International Journal of
Geo-Information

MDPI

Article

An Efficient Shortest Path Routing Algorithm for Directed Indoor Environments

Sultan Alamri

College of Computing and Informatics, Saudi Electronic University, Riyadh 32256, Saudi Arabia; salamri@seu.edu.sa; Tel.: +966-593-530-022

Received: 24 January 2018; Accepted: 22 March 2018; Published: 26 March 2018

Abstract: Routing systems for outdoor space have become the focus of many research works. Such routing systems are based on spatial road networks where moving objects (such as cars) are affected by the directed roads and the movement of traffic, which may include traffic jams. Indoor routing, on the other hand, must take into account the features of indoor space such as walls and rooms. In this paper, we take indoor routing in a new direction whereby we consider the features that a building has in common with outdoor spaces. Inside some buildings, there may be directed floors where moving objects must move in a certain direction through directed corridors in order to reach a certain location. For example, on train platforms or in museums, movement in the corridors may be directed. In these directed floor spaces, a routing system enabling a visitor to take the shortest path to a certain location is essential. Therefore, this work proposes a new approach for buildings with directed indoor spaces, where each room can be affected by the density of the moving objects. The proposed system obtains the shortest path between objects or rooms taking into consideration the directed indoor space and the capacity of the objects to move within each room/cell.

Keywords: spatial databases; directed indoor space; routing algorithm

1. Introduction

With the advanced enhancements of the current positioning devices such as RFID, Bluetooth and WiFi, the concentration on the data structures of indoor spaces and the querying for spatial objects have become important foundations for many applications [1–3]. In indoor spaces, the efficiency of the spatial database can be affected by the structure of the building [4,5]. Therefore, every feature of each indoor space should be considered in order to achieve an efficient data structure for certain queries pertaining to the indoors [6–8]. Moreover, the data structure has to take into account all the issues that affect the routing of objects and the query processing associated with indoor spaces.

Furthermore, the significance of the data structure of the indoor space can be different depending on the shape and features of the indoor space. There are many forms of indoor spaces, such as an open indoor, multi-floor environments and directed indoor spaces [1,4]. Most of the current research on indoor spaces has not considered the directed indoor spaces [9,10], although, in many buildings, the movement of the objects is directed. For example, the movement in some railway station platforms is directed to facilitate the traffic especially during rush hour. Moreover, in some museums or stores (i.e., IKEA), the movement is directed whereby the object is required to use a certain directed corridor to reach a certain area. Therefore, building a routing system for directed indoor spaces in such buildings is important and facilitates many applications.

In this paper, we propose a new routing structure for objects moving in directed indoor spaces. This work exploits the notion of cellular space, which takes into account the indoor walls and partitions that control movement in an indoor environment. Moreover, we focus on directed indoor spaces where the moving objects are required to move in a certain direction through rooms/corridors as a means of controlling indoor traffic. In addition, data density plays an essential role in ensuring the accuracy of the routing structure. Therefore, the dense cells in an indoor environment in data structures must be taken into consideration as this can increase system accuracy [5,6,11]. This is greatly invested in outdoor spaces' traffic, where any traffic jam will be considered in the routing time. Thus, this work also considers the impact of the density of the objects inside the cell, which affects the objects' movement towards a certain cell [1,7,12]. Table 1 explains the notations used throughout this paper.

The rest of the paper is organized as follows. Section 2 explains the background, which includes the differences between the spatial road network and indoor cellular, directed indoor space and indoor positioning technologies. Section 3 reviews several moving object structures and routing in indoor spaces. Section 4 explains our proposed structure including the indoor space as a connected directed graph, directed indoor adjacency matrix, directed indoor adjacency tree, directed indoor capacity and the routing algorithm. Section 5 presents the results of a performance evaluation of the proposed data structures. Finally, we conclude in Section 6 with possible directions for future work.

Table 1. Notations used throughout this paper.

Notation	Definition
C	Cell
S	Space
G	Digraph
$O_i \overset{in}{\rightarrow} C_i$	O_i located inside cell C_i
$C_i \overset{to}{\rightarrow} C_j$	The direction of movement from C_i to C_j
DIA	Directed indoor adjacency
C_i<>C_j	C_i connected to C_j
C_i><C_i	C_i not connected to C_j

2. Background

2.1. Spatial Road Network and Indoor Cellular

In this section, we explain the difference between a spatial road network and a cellular space. A spatial road network considers the distance between any two spatial objects O_{11} and O_{22} as a network distance (or shortest road) [1,13,14]. Figure 1 shows an example of a spatial road network. However, the cellular space is a representation of locations using sets of cells that include the spatial objects. A cellular space presents location information in less refined form as opposed to that produced by a spatial road network. Several works on indoor routing have considered indoor space in terms of metric distance similar to a spatial road network using distance-aware models [15]. Moreover, other works have taken the same approach when dealing with large, critical indoor infrastructures such as warehouses, airport terminals or railway stations [16,17]. However, we argue that indoor space should be treated as cellular space, where the cells are considered as the containers of spatial objects. The distance between the spatial objects in the cellular space is the number of hops between them [1]. Basically, in a spatial road network, a query or a spatial object is determined using coordinates such as $(x_i; y_i)$ and $(x_j; y_j)$. On the other hand, a query or a spatial object in cellular space is mainly based on cellular notations such as "What objects are in Cell or Room 11?" In this example, the cell number is the identifier of the objects' locations. Figure 2 shows an example of an indoor space.

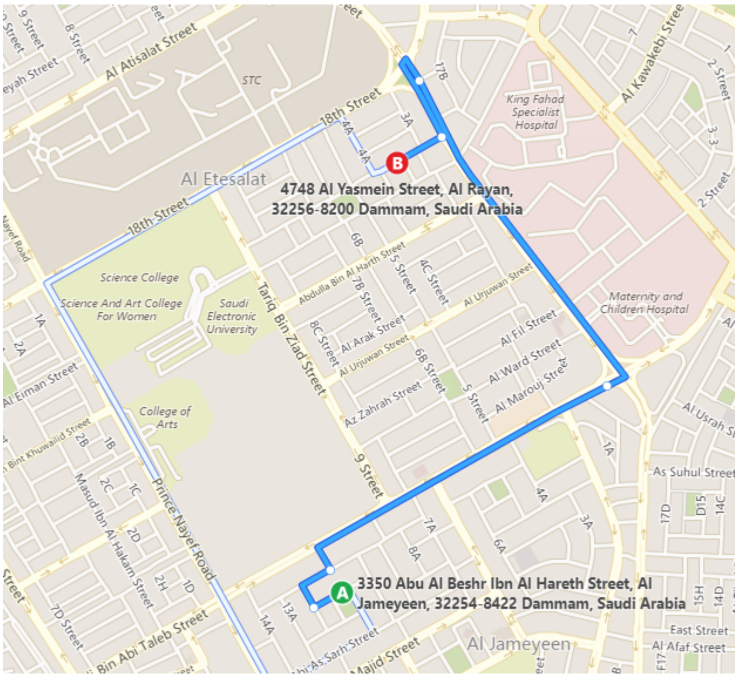


Figure 1. The shortest path between *A* and *B* in a spatial road network.

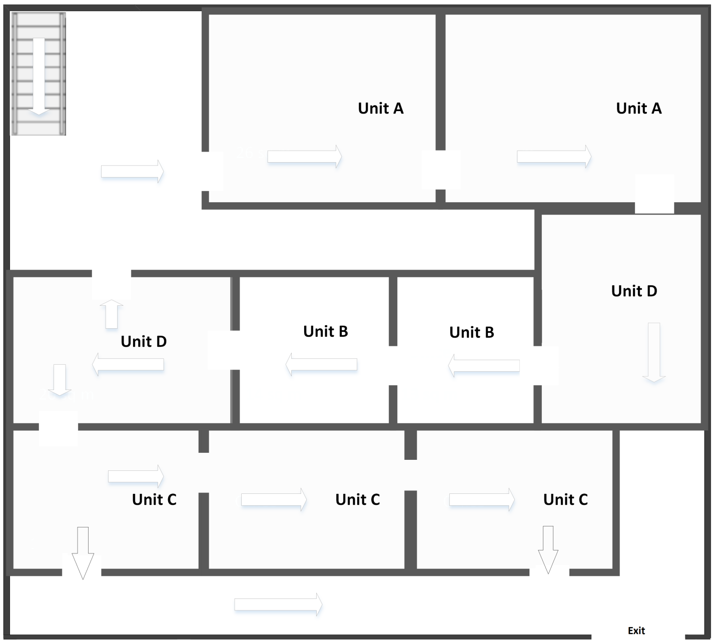


Figure 2. An example of indoor cells.

2.2. Directed Indoor Space

In some buildings, the movement of objects in the indoor space is directed just as it is in some outdoor environments. Some indoor spaces restrict the movement of the object; for example, the objects most use a certain corridor and direction in order to reach a certain room. This is very obvious in some indoor environments such as museums where visitors travel in a specific way and direction until they reach the exit point. Other examples can be seen in some airports or shopping centers, where the movement of objects is directed.

Furthermore, the directed indoor space has played an important role in indoor traffic in recent years as it facilitates the evacuation of buildings and eases people traffic in buildings (such as metro platforms), among many other advantages. Therefore, we argue the importance of building a routing approach for directed indoor spaces. In this work, we use the cell-based concept of indoor space, where each room can be considered as a cell. Moreover, corridors or stairs can be divided into several cells as needed. In this work, we assume that there is no significant difference in size between the cells and that the velocity of the moving objects is fixed. The impact of the significant difference in size between cells and velocities will be considered in future work.

In addition, for the positioning of moving objects in the directed indoor space, this work will be based on indoor WiFi positioning triangulation. The basic idea is to use multiple access points (nodes/cells) fixed in the indoor spaces [18,19]. The nodes will be installed within a short distance from each other. These "referenced nodes" will be located in the cells on different floors; thereby making it appropriate for a multi-floor building. Furthermore, indoor positioning techniques based on the pattern of observationsinclude multiple WiFi hotspots (nodes) and fingerprinting. Here, the observations are compared to the mapped positions and based on "trilateration", which is used to indicate the distance from the transmitter to locate the device using a geometric calculation [20,21]. Here, the WiFi-based positioning technologies take advantage of the rapid developments of wireless nodes (access points) in indoor spaces and urban areas. Figure 3 illustrates a WiFi-based positioning system in an indoor space.

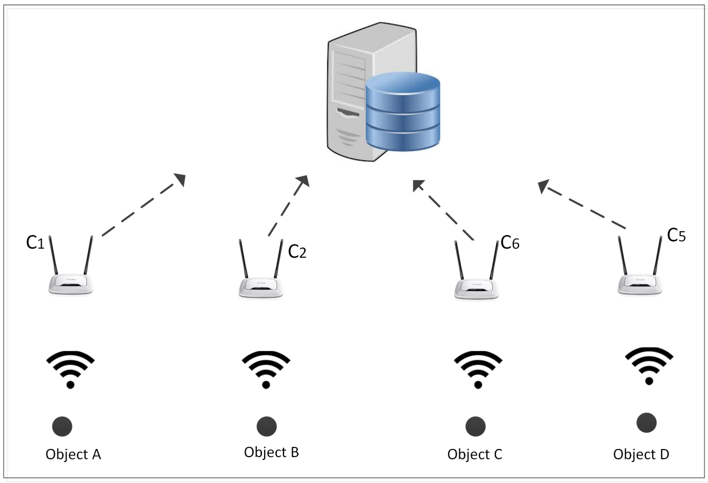


Figure 3. Indoor WiFi-based positioning system.

3. Related Works

This section discusses some of the current research in the querying and indexing of moving objects in indoor spaces.

In the RTRand TP2Rtrees, the RFID reader is used in order to collect the data in the indoor environment [5]. The RTR-tree uses the basic node organization in the R-tree. In the indoor structure, the trajectory basically is a set of line segments. For instance, in any range query, the Euclidean range space will be the RFID-set readers. Then, the search is performed just as is usually done in the R-tree. However, there are differences between the TP2R-tree and the RTR-tree. In the TP2R-tree, the index entities in the leaf nodes have the form (MBR, Ct, RID), where the minimum boundary of a record is MBR, which is determined via RID, and Ct shows the period of the reading by a reader (or the time parameter). Note that Ct has time end or time start. Furthermore, the TP2R-tree has non-leaf nodes that contain the index entities in the form (MBR, Ct, Cp), where the higher MBR contains all lower MBRs in the child nodes, where Cp is the child pointer and Ct is a parameter for time.

Xie et al. [14] proposed an upper/lower bound topology for indoor distance. These topological upper/lower bounds are used in two types of query processing: range and Knn. Furthermore, it concentrated on the distance in indoor environments between the object O_2 and the query Q, the positions of which are indicated by the positioning system. Therefore, the basic idea here is to divide the inaccurate moving object positions inside disjointed sub-regions. Here, each sub-region is dropped into one indoor partition. Then, based on the topological characteristics, the distances between the sub-regions and the query will be categorized. Note that this takes indoor distances into account. Figure 4 illustrates the composite index for indoor environments, which basically consists of three layers: the geometric layer, the topological layer and the object layer.

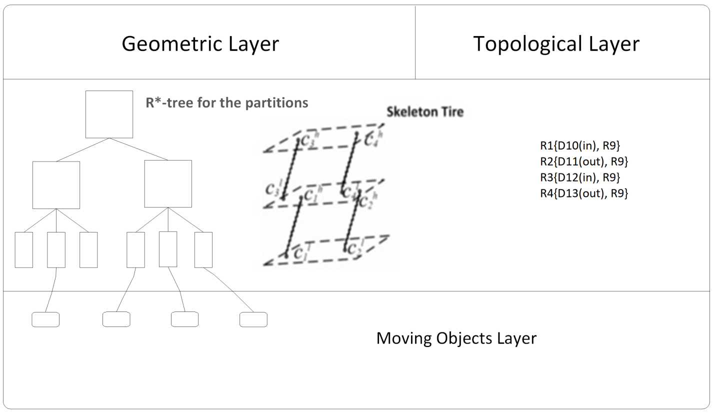


Figure 4. The composite index includes three layers.

In [3], the adjacency-based indexing method was proposed, which is based on the connectivity graph (see Figure 5). In this research, the temporal aspects of moving objects in indoor spaces were introduced. For this aim, the suggested data structure is the Moving Objects Timestamping-tree (MOT-tree). The idea is to track only the updated moving objects (i.e., those that have checked out or checked in to a new cell/room). The MOT-tree is based on the cell-based indexing method, whereby objects moving at the current time are linked with their predecessors (see Figure 5). In addition, [3] proposes another indexing structure, the indoor Trajectories Deltas Index (ITD-tree), the aim of which is to reduce the update cost by using the deltas techniques as a temporary tracker of the object's modifications in the floor space.

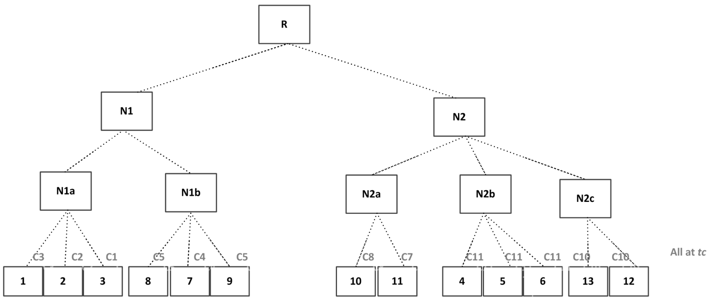


Figure 5. An example of the Moving Objects Timestamping-tree (MOT-tree).

In [22], this data structure basically combines geospatial data such as route graphs and sensor graphs. Furthermore, it includes environmental data such as sensor readings to obtain real-time risk-aware evacuation planning in emergency cases. This work is concerned with indoor evacuation. Therefore, the data model has been designed for that purpose. The proposed data model, implemented in a personalized evacuation system, will allow the creation of a dynamic evacuation route for the targeted people. This system depends on the reading of the current sensor and takes risk assessments into account. The likely benefits of the model can also extend to a system that links a route network with a sensor network for a variety of indoor space applications.

Furthermore, there are significant differences that make indoor routing more difficult than outdoor routing. Outdoor routing is generally implemented in two-dimensional spaces, while indoor routing allows the routing of three-dimensional spaces such as those in multi-story buildings. In the case of indoor routing, the method of using three-dimensional spaces will identify an object accurately by storing spatial data that are represented in undirected graph form. This has the features of three-dimensional data where x and y are the coordinates of a point and z represents the height level of the point. In order to determine the shortest route between two points, a shortest path algorithm can be implemented that is based on the three-dimensional space structure. The work of [23] implemented an indoor routing system using the method in three-dimensional spaces in order to determine the shortest route in general and the closest shortest route, which is the shortest route that will not pass through the open space (to avoid some environmental conditions such as rain or sun). In [23], the indoor space is undirected, meaning that there is free movement, unlike the directed indoor space considered in this work. Moreover, the aim of [23] was to test different algorithms such as the Floyd–Warshall algorithm, the Bellman–Ford algorithm, the A* algorithm and Dijkstra's algorithm in order to find the shortest path in the buildings of Telkom University. Note that in [23], the algorithm used a metric distance, which is not applied in our proposed system.

As is clear, most of the previous research has focused on regular indoor spaces where movement is not restricted or directed [23–25]. Moreover, most of the current indoor space data structures and querying do not consider the density of objects moving in indoor space, which actually has an important impact on movement indoors [26–28]. Moreover, any spatial queries such as routing or KNN queries in indoor spaces can easily be affected by the cell density. Therefore, this work has concentrated on the directed indoor spaces, in addition to considering the density of moving objects inside the rooms/cells of the indoor environments.

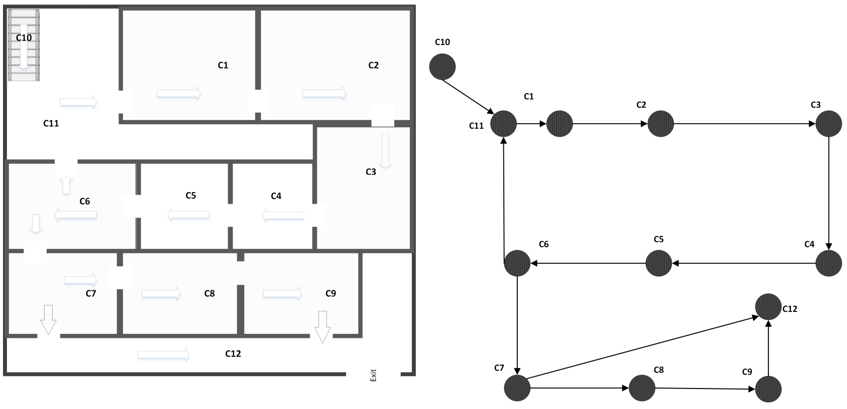
The majority of works in indoor space routing or querying have been done for the regular indoor space where objects can move freely. However, as explained previously, there are certain buildings where the movement is directed. Many airports, libraries and large mosques actually have traffic for the movement of objects inside [1]. Moreover, many shopping centers and department stores such as IKEA have a direction corridor for the customer. Therefore, this work concentrates on building a routing approach for these kinds of indoor spaces. Moreover, the density of the moving objects in indoor space has not been considered in most of the research on routing or querying in indoor spaces [1,8,27]. Thus, this work also considers the density of the moving objects inside the cells/rooms in the directed indoor space, which has an important impact on the movement of the objects.

4. Proposed Structure

The construction of a data structure involves the following: representing the indoor space as a connected directed graph and directed indoor adjacency matrix; converting the connectivity graph into a tree (directed indoor adjacency tree); the cells' capacity (density) mechanisms; and the routing algorithm in directed indoor spaces.

4.1. Indoor Space as a Connected Directed Graph

In this section, we convert the indoor space into a logical model. The logical model consists of a graph that contains both directed edges (connections) and undirected cells. First, the indoor spaces are divided. An indoor space can contain things such as rooms, doors, corridors, floors, stairs, elevators and pathways between buildings. Therefore, the mappings from domain concepts to modeling concepts can be summarized as follows: A room is mapped to a cell and a door to an edge undirected connection of two cells. The directed corridor, or any other kind of pathway (circulation area), is a maximally-long linear structure of directed connected cells based on the structure of the indoor space. For example, Figure 6a shows the 2D indoor structure of a museum and the direction in which moving objects are required to move. Note that if a corridor is not too long, we can model it as one cell. Otherwise, it is possible to divide it into several cells for the logical model. This is done manually, based on the judgment of the modeler. Note that the stairs are treated similarly; therefore, they can be mapped to a single cell or multiple cells as needed. Moreover, the elevator is mapped to a single cell. Note that floors and buildings can vary.



(**a**) (**b**)

Figure 6. Directed graph for the building. (**a**) An example of a directed floor; (**b**) directed indoor graph.

Definition 1. *Given a set of cells (or nodes) C =* $\{C_1, C_2, .., C_n\}$ *and spatial objects* $\{O_1, O_2, ..., O_n\}$*, the space S is called the indoor cellular space if: S =* $\bigcup C_i$*, where* $O_i \overset{in}{\rightarrow} C_i$*.*

Definition 2. *An indoor space is a connected directed indoor C =* $\{C_1, C_2, .., C_n\}$ *where a set of cells (rooms) is considered as a set of edges/corridors (*C_{e1}*,* C_{e2}*,* C_{e4} *..*C_{en}*), where the objects' movement is restricted as* $C_{e1} \overset{to}{\rightarrow} C_{e2}$*,* $C_{e2} \overset{to}{\rightarrow} C_{e4} .... \overset{to}{\rightarrow} C_{en}$*.*

An example of indoor space depicted as a connected directed graph is shown in Figure 6.

4.2. Directed Indoor Adjacency Matrix

In this section, we introduce an adjacency matrix for the directed indoor spaces. The basic idea here is to illustrate how the directed cells are connected to each other. The directed indoor adjacency matrix is important in order to extract the directed indoor graph. The directed indoor adjacency matrix of an indoor graph with n cells, m lines and no self-loops is an nxm matrix $DIA = [ij]$, where rows correspond to lines (connects) and columns correspond to cells (nodes).

Definition 3. *Let G be a directed graph with n cells and m line. The directed indoor adjacency matrix* $DIA = [ij]$ *of the digraph G is an nxm (0, 1) matrix defined by:*

$$DIA = \begin{cases} 1 & \mathit{if}\ i <> j. \\ 0 & \mathit{if}\ i >< j. \end{cases}$$

For example, consider the directed indoor graph depicted by Figure 7. C_{10} is connected with C_1 through a, and C_3 is connected through b to C_2. Note that C_{10}, C_1, C_2 are all considered as cells, although they can be rooms or corridors. The idea here is to use the directed indoor adjacency matrix for the routing algorithm in order to use all the possible tracks and directions in indoor space.

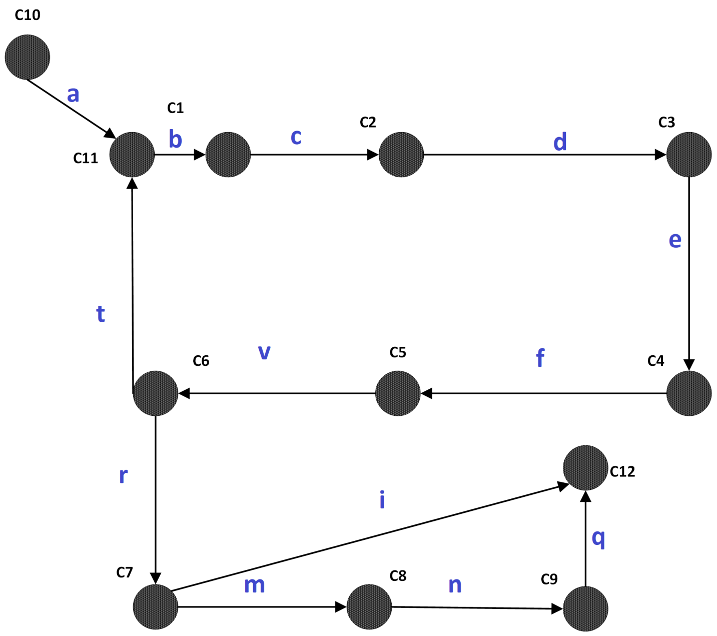


Figure 7. Example of an indoor directed graph.

The directed indoor adjacency matrix of G is as follows:

$$\left(\begin{array}{ccccccccccccc}
 & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} \\
a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
b & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
c & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
e & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
f & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
r & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
m & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
n & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
q & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right)$$

4.3. Directed Indoor Adjacency Tree

In this section, we convert the directed graph to a directed indoor adjacency tree. However, before that, we need to indicate the cells in the graph. Figure 6a shows the previous distribution of indoor space and its directed graph. As shown, both the stairs and the corridors are considered as cells.

In the directed indoor adjacency tree, first we select one cell as the main cell. This cell is usually the entrance or the stairs that direct objects to a certain floor (for example, cell C_{10} in Figure 6a). Next, we perform a directed expansion where each cell will be connected with the next cell to which it is directed. For example, C_{11} is directed to C_1, and C_1 is directed to C_2, and so on. Note that, in some cases, the directed expansion can take into consideration more than one cell such as C_6 directed to both cells C_7 and C_{12}. Each cell will have a list of the moving objects inside it. Figure 8 shows a directed indoor adjacency tree for the floor in Figure 6a.

Definition 4. *Given a set of cells* $C = \{C_1, C_2, .., C_n\}$, C_j *is considered as a directed cell iff* $C_j <> C_n$, *and the direction is restricted as* $C_j \overset{to}{\rightarrow} C_n$.

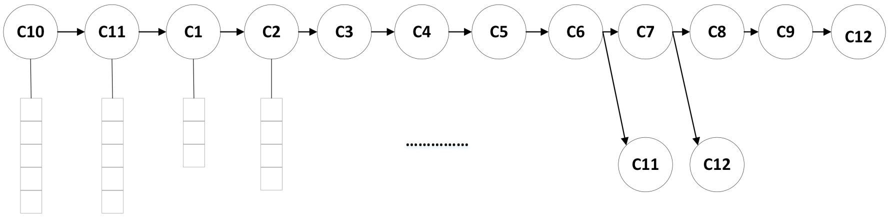


Figure 8. Directed indoor adjacency tree.

4.4. Directed Indoor Capacity

In this section, we propose a directed indoor capacity (DIC) for the directed indoor spaces. The basic idea here is to illustrate the impact of the cell capacity or the density of the objects inside any cells (rooms or corridor). For the directed indoor capacity of a digraph with n cells, at time t, the cell capacity can be defined as follows:

Definition 5. *Given a set of cells* $C = \{C_1, C_2, .., C_n\}$, *and a set of objects* $O = \{O_1, O_2, ..., O_n\}$, *the capacity of* C_i *is the range of lowest and highest density of objects in* C_i.

Definition 6. *Let G be a directed graph with the set of cells* $C = \{C_1, C_2, .., C_n\}$. *The directed indoor capacity values at time (t) of the digraph G are (1, 0, −1, −2), defined by:*

$$DIC = \begin{cases} 1 & \textit{if } C_i \textit{ is in low density.} \\ 0 & \textit{if } C_i \textit{ is in regular density} \\ -1 & \textit{if } C_i \textit{ is in high density.} \\ -2 & \textit{if } C_i \textit{ is in extreme density.} \end{cases}$$

Note that the capacity values are determined by the admin, taking into account the size of the indoor space, the cells and the timing. For example, in Figure 7, the directed indoor capacity is defined as follows: If the room/corridor has a density of 1–5 persons, it will be considered as a low density cell. If the room/corridor has a density of 6–10 persons, it will be considered as a regular density cell. If the room/corridor has a density of 11–20 persons, it will be considered as a high density cell. If the room/corridor has a density of 21 or more, it will be considered as an extreme density cell. Therefore, the high and extreme density cells will slow the movement (by the negative values) of the moving objects in the indoor spaces.

In addition, this approach considers routing in a multi-floor building. Therefore, stairs can be one or more nodes/cells with one or more edges. The movement of objects is usually slower on the stairs than in corridors; therefore, the capacity of directed stairs can be defined as follows:

Definition 7. *Let G be a directed graph with the set of stair cells* $C = \{C_1, C_2, .., C_n\}$. *The directed indoor capacity values at time (t) of the digraph G is (0, −1, −2, −3), defined by:*

$$DIC = \begin{cases} 0 & \textit{if } C_j \textit{ is in low density.} \\ -1 & \textit{if } C_j \textit{ is in regular density} \\ -2 & \textit{if } C_j \textit{ is in high density.} \\ -3 & \textit{if } C_j \textit{ is in extreme density.} \end{cases}$$

Note that different means of transmission such as lifts or escalators will not be considered in this paper, but will be future work. The value of DIC is used by the routing algorithm to determine the flow of cells that can used by moving objects to reach their final cell destination. Next, we explain the routing algorithm in directed indoor spaces.

4.5. Routing in Directed Indoor Spaces

In this section, we discuss routing in directed indoor spaces. Algorithm 1 illustrates the steps involved in routing for directed indoor spaces in order to obtain the fastest indoor route by taking into consideration the densities of cells. The routing is shown between two cells: one as the start and one as the destination. Furthermore, it can obtain the fastest route to an object in a certain cell, where the object is actually registered to a cell. Then, a modified breadth-first search algorithm is used to obtain the shortest route between the start cell and the destination cell in the directed indoor space. The modified breadth-first search algorithm basically picks a source cell C to start. Then, taking the direction into account, it finds the cells that are adjacent to C. Then, each next C in turn and finds i adjacent cells until the final destination cell. Note that the Modified Breadth-First Search (M-BFS) traverses the directed graph to find all paths from a starting cell (or node). Here, we modify the algorithm so that $\Pi(n)$ does not store only one predecessor, but will list all possible predecessors. Note that all possible predecessors will be stored in a candidate's path table. Then, we apply the DIC value to all cells in a candidate's path as explained for directed indoor capacity. Note that, in this work, we assume that the velocity is fixed for the moving objects; in future work, we will consider the various details pertaining to the velocity of moving objects in indoor spaces. The shortest route in directed indoor spaces is the lowest value that is obtained from:

$$\text{The shortest route in the directed indoor space} = NC - DIC$$

where NC indicate the number of cells in the route (number of hops) and DIC indicate the value of the directed indoor capacity. For example, in Figure 9, the possible routes between O_1 and O_3 are four routes as follows: C_3-C_2-C_1-C_6-C_7-C_8-C_{10}-C_5-C_{16}-C_{13}-C_{18}, and C_3-C_2-C_1-C_6-C_9-C_{14}-C_{15}-C_{17}-C_{18}, C_3-C_2-C_1-C_6-C_7-C_8-C_{10}-C_5-C_{16}-C_{13}-C_{18}-C_{12}-C_{11}-C_9-C_{14}-C_{15}-C_{17}-C_{18} and C_3-C_2-C_1-C_6-C_7-C_8-C_{10}-C_5-C_{16}-C_{13}-C_{12}-C_8-C_{10}-C_5-C_{16}-C_{13}-C_{18}. All cells are considered as regular density where DIC is zero. However, C_{14}, C_{15} and C_{17} are considered as high density where DIC is -1. Therefore, based on our formula, the shortest path is (C_3-C_2-C_1-C_6-C_7-C_8-C_{10}-C_5-C_{16}-C_{13}-C_{18}), since it has the lowest value.

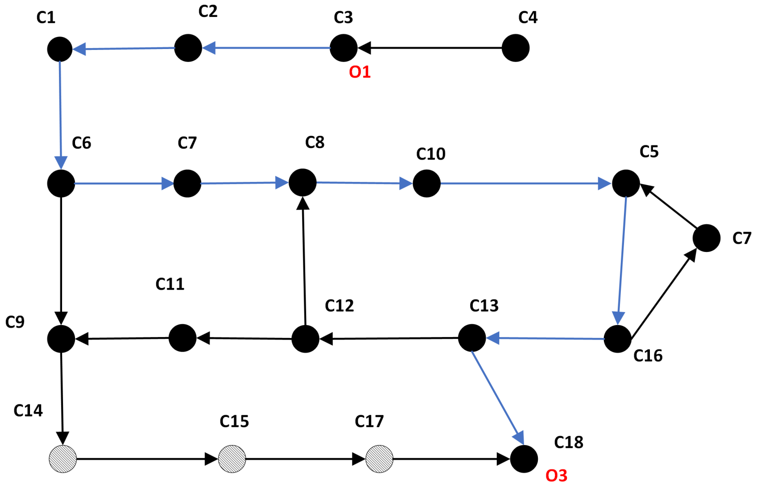


Figure 9. The shortest path between O_1 and O_3 considering the cells' densities.

Algorithm 1 *Directed_Indoor_Routing_Algorithm.*

```
/* Routing in directed indoor algorithm.
Insert start destination cell
Insert object ID
if $O_i$ Is Inserted then
    Check directed indoor adjacency tree
    Return $C_i$ where $O_i$ is located
end if
Use $M - BFS$ for possible routes
Use directed indoor capacity
Return $C_i$ where $O_i$ is located
Check each cell $DIC$ value
Assign $C_i$ value based on the cell density
Return $S$ = $NC$ - $DIC$
The shortest route is $S$
```

5. Experimental Results and Performance Analysis

In this section, we present our experimental results to evaluate the proposed routing approach. The experiment was carried out on an Intel Core i7-4765T processor 2.00-GHz PC, with 16 GB of RAM running on 64-bit Windows 10 Enterprise. Furthermore, we used NetBeans IDE 8.1 with the standard Java platform JDK 8.

In this experiment, we used real indoor environments: the School of Computing Buildings D, E and F of Telkom University, Indonesia. We modified certain cells in the building in order to accommodate the directed indoor space. We compare our proposed system with (TELUR)[23]. Note that we generate the cell densities synthetically based on realistic scenarios where some cells will be crowded at certain times, such as during lunch break. Moreover, the cells that have been identified in our approach in Buildings D, E and F of Telkom University have no significant difference in size. Note that the operations are performed seven times, and the average is calculated. The parameters used are summarized in Table 2.

Table 2. Parameters and their settings.

Parameter	Setting
Number of moving objects in building	(10–500)
Max cell density	40
Indoor space	36, 27, 26 cells
Directed stairs	81 cells
Operations	construct, route, false hits
Environment	Buildings D, E and F, Telkom University

Figure 10 shows the floor plan of the building and the simulated positioning points of moving objects. Buildings D, E and F of Telkom University, Indonesia, were selected as the real data experimental environment. The experiment involved routing between the cells inside these buildings. Although these buildings are actually connected with each other through corridors and thus can be considered as one building, we test the routing in different scenarios: as the same building and as different buildings to determine the effect of the number of hops between rooms. Note that the buildings have three floors and extend into two wings. To support related queries, the buildings are assumed to have directed indoor spaces, with few changes in the corridors, as we assigned some corridors to have a certain direction for evaluation purposes. The level of each store and brand was manually established. Figure 11 shows the plan of the real environment and the extracted routes between object O_{112} and object O_{201}.

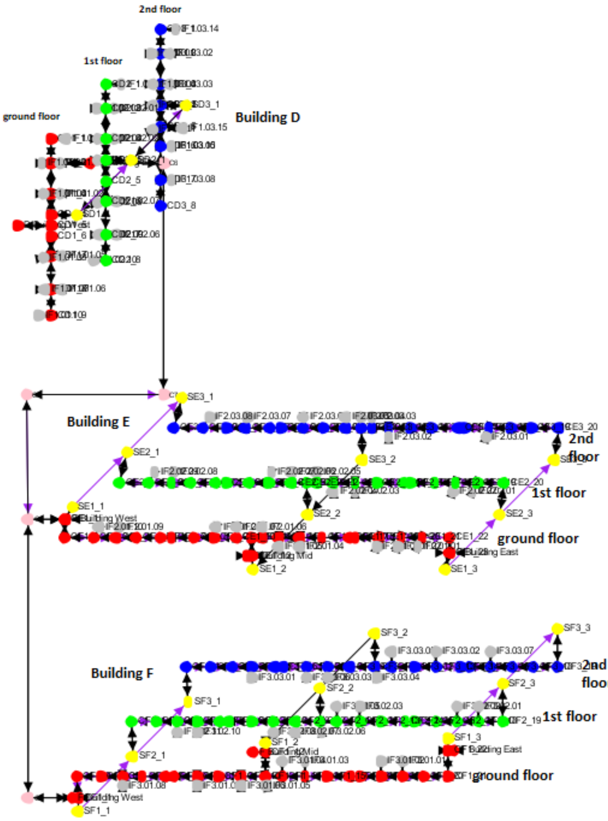


Figure 10. Buildings D, E and F.

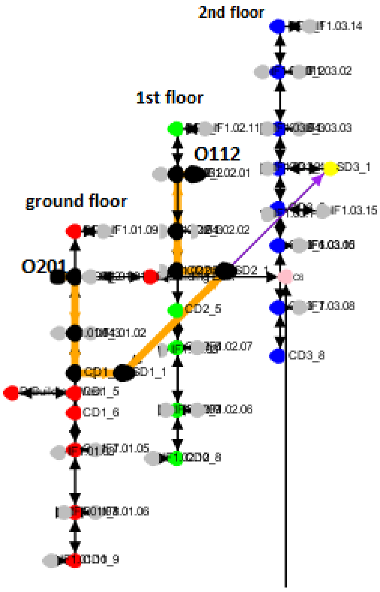


Figure 11. The shortest path between O_{112} and O_{201} considering the cells densities.

First, we investigate the following: the false hits of routing between two cells in order to obtain the shortest path. For the former, the execution time is measured for each test and false hits. Figure 12a shows the number of false hits that result from performing 18 routing cases in each density environment. As is evident, [23] has many false hits due to the lack of supporting the cell densities. Therefore, the retrieved shortest path is not actually correct in some cases, which are considered as false hits. Figure 12b shows the false hits of routing between two cells in the same building and different floors.

Figure 13a shows the false hits of routing between two cells in a different building and on the same floor. Figure 13b shows false hits of routing between two cells in a different building and on a different floor. As result, it is clear that the proposed system efficiently reduces the false hits in routing the indoor spaces because the cells' densities have been taken into account.

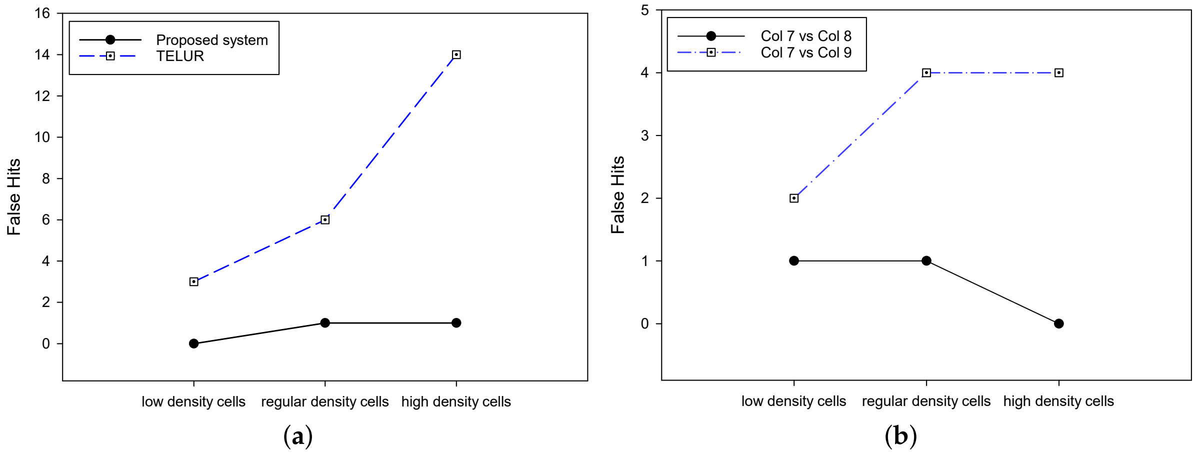


Figure 12. False hits for the path between two cells in the same building. (**a**) Same building and same floor; (**b**) same building and different floor.

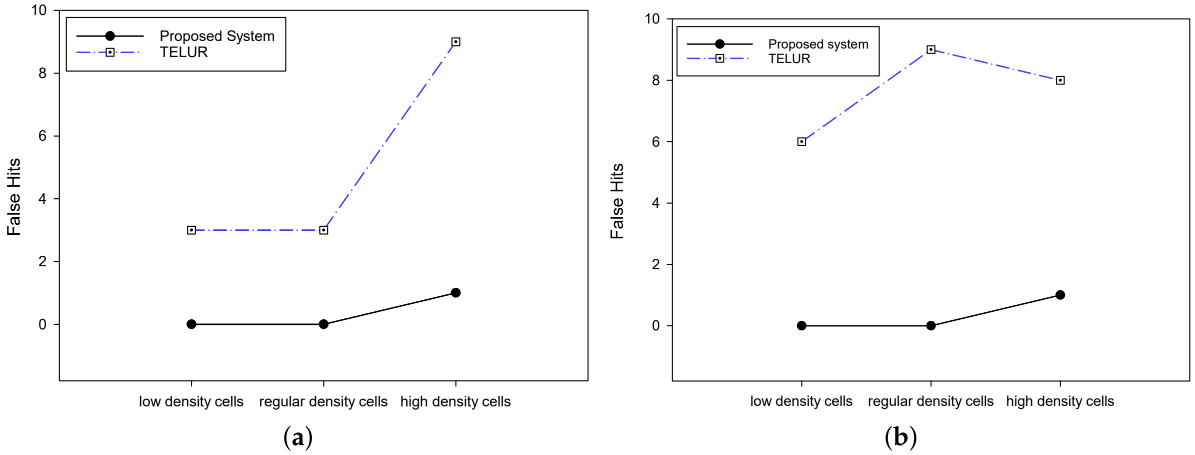
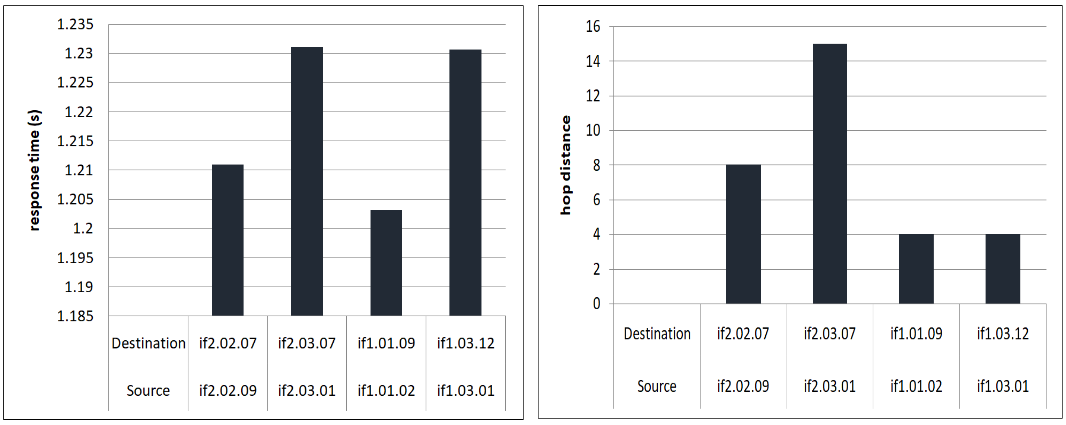


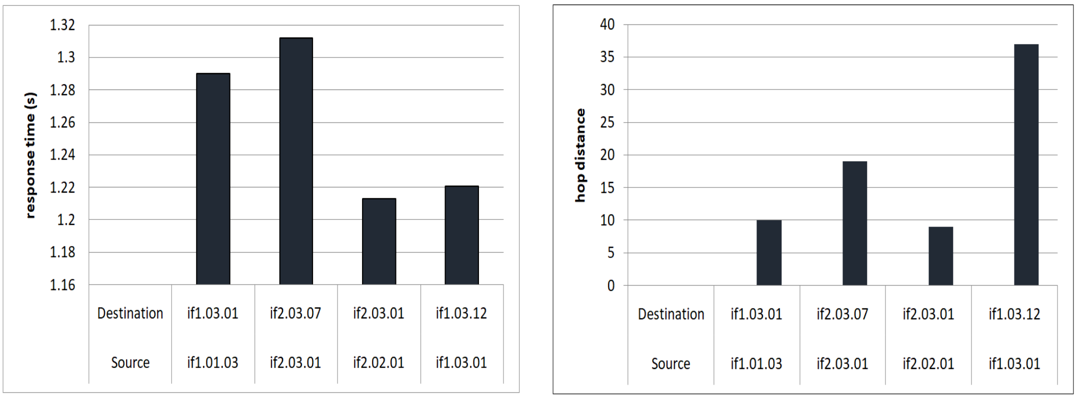
Figure 13. False hits for the path between two cells in different buildings. (**a**) Different building and same floor; (**b**) different building and different floor.

Moreover, for the proposed algorithm, we measure the costs of the routing, taking into account the number of hops between the cells (in the same or different buildings and floors). Figure 14 shows the comparison of four routing cases in the same building and similar floor. Although the number of hops increases, the response time is not affected. Figure 15 shows the comparison of four routing cases in the same building and different floor. Figure 16 shows the comparison of four routing cases in the different building and different floor. Note that in this section, the cell densities are regular and have not been modified. Therefore, some cells might have high densities of moving objects, and some might have low densities. That will depend on the scenarios where the densities are applied. Here, we see that in some cases, even with the increase in the number of hops in routing in different buildings, the performance in terms of cost is still efficient.



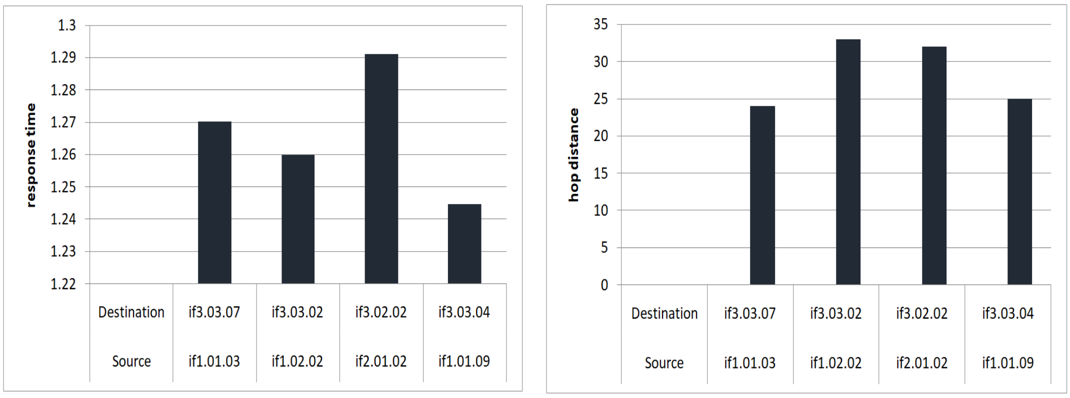
(**a**) (**b**)

Figure 14. Performance of routing between two cells in four cases in the same building and floor. (**a**) Response time performance; (**b**) the distance of hops in the same buildings and floor.



(**a**) (**b**)

Figure 15. Performance of routing between two cells in four cases in the same building and different floors. (**a**) Response time performance; (**b**) the distance of hops in the same buildings and different floors.

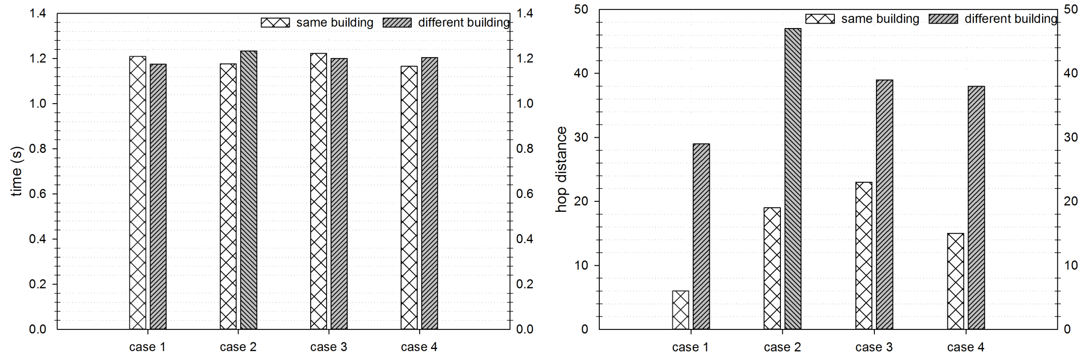


(**a**) (**b**)

Figure 16. Performance of routing between two cells in four cases in different buildings and different floors. (**a**) Response time performance; (**b**) the distance of hops in different buildings and different floors.

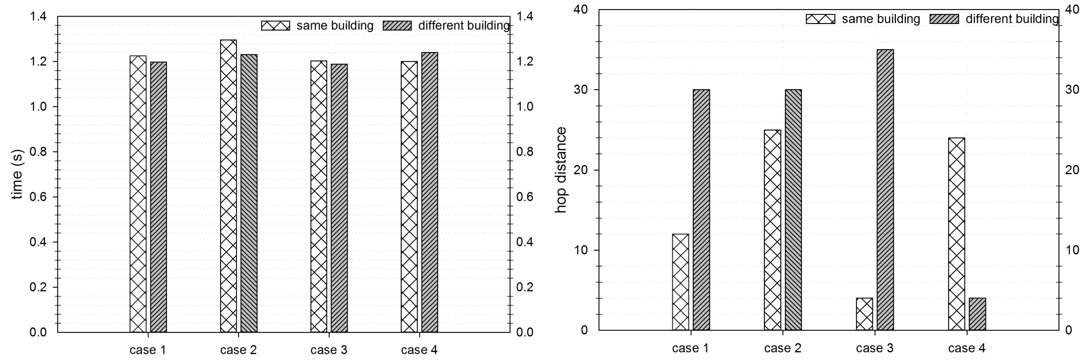
Furthermore, we measure the costs for the proposed system, taking into account the number of moving objects in the cells (the effect of the cells' densities). In this system, we determine whether the routing between two cells has regular cell densities, low cell densities or extreme cell densities. The aim here is to evaluate the effect of the densities on the performance of the system. Note that the number of hops is also considered here, since we test the proposed system in different routing cases and different floors and buildings.

Figure 17 shows the comparison of four routing cases in the same building and different buildings with regular cell densities. This means that the number of moving objects in the majority of cells is identified as regular density (in our previous example, this is between six and 10 persons, which was used in the experiments). We observe that, although the number of hop distances increases, the response time still performs effectively with no significant effects. Figure 18 shows the comparison of four routing cases in the same building and different buildings with low cell densities. Here, we can see that the response time is slightly lower in some cases than for those cases with regular or extreme densities. Figure 19 shows the comparison of four routing cases in the same building and different buildings with extreme cells densities. Here, the response time is slightly higher in most cases compared with regular or low densities. Here, we notice that, although the extreme densities cases have a slightly higher response time, the performance is still efficient, and the density does not have a significant effect on the performance of the proposed algorithm.



(**a**) (**b**)

Figure 17. Performance of the path between two cells in four cases in regular density environments. (**a**) Response time performance; (**b**) the distance of hops in the same and different buildings.



(**a**) (**b**)

Figure 18. Performance of the path between two cells in four cases in a low density environment. (**a**) Response time performance; (**b**) the distance of hops in same and different buildings.

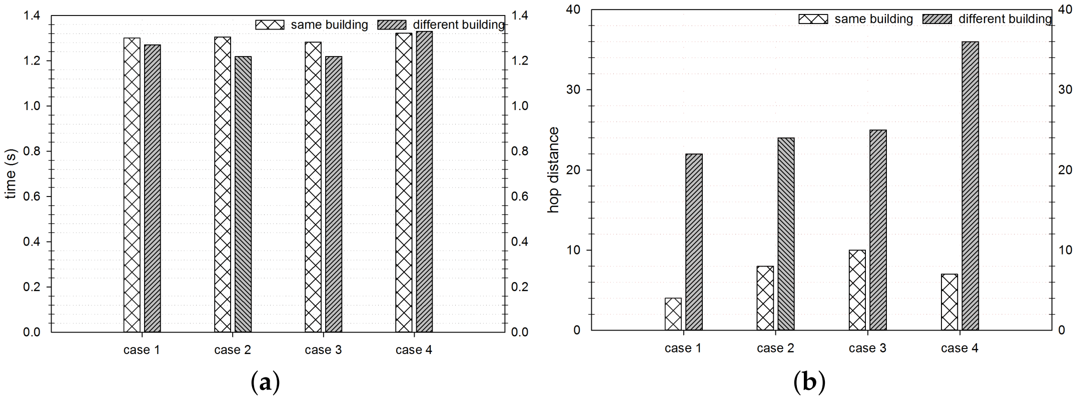


Figure 19. Performance of the path between two cells in four cases in an extreme density environment. (**a**) Response time performance; (**b**) the distance of hops in same and different buildings.

Thus, we can say that the proposed routing algorithm can successfully reduce the false hits when routing the indoor spaces as it considers the densities of cells. Compared with [23], our proposed system reduced the number of false hits by more than eight times in some cases. This system can accommodate the number of hops of the routing system in addition to the cells' densities, which can play an important role in determining the correct shortest path between any cells or moving objects. Moreover, the proposed routing algorithm still performs efficiently without any significant impact resulting from an increase in the number of hops in the same or different buildings. The impact of the increased number of hops in the same/different buildings is less than 2% in some cases, while, in others, it performs more efficiently even with more hops between the objects in the same or different buildings. For example, for the first case of regular density performance, the number of hops of different building is 29 hops and six hops for the same building; however, for the performance in the same building case is is slightly higher (2.4%) compared with the different building, even though it has a higher number of hops. Moreover, a comparison of four routing cases in the same building and different buildings with low cell densities shows that the impact of the extreme density compared with that of lower densities is 7% in some cases. However, in some cases, it is less than 3%. Therefore, the performance is still efficient, and the density does not have a significant effect on the performance of the proposed algorithm.

This algorithm provides a mechanism for retrieving all the possible routes between two cells; however, this may prove to be expensive and may affect the performance of the algorithm. Although we proposed a modified breadth-first search, our aim in future work is to improve the retrieval mechanism in order to enhance the search complexity of the algorithm.

6. Conclusions

This paper addresses the challenge of routing in directed indoor spaces. Most of the current related research has not considered directed indoor spaces. Nowadays, there are many buildings in which the movement of objects is direction-restricted. This is evident on some train station platforms, museums or stores (e.g., IKEA). This work proposed a new routing structure for objects moving in directed indoor spaces. The proposed system exploits the notion of cellular space, which takes into account the indoor walls and partitions that control movement in an indoor environment. Moreover, since most of the current research does not focus on directed indoor space, which has become increasingly important in many applications nowadays, we focus on those particular types of buildings where objects are required to move in a certain direction through a room/corridor in order to control indoor traffic. Furthermore, data density has an essential impact on the routing structure accuracy in indoor spaces. Therefore, this work considers the dense cells in an indoor environment, which significantly increases the accuracy of the routing system. The experiments' results show that the proposed routing algorithm can successfully reduce the false hits in directed indoor routing besides improving the performance efficiently.

This work can be extended in several directions. First, improving the search algorithm for retrieving all possible routes for certain cells or objects is an important extension, which may improve the search complexity of the algorithm. Second, a routing model that takes into account the velocity of the moving objects may improve the routing accuracy. Another interesting focus for future work is to consider the possible laziness or otherwise of moving objects. Some moving objects might prefer to take the shortest path while others favor the longest path possible for fitness purposes.

Acknowledgments: The source of our implementations can be downloaded from the following URL: https://goo.gl/USR9m8.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Alamri, S.; Taniar, D.; Safar, M. A taxonomy for moving object queries in spatial databases. *Future Gener. Comput. Syst.* **2014**, 37, 232–242.
2. Al-Khalidi, H.; Taniar, D.; Betts, J.; Alamri, S. On finding safe regions for moving range queries. *Math. Comput. Model.* **2013**, 58, 1449–1458.
3. Alamri, S.; Taniar, D.; Safar, M.; Al-Khalidi, H. Spatiotemporal indexing for moving objects in an indoor cellular space. *Neurocomputing* **2013**, 122, 70–78.
4. Alamri, S.; Taniar, D.; Safar, M.; Al-Khalidi, H. A connectivity index for moving objects in an indoor cellular space. *Pers. Ubiquitous Comput.* **2014**, *18*, 287–301.
5. Jensen, C.; Lu, H.; Yang, B. Indexing the Trajectories of Moving Objects in Symbolic Indoor Space. In *Advances in Spatial and Temporal Databases*; Part of the Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, German, 2009; Volume 5644, pp. 208–227.
6. Jensen, C.S.; Lin, D.; Ooi, B.C.; Zhang, R. Effective Density Queries on Continuously Moving Objects. In Proceedings of the 22nd International Conference on Data Engineering (ICDE), Atlanta, GA, USA, 3–7 April 2006; p. 71.
7. Abeywickrama, T.; Muhammad, A.C.; Taniar, D. k-Nearest Neighbors on Road Networks: A Journey in Experimentation and In-Memory Implementation. *Proc. VLDB Endow.* **2016**, 9, 492–503.
8. Alamri, S. Indexing and querying moving objects in indoor spaces. In Proceedings of the 2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW), Brisbane, QLD, Australia, 8–12 April 2013; 318–321.
9. Alamri, S.; Taniar, D.; Safar, M. Indexing of Spatiotemporal Objects in Indoor Environments. In Proceedings of the 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), Barcelona, Spain, 25–28 March 2013; pp. 453–460.
10. Raptopoulou, K.; Papadopoulos, A.N.; Manolopoulos, Y. Fast Nearest-Neighbor Query Processing in Moving-Object Databases. Geoinformatica **2003**, 7, 113–137.
11. Sidlauskas, D.; Saltenis, S.; Jensen, C.S. Processing of extreme moving-object update and query workloads in main memory. *VLDB J.* **2014**, 23, 817–841.
12. Chang, J.-W.; Um, J.-H.; LeeP, W.-C. A new trajectory indexing scheme for moving objects on road networks. In *Flexible and Efficient Information Handling*; Part of the Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, German, 2006; Volume 4042, pp. 291–294.
13. Alamri, S.; Taniar, D.; Safar, M.; Al-Khalidi, H. Tracking moving objects using topographical indexing. Concurr. Comput. Pract. Exp. **2014**, 27, 1951–1965.
14. Xie, X.; Lu, H.; Pedersen, T. Efficient distance-aware query evaluation on indoor moving objects. In Proceedings of the 2013 IEEE 29th International Conference on Data Engineering (ICDE), Brisbane, QLD, Australia, 8–12 April 2013; pp. 434–445.
15. Lu, H.; Cao, X.; Jensen, C.S. A foundation for efficient indoor distance-aware query processing. In Proceedings of the 2012 IEEE 28th International Conference on Data Engineering (ICDE), Washington, DC, USA, 1–5 April 2012, pp. 438–449.

16. Ciuonzo, D.; Buonanno, A.; D'Urso, M.; Palmieri, F.A.N. Distributed classification of multiple moving targets with binary wireless sensor networks. In Proceedings of the 14th International Conference on Information Fusion, Chicago, IL, USA , 5–8 July 2011; pp. 1–8.
17. Buonanno, A.; D'Urso, M.; Prisco, G.; Felaco, M.; Meliadò, E.F.; Mattei, M.; Palmieri, F.; Ciuonzo, D. Mobile sensor networks based on autonomous platforms for homeland security. In Proceedings of the 2012 Tyrrhenian Workshop on Advances in Radar and Remote Sensing (TyWRRS), Naples, Italy, 12–14 September 2012; pp. 80–84.
18. Shin, B.-J.; Lee, K.-W.; Choi, S.-H.; Kim, J.-Y.; Lee, W.J.; Kim, H.S. Indoor WiFi positioning system for Android-based smartphone. In Proceedings of the 2010 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 17–19 November 2010; pp. 319–320.
19. Luo, Y.; Hoeber, O.; Chen, Y. Enhancing WiFi fingerprinting for indoor positioning using human-centric collaborative feedback. *Hum.-Centric Comput. Inf. Sci.* **2013**, 3, 1–23.
20. Lassabe, F.; Canalda, P.; Chatonnay, P.; Spies, F. Indoor WiFi positioning: Techniques and systems. *Ann. Telecommun. Ann. Des Telecommun.* **2009**, 64, 651–664.
21. Navizon. Indoor GPS and Indoor Tracking Solutions by Navizon (2014). Available online: http://www.navizon.com/indoors-solutions (accessed on 2 May 2015).
22. Wang, J.; Winter, S.; Langerenken, D.; Zhao, H. Integrating Sensing and Routing for Indoor Evacuation. In Proceedings of the Geographic Information Science: 8th International Conference, GIScience 2014, Vienna, Austria, 24–26 September 2014; pp. 268–283.
23. Dionti, T.A.; Adhinugraha, K.M.; Alamri, S.M. Inter-building routing approach for indoor environment. In *International Conference on Computational Science and Its Applications*; Springer: Berlin/Heidelberg, German, 2017; Volume 10404, pp. 247–260.
24. Alamri, S.; Taniar, D.; Safar, M. Indexing moving objects in indoor cellular space. In Proceedings of the 2012 15th International Conference on Network-Based Information Systems (NBiS), Melbourne, VIC, Australia, 26–28 September 2012; pp. 38-44.
25. Yuan, W.; Schneider, M. Supporting continuous range queries in indoor space. In Proceedings of the 2010 Eleventh International Conference on Mobile Data Management, Kansas City, MO, USA, 23–26 May 2010; pp. 209–214.
26. Saltenis, S.; Jensen, C.S.; Leutenegger, S.T.; Lopez, M.A. Indexing the positions of continuously moving objects. *SIGMOD Rec.* **2000**, 29, 331–342.
27. Alamri, S.; Taniar, D.; Nguyen, K. Vertical Indexing for Mobile Objects in Multi-Floor Environments. *Mob. Inf. Syst.* **2018**, 2018, 4175298.
28. Alamri, S.; Taniar, D.; Safar, M. Indexing moving objects for directions and velocities queries. *Inf. Syst. Front.* **2013**, 15, 235–248.