A SDBMS-based 2D-3D Hybrid Model for Indoor Routing

Hyeyoung Kim, Chulmin Jun and Hyunjin Yi
Dept. of Geoinformatics, University of Seoul
{mhw3n, cmjun, yihyunjin}@uos.ac.kr

***Abstract*—Currently used 3D models, which are mostly focused on visualization of 3D objects and lack topological structure, have limitation in being used for 3D spatial analyses and applications. However, implementing a full topology for the indoor spatial objects is less practical due to the increase of complexity and computation time. In this study, an alternative method to build a 3D indoor model with less complexity using a spatial DBMS is suggested. It is a 2D-3D hybrid data model, which combines the 2D topology constructed from CAD floor plans and the 3D visualization functionality. We show the process to build the proposed model in a spatial DBMS and use spatial functions and queries to visualize in 2D and 3D. And, then, as an example application, we illustrated the process to build an indoor way-finding simulator.**

Keywords - 3D data model; 3D topology; indoor routing; spatial DBMS

I. INTRODUCTION

Currently used 3D models, frequently called 3D GIS, either lack topological structure or are actually 2.5D which have only one z-value for every (x, y) location. Also, since they are mostly maintained as files instead of databases, they have limitations in terms of consistency, shared uses and computation time. However, implementing a full topology for the indoor spatial objects and their complex relationships is less practical due to the increase of complexity and computation time. In this study, we suggest a new method to achieve 3D indoor model with the price of building 2D. We call it a 2D-3D hybrid model, which combines 2D topology and 3D visualization.

We used a spatial DBMS for storing the spatial data, which increases data access speed and flexibility in defining and editing related tables. The PostgreSQL / PostGIS [17], a SDBMS used in this study conforms the OGC (Open GIS Consortium) specifications, which helps portability across different platforms or DBMSs. We showed the process to build the proposed model in a SDBMS and use spatial functions and queries to visualize in 2D and 3D. And, then, as an example application, we illustrated the process to build an indoor way-finding simulator.

II. RELATED WORK

The typical 3D modeling types found in the literature in the last decade are generally categorized into following three:

1. BODY-FACE-EDGE-NODE
2. BODY-FACE-NODE
3. BODY-FACE

Currently available spatial DBMSs (e.g. PostgreSQL, Oracle Spatial, etc.) do not support explicit topological relations between geometric components. Thus studies suggesting *Type 1* ([5], [6], [8]) used standard non spatial tables to achieve topological relations. However, since the cardinality between each pair in *Type 1* (e.g. BODY-FACE, FACE-EDGE and EDGE-NODE) is many-to-many, each pair requires additional association table. This leads to long hierarchical joins to perform queries that use these 4 entities. Study [6] shows the significant increase of query time for window range queries.

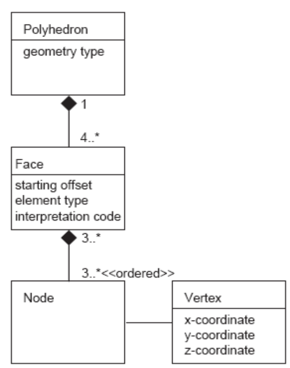

Figure 1. UML diagram of *Type 2* model ([2])

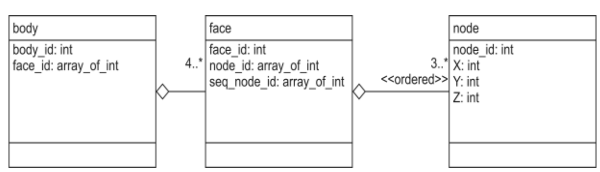

Figure 2. UML diagram of *Type 3* model ([10],[12])

In case *Type 2* and *3*, entity relations are more or less simplified by removing one or two entities ([2], [4], [10], [11], [12]). Just as we see in the de-normalization of relational databases, these simplified types help reducing query time and data accesses but suffer similar redundancy problems. For example, in Figure 2, a cube is composed of six faces, each of which has four nodes. But instead of reusing two nodes when defining the adjacent face, all four nodes are newly assigned for the next face. This redundancy is known to be unavoidable in the current SDBMSs.

Above three types including some variations have been studied focusing on defining exterior building volumes rather than indoor spaces. Indoor spaces would require far more considerations if topology is incorporated. In this study, we are focusing on "navigation" in indoor spaces. Since the navigation takes place on the floor surfaces, all the complex three dimensional relations do not need to be defined. Instead, 2D topological structure as in 2D GIS layers can be used for defining indoor floors. The next section describes how we achieved such a "quasi" 3D model using 2D data structure.

III. 2D-3D Hybrid Data Model

Although [3] mentions "2D-3D hybrid GIS", they used it as a different meaning when they described it as a means to display 2.5D terrain model. In [9], the author and others used the similar term for displaying building exterior and interior in both 2D and 3D. This model used two separate data structures – 2D GIS shapefiles and 3D MAX files. And these two are connected through the shared data table, where each space ID is shared by each other's counterpart space.

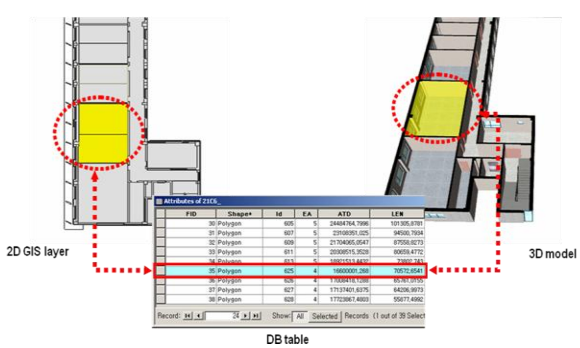

Figure 3. Indoor 2D-3D combination in [9]

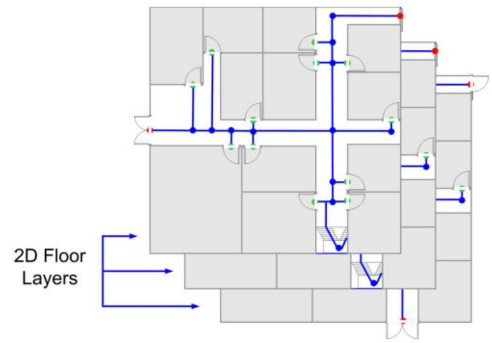

Figure 4. 2D floor layers in 2D-3D hybrid model

Although using file-based structures would suffice when storing and displaying a single building, computational performance becomes degraded significantly as the number or size of buildings increases. As an extended study of [9], we used a spatial DBMS instead of separate files for 2D and 3D. By employing a single DBMS, the need for using a shared data table has now been removed while achieving the diverse advantages DBMSs provide including data sharing, access speed and consistency.

In this new 2D-3D hybrid model, basic structure is the 2D GIS layers. CAD files for floor plans are first converted to GIS shapefiles, and then are stored into PostGIS tables. For indoor navigation, network node-links are separately built along the hallways and also stored in the database (Figure 4). Once data are stored, it becomes possible to use topology-based functions either OGC or PostGIS provides such as Intersects, Disjoint, Covers and Touches.

Figure 5 describes how 2D and 3D views interact with the DBMS. Not like Figure 3 where 2D data are isolated from 3D, all the data are integrated into a single DBMS. Then, both 2D and 3D views are constructed from the data extracted from the DBMS. In our test navigation simulator that will be described later in detail, the 2D interface also designate the origin and destination information to be sent to the DBMS.

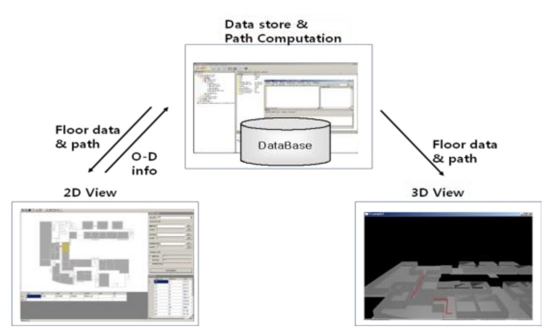

Figure 5. 2D and 3D views from DBMS

IV. An Indoor Routing Application

We applied the proposed 2D-3D hybrid model to developing an indoor way-finding application. The processes for building the system is described in the following.

A. Data construction

CAD-based floor layers are first converted to 2D vector layers, and then are stored in the PostGIS (Figure 6). When converting to 2D layers, we used the ESRI ArcGIS [13] and for storing into the PostGIS, we used the QuantumGIS (often abbreviated to QGIS) which is freely distributed from [14]. The QGIS creates the table schema from the shapefile and fill the table. Although the stored polygon spaces are disjoint with each other inherently without explicit topology, the PostGIS provides many OGC-specified topology functions.

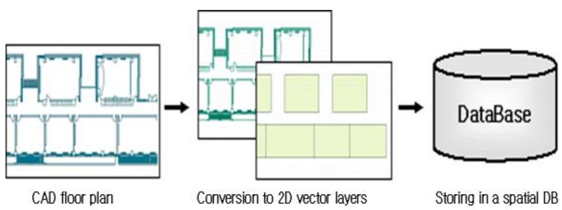

Figure 6. Constructing floor layer data

As in 2D GIS transportation network, we used node-link structure for indoor navigation. Each room or each intersection of links is assigned a node, while a link connects two nodes. In order to connect the floors, the same node-link method is applied along the stairs. This approach was introduced in our previous study [7]. Node and links are stored as two separate tables in the PostGIS each having linestrings and points geometries respectively.

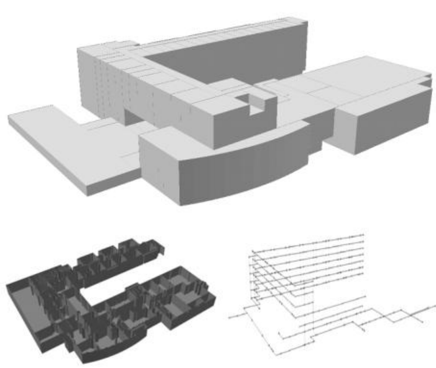

Figure 7. Constructing indoor networks

B. Routing computation using pgRouting

In most routing-related applications including our previous study [9], some kind of shortest path algorithms are implemented or built-in functions are used in the application to read the memory-loaded network data. However, in case of using a DBMS, rather than communicating with the DBMS data from the application, it is far more efficient to use DBMS-provided functions to compute the paths in terms of speed and memory use. In our system, we used a routing function in the pgRouting. The pgRouting is a group of routing-related functions that can be used in the PostGIS ([1], [16]). The algorithms of the pgRouting include Dijkstra, A*, Shooting Star and TSP(Traveling Sales Person problems).

Figure 8 shows an example of the query statement and the results using the pgRouting function and 2D visualization. As shown in the query, we simply call ‘shortest_path’ function with parameters including a source, a target and the field that contains the cost of links. Here we used the lengths of the links as the costs. If we designate the exits as the target nodes, the function computes the shortest evacuation route from the source node. Certain links can be assigned zero or lower values to represent disconnection or some kind of impedances due to fire or smoke during the evacuation.

Methods for Analysis (Query)

```
select * from shortest_path('select id, source, target,
cost from total_21c_link', 124, 135, false, false);
```

Results

데이터의 출력 | 해석 | 메세지 | 히스토리

	vertex_id integer	edge_id integer	cost double preci
1	124	25	7940.53
2	125	16	2578.86
3	126	29	1307.86
4	128	30	2826.33
5	129	17	1971.46
6	130	18	11669.3
7	131	19	3882.53
8	132	36	3165.49
9	133	37	2508.11
10	134	26	979.514
11	135	-1	0

Visualization

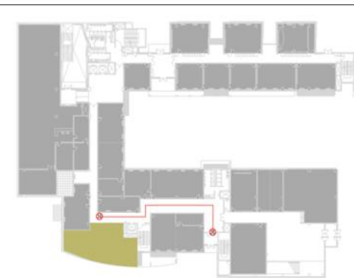

Figure 8. Computing paths using pgRouting

C. *2D and 3D visualization*

Our system consists of two main visual modules, the first being the 2D floor viewer and second the 3D viewer (Figure 9). As described in Figure 5, each module interacts with the DBMS. The user-specified 2D floors are queried in the DBMS and read into the 2D module. Here, the user may be able to perform queries either by typing attribute values or selecting the spaces on the floor map. The queried data are displayed with colored symbols in the map and the data table. Our system is developed using C# and as the 2D visualization module, we used a .NET-compatible GIS library [15].

The 3D viewer is implemented using OpenGL. The queried geometry components of building floors are vertically layered using the z values of the floor data. Then, the walls are extruded from the 2D polygons using the wall heights stored as attribute values in the table.

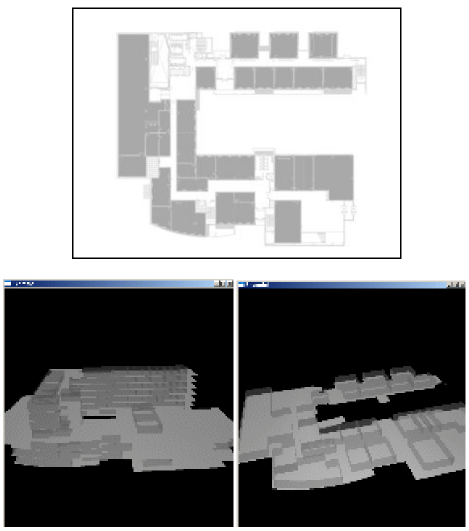

Figure 9. 2D and 3D views in the system

D. *Indoor routing tests*

We tested our system on a fire evacuation case. Although a real situation would require identifying a person's location by real time positioning sensors such as RFIDs, we assumed that we have acquired the position and replaced it with user inputs. The user selects a room or a node as the source point and then it is sent to the DBMS to be used as a parameter in the pgRouting shortest_path function. The resulting paths are displayed either in 2D module or 3D module as the user need. In case of normal way-finding, the user would also need to specify the destination node.

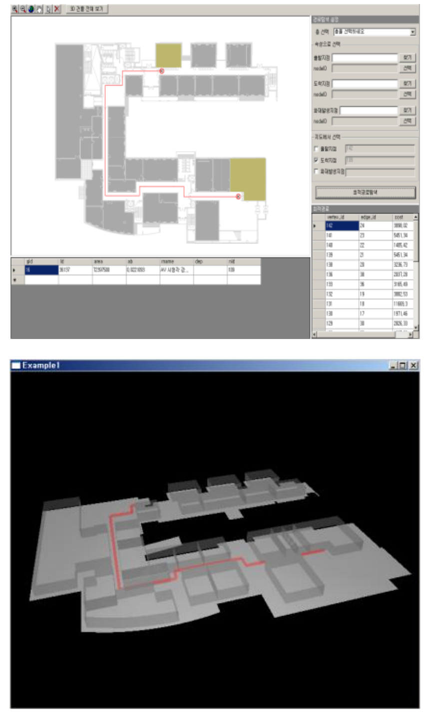

Figure 10. Displaying paths in 2D & 3D modules

Figure 10 shows the computed evacuation path displayed in 2D and 3D modules. If the path passes across multiple floors, the user can load the floor layers by switching them in the interface.

On top of indoor positioning sensors, other fire-related sensors such as temperature or smoke sensors may be used. Figure 11 illustrates how our system can interact with fire sensors.

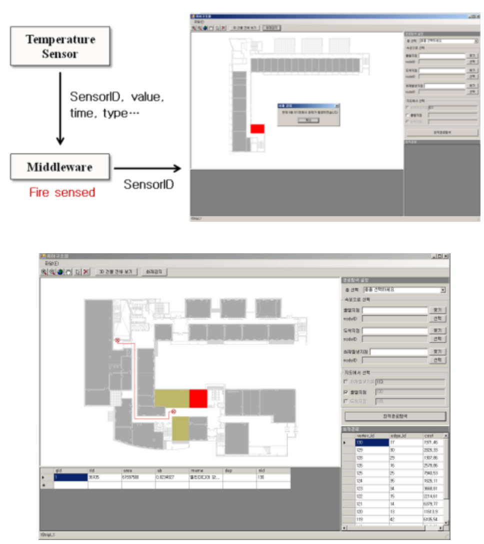

Figure 11. Interacting with temperature sensors

For example, a temperature sensor may transfer the current temperature to the middleware which determines whether it is unusually high. Then, the system receives the sensorID which is mapped on the floor layer. Then, this information is also sent to the DBMS where the evacuation paths are generated avoiding fire or other obstacles. The figure in the bottom in Figure 11 shows how the system generates an alternative route avoiding the fire instead of providing the shortest path to the nearest exit.

V. Concluding Remarks

This study suggested an alternative method to build a 3D model at a lower cost and we called it a 2D-3D hybrid data model. Instead of considering complicated relationships between 3D objects, we used 2D floor layers as a base data structure. Maintaining 2D surface information sufficed for such applications as indoor navigation. Storing data in a spatial DBMS provides more advantages than using file-based models such as better computational speed, maintainability and data sharing. The data in a SDBMS can be accessed and displayed in 2D and 3D. Although in our system, we stored 2D layers and extruded to 2.5D using the wall height values in visualization time, these extruded walls could also be stored in the DBMS as needed. We used free softwares or libraries for SDBMS and 2D/3D modules, which can help reducing costs for development and maintenance. As described in the system test, our SDBMS-based evacuation system can be effectively integrated with real time positioning or fire sensors.

Acknowledgements

This research was supported by a grant(08KLSGC04) from Cutting-edge Urban Development - Korean Land Spatialization Research Project funded by Ministry of Land, Transport and Maritime Affairs.

References

[1] Allan, A., “Multicriteria Decision Analysis for Topological Routing Using PostgreSQL/PostGIS and pgRouting”, University of Texas at Dallas GIS Workshop - Summer 2008, 2008.

[2] Arens, C., Maintaining reality: modelling 3D spatial objects in a GeoDBMS using a 3D primitive. M.Sc. Thesis, Delft University of Technology, The Netherlands, 2003.

[3] Brooks, S. and Whalley J., "A 2D/3D Hybrid Geographical Information System", In Proc. of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia, pp. 323-330, 2005.

[4] Chen, T., A. Abdul-Rahmana, and S. Zlatanova, "3D Spatial Operations for geo-DBMS: geometry vs. topology", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B2. Beijing, 2008.

[5] Ellul, C. and Haklay, M.M., “Using a B-rep structure to query 9-intersection topological relationships in 3D GIS”, In LNG&C , Springer, pp. 127-151, 2008.

[6] Gröger, G., Reuter, M. and Plümer, M., "Representation of a 3-D city model in spatial object-relational databases", In Proc. of the 20th Congress of International Society for Photogrammetry and Remote Sensing, Istanbul, Turkey, 2004.

[7] Kim, H., Jun, C. and Kwon, J., "Syntax-based Accessibility for 3D Indoor Spaces", J. Korean Society for Geospatial Info. Sys., 15(3): 11-18, 2007.

[8] Kolbe, T. H. and Gröger, G., “Towards Unified 3D-City-Models”, In Proc. of. ISPRS Commission IV Joint Workshop on Challenges in Geospatial Analysis, Integration and Visualization II, Sept. 8-9, Stuttgart, Germany, 2003.

[9] Park, I., Jun, C. and Choi, Y., "3D-GIS Network Modeling for Optimal Path Finding in indoor Spaces", J. Korean Society for Geospatial Info. Sys., 15(3): 27-32, 2007.

[10] Stoter, J.E. and van Oosterom, P.J.M., “Incorporating 3D geo-objects into a 2D geo-DBMS", ACSM-ASPRS 2002.

[11] Stoter, J.E. and S. Zlatanova, “Visualising and editing of 3D objects organised in a DBMS”. Proceedings EUROSDR Workshop: Rendering and Visualisation, pp.14-29, 2003.

[12] Zlatanova, S., “3D GIS for urban development”, PhD thesis, Institute for Computer Graphics and Vision, Graz University of Technology, Austria, ITC, the Netherlands, 2003.

[13] http://www.esri.com

[14] http://www.osgeo.org

[15] http://www.codeplex.com/SharpMap

[16] http://pgrouting.postlbs.org

[17] http://postgis.refractions.net