fire and Explosion Index [26] and Fire Safety Evaluation System [27]. In particular, the risk assessment mechanism is an independent part of the entire application, and could also be easily replaced by any other risk assessment method. The path generation module is that of YAMAMOTO, but now using dynamic route graph of the proposed model. The user interface is built by Open Graphics Library for Embedded System. Figure 8 shows section of the office building model together with sensor detection areas, and Figure 9 gives the interface of the system.

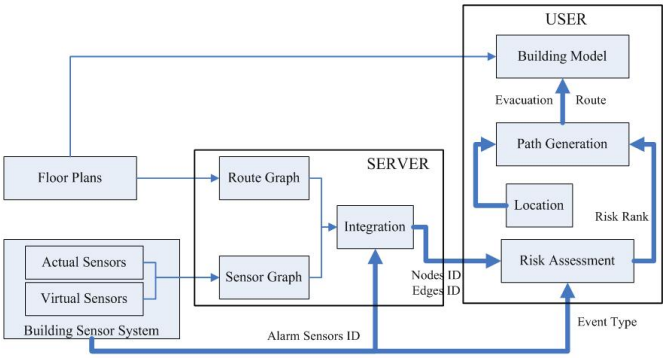


Fig. 7. System framework

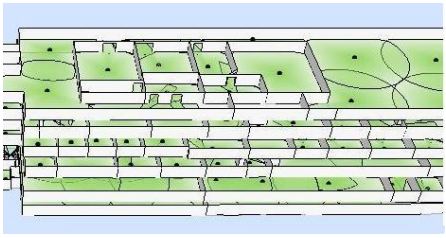


Fig. 8. Section of building model with sensor layers

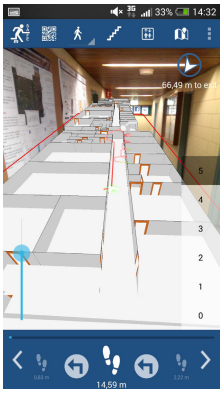


Fig. 9. System implementation

5.2 Simulation and Result Analysis

The office building used in this experiment has five floors and three blocks including office rooms, lecture theaters and laboratories. Each block with an exit is connected to an adjacent block by a staircase. Blind, static and sensor tracking and risk aware evacuation plans according to the explanation of Section 4 are simulated in Repast for the same twenty fire scenes to keep the results comparable. The speed of user movement is assumed to be $1.5m$ per second. The initial position of 100 evacuees are randomly given and recorded into an XML file which will be imported at the beginning of each scene simulation to ensure all evacuees escape from the same initial position.

Figure 10 shows a sensor tracking and risk aware evacuation scene with a part of the route graph. The blue cube indicates the evacuee, the red circles the blocked, and the yellow circles the risky vertices in the route graph. For readability the sensor graph is not shown. Only sensor vertices are indicated (red cube: blocked, yellow: risky, gray:

normal, dark: extinguished). The fire source is on the fourth floor, and most of the evacuees are near the staircase except E_1. The evacuation system generates evacuation routes, so most evacuees are escaping toward the staircase, and E_1 escapes to the other block as the only pathway to the staircase in this block is blocked.

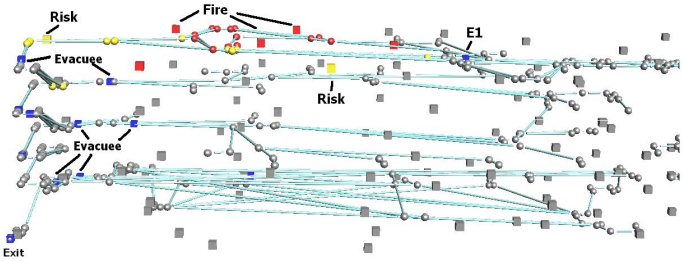


Fig. 10. Evacuation simulation

Figure 11(a) includes part of the data exported by Repast. The *success* columns of three evacuation plans are the number of successful evacuees in different scenes. From the data and the corresponding line chart in Figure 11(b), all evacuees in the twenty scenes of the sensor tracking and risk aware evacuation plan are successful, while there are some unfortunate people who fail to evacuate in static or blind evacuation plans. Therefore, the sensor tracking and risk aware evacuation plan is the safest for evacuees.

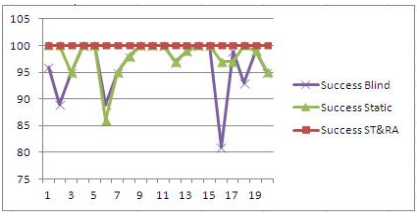
The *AveEvaTime* columns in Figure 11(a) and the corresponding chart in Figure 11(c) depict the average evacuation time in each scene of the three plans. Generally, the average evacuation time in sensor tracking and risk aware planning is the least but there are some exceptions. In Scene 16, the *AveEvaTime* of this mode is 103.5 seconds which is greater than the other two. There are two possible reasons. The first one is the average evacuation time is the quotient of the sum of evacuation time and the number of success evacuees. People who fail to evacuate in static or blind evacuation plans are not included in the calculation, but those people may need a little more time to evacuate with sensor tracking and risk aware plans which leads to the special case. Another reason is a blocked sensor vertex at T_1 may change to extinguished at T_2. The sensor tracking and risk aware planning will generate a longer but safer evacuation route at T_1. When evacuees follow this route it will take more time to escape. In blind evacuation, since no feedback from sensor and risk assessment evacuees may follow the shortest path at T_1 and at T_2 they are so lucky to pass the extinguished area which were blocked at T_1. The special cases are only 15% of the total scenes marked with gray background in Figure 11(a).

The *MaxTime* columns are the maximum evacuation time in each scene in different plans. The maximum evacuation time refers to the escape time of the last successful person. Regardless of the unsuccessful evacuees, the average maximum evacuation time of sensor tracking and risk aware plans are still the least which is 198.3 seconds.

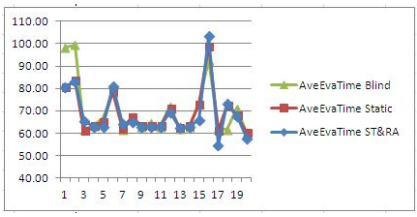
Thus, the results of the simulation demonstrate that sensor tracking and risk aware evacuation plans are safer and more reliable than the other two evacuation plans.

Emergency Scenes	ST&RA			STATIC			BLIND		
	Success	AveEvaTime	MaxTime	Success	AveEvaTime	MaxTime	Success	AveEvaTime	MaxTime
1	100	80.68	177	100	80.68	177	96	98.42	200
2	100	83.38	177	100	83.38	177	89	99.52	199
3	100	65.74	193	95	60.95	177	95	61.48	177
4	100	63.08	177	100	63.08	177	100	63.32	177
5	100	63.20	177	100	65.08	177	100	66.80	189
6	100	81.03	216	86	78.75	373	89	78.55	375
7	100	64.15	194	95	62.14	194	95	61.53	194
8	100	65.27	293	98	67.31	155	98	67.36	155
9	100	63.15	177	100	63.15	177	100	63.15	177
10	100	63.08	177	100	63.08	177	100	64.76	193
11	100	63.08	177	100	63.08	177	100	63.08	177
12	100	69.55	177	97	71.24	177	97	71.92	177
13	100	62.49	276	99	62.18	155	99	62.04	155
14	100	63.08	177	100	63.08	177	100	63.08	177
15	100	65.97	177	100	72.95	314	100	73.44	320
16	100	103.50	239	97	99	224	81	92.81	230
17	100	54.87	188	97	61.19	155	99	61.93	155
18	100	73.39	177	100	72.39	193	93	61.47	182
19	100	67.99	242	99	67.99	302	99	71.00	312
20	100	58.03	177	95	60.19	177	95	60.31	177
Average	*100*	*68.7*	*198.3*	*98*	*69.0*	*200.6*	*96*	*70.3*	*204.9*

(a) Data table



(b) Successful evacuation



(c) Average evacuation time

Fig. 11. Result analysis

5.3 Real-Time System Efficiency

Time is extraordinarily sensitive for all evacuation systems. The presented model in this study, however, is mainly based on pre-calculated networks and their integration, as described in Section 5.1, which has no effect on real-time system efficiency. Dynamic updates only apply to the presumably small areas of impact. These dynamic updates have the following complexity:

1. The cost to communicate affected sensor nodes to the route graph; this cost is linear because the integration is organized through a table.
2. The cost to renew the risk rank value for affected nodes and edges in the route graph. These costs depend on the chosen risk assessment model (the presented simple model is also linear), but for small areas of impact it will affect not many sensor nodes.
3. The cost of updating the route graphs on all mobile devices, which is a communication cost in a push or pull mode.
4. Positioning costs of the evacuees' mobile devices are constant, since each device locates itself.

5. The cost to calculate the shortest paths for each evacuee on the updated route graph happens in parallel on the mobile devices. We use an A* algorithm, which has a time complexity of $O(2^n)$ in the worst case, and $O(n \log(n))$ in the best [28, 29], n being the number of nodes in the route graph.

In brief, the eventual time efficiency of the system to a great extent relies on the size of the route graph.

6 Conclusion and Future Work

This study proposes an integrated data model for indoor evacuation on the basis of building sensor systems and building structure. With the purpose of employing real-time sensor data as references for evacuation route calculation, this paper converts monitoring sensors to sensor graphs, and integrates these sensor graphs to a route graph. With this integration, sensor tracking and risk aware evacuation routes may be generated dynamically for evacuees, which have been shown to be superior to static or blind evacuation plans.

Future work may optimize the sensor graph generation algorithm, and aim for automatic indoor positioning. But fundamentally, this model lays foundations for refined integration of modern building facilities and indoor GIS applications, for example through a more detailed risk model. Any improved risk model would only improve further the results of the presented model. Currently, only the risk of nodes and edges in route graph to be affected next by the event is considered. Other risks could be introduced as well, such as event propagation models, or congestion risk. Congestion risk, however, which is a function of network centrality and distribution of evacuees at the time of the alarm, requires either a central planning or a coordinated planning; in the current implementation route planning happens individually on the mobile devices. Standardization of interfaces between building sensor systems and evacuation systems will help adopting mobile evacuation applications on a large scale.

Acknowledgments. Funding of the European Union via mSAFE under grant agreement No. FP7-PEOPLE-2011-IRSES 295269 is gratefully acknowledged (`http://msafe.informatik.uni-bremen.de/`). Access to YAMAMOTO was kindly provided by Christoph Stahl, Bremen.

References

1. Richter, K.F., Winter, S., Rüetschi, U.J.: Constructing hierarchical representations of indoor spaces. In: Tenth International Conference in Mobile Data Management, Workshop on Indoor Spatial Awareness, pp. 686–691. IEEE Press, Taipei (2009)
2. OGC: Indoor geography markup language introduction (2013), `http://www.opengeospatial.org/projects/groups/indoorgmlswg`
3. Lee, J., Kwan, M.: A combinatorial data model for representing topological relations among 3D geographical features in micro-spatial environments. International Journal of Geographical Information Science 19(10), 1039–1056 (2005)

4. Stoffel, E.-P., Lorenz, B., Ohlbach, H.J.: Towards a semantic spatial model for pedestrian indoor navigation. In: Hainaut, J.-L., et al. (eds.) ER Workshops 2007. LNCS, vol. 4802, pp. 328–337. Springer, Heidelberg (2007)
5. Becker, T., Nagel, C., Kolbe, T.H.: A multi-layered space-event model for navigation in indoor spaces. In: Lee, J., Zlatanova, S. (eds.) 3D Geo-Information Sciences. Lecture Notes in Geoinformation and Cartography, pp. 61–77. Springer, Berlin (2008)
6. Wu, C., Chen, L.: 3D spatial information for fire-fighting search and rescue route analysis within buildings. Fire Safety Journal 48, 21–29 (2012)
7. Liu, J., Lyons, K., Subramanian, K., Ribarsky, W.: Semi-automated processing and routing within indoor structures for emergency response applications. In: Buford, J.F., Tolone, W.J., Jakobson, G., Ribarsky, W., Erickson, J. (eds.) Cyber Security, Situation Management, and Impact Assessment II; and Visual Analytics for Homeland Defense and Security II, vol. 7709, p. 77090Z. The International Society for Optical Engineering (2010)
8. Kraus, L., Stanojevic, M., Tomasevic, N., Mijovic, V.: A decision support system for building evacuation based on the EMILI SITE environment. In: 20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), p. 334 (2011)
9. Han, Z., Weng, W., Zhao, Q., Ma, X., Liu, Q., Huang, Q.: Investigation on an integrated evacuation route planning method based on real-time data acquisition for high-rise building fire. IEEE Transactions on Intelligent Transportation Systems 14(2), 782–795 (2013)
10. Jiyeong, L.: A three-dimensional navigable data model to support emergency response in microspatial built-environments. Annals of the Association of American Geographers 97(3), 512–529 (2007)
11. Zhang, L., Wang, Y., Shi, H., Zhang, L.: Modeling and analyzing 3D complex building interiors for effective evacuation simulations. Fire Safety Journal 53, 1–12 (2012)
12. Tang, F., Ren, A.: GIS-based 3D evacuation simulation for indoor fire. Building and Environment 49, 193–202 (2011)
13. Lei, Z., Gaofeng, W.: Design and implementation of automatic fire alarm system based on wireless sensor networks. In: Proceedings of the International Symposium on Information Processing, pp. 410–413 (2009)
14. Nauman, Z., Iqbal, S., Khan, M.I., Tahir, M.: WSN-based fire detection and escape system with multi-modal feedback. In: Dziech, A., Czyżewski, A. (eds.) MCSS 2011. CCIS, vol. 149, pp. 251–260. Springer, Heidelberg (2011)
15. Sha, K., Shi, W., Watkins, O.: Using wireless sensor networks for fire rescue applications: Requirements and challenges. In: IEEE International Conference on Electro/Information Technology, pp. 239–244. IEEE (2006)
16. Al-Karaki, J.N., Kamal, A.E.: Routing techniques in wireless sensor networks: A survey. IEEE Wireless Communications 11(6), 6–28 (2004)
17. Nguyen, M., Ho, T., Zucker, J.: Integration of smoke effect and blind evacuation strategy (SEBES) within fire evacuation simulation. Simulation Modelling Practice and Theory 36, 44–59 (2013)
18. Stahl, C., Schwartz, T.: Modeling and simulating assistive environments in 3D with the YAMAMOTO toolkit. In: International Conference on Indoor Positioning & Indoor Navigation (IPIN), p. 1 (2010)
19. Munzer, S., Stahl, C.: Learning routes from visualizations for indoor wayfinding: Presentation modes and individual differences. Spatial Cognition and Computation 11(4), 281–312 (2011)
20. Stahl, C.: New perspectives on built environment models for pedestrian navigation. In: Spatial Cognition 2008 Poster Proceedings. Universität Freiburg (September 2008)
21. Stahl, C.: Spatial modeling of activity and user assistance in instrumented environments. PhD thesis, DFKI, University of Saarland (2010)

22. Chandra, A.M.: Dividing a circular arc into equal numbers of division. IOSR Journal of Mathematics 4, 38–39 (2012)
23. Stanford University: Geometric algorithms design and analysis (1992), http://graphics.stanford.edu/courses/cs268-09-winter/notes/handout7.pdf
24. National Fire Protection Association: National Fire Alarm Code. 2007 edn. An International Codes and Standards Organization (2007)
25. Aziz, N.A.A., Aziz, K.A., Ismail, W.Z.W.: Coverage strategies for wireless sensor networks. In: Proceedings of the World Academy of Science, Engineering and Technology, vol. 50, pp. 145–150 (2009)
26. American Institute of Chemical Engineers, ed.: DOW's Fire & Explosion Index Hazard Classification Guide. 7th edn. Wiley (1994)
27. National Fire Protection Association: Guide on Alternative Approaches to Life Safety. 2013 edn. An International Codes and Standards Organization (2013)
28. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics 4(2), 100–107 (1968)
29. Hart, P.E., Nilsson, N.J., Raphael, B.: Correction to a formal basis for the heuristic determination of minimum cost paths. ACM SIGART Bulletin (37), 28–29 (1972)