Construction operators for modelling 3D objects and dual navigation structures

Pawel Boguslawski, Christopher Gold

Faculty of Advanced Technology, University of Glamorgan
Pontypridd, CF37 1DL, UK
{pbogusla, cmgold}@glam.ac.uk

Abstract. This work presents new operators for construction of 3D cell complexes. Each cell in a complex is represented with the Augmented Quad-Edge (AQE) data structure. Cells are linked together by the dual structure and form a mesh. This structure stores information about the geometry and topology of a modelled object. Navigation in the mesh is possible using standard AQE operators. A new set of atomic operators was developed to simultaneously construct the Primal and the Dual structures. This allows 3D tetrahedralization as well as the construction of different types of objects, such as buildings composed of multiple rooms and floors.

Keywords: 3D structure, modelling, AQE, emergency response, disaster management

1 Introduction

3D modelling is getting very important in many fields of science and industry. Current methods are often insufficient because many do not support complex connected 3D structures, and it is impossible or very difficult to integrate different models to analyze more complex influences. An example is geology where 3D models are essential and very often other, lower dimensional ones are not adequate. Very important is spatial connection and the influences between layers - for example between different rock formations. Another field is GIS and in particular modelling of buildings. Disaster management systems, simulation of terrorist attacks and looking for escape routes from buildings and other multi-level structures in urban areas are vital especially after 9/11. Many emergency management systems use 2D models and do visualization in 3D. Others use 2.5D models to navigate inside buildings [1], but 3D models are essential to this issue. Results presented by Kwan and Lee [2] show that extending 2D building models to 3D can significantly improve the speed of rescue operations. The next improvement that gives better calculations in a model is the combination of geometry and logical models [3] [4]. The logical model would be used for compuing escape routes and the geometry model for visualization. Lee in his work [5] described current 3D methods of modelling with particular reference to emergency management. He presents a 3D Navigable Data Model (NDM) to support building evacuation. This is a digital geographic database of a transportation system that can support emergency guidance operations at a high level. An emergency situation in a real building was simulated.

The most essential item in each case is topology - adjacency and connectivity between objects. In a general topological model a dual navigation graph is derived from the geometric one (primal outline graph). For real-time updating we need automatic dual graph construction during dynamic (insertion and deletion) primal graph maintenance. New structures which will store both of them simultaneously will improve the efficiency of calculations significantly in the above issues.

2 Related works

3D data models can be classified into: Constructive Solid Geometry (CSG), boundary-representations (b-rep), regular decomposition, irregular decomposition and non-manifold structures [6]. For our research b-reps and irregular decomposition models are the most relevant. The well known b-reps are: Half-Edge [7], DCEL [8], Winged-Edge [9] and Quad-Edge [10]. Irregular decomposition models (e.g. for constructing a 3D Delaunay tetrahedralization) can be constructed with a Half-Faces [11], G-maps [12] and Facet-Edges [13]. The most important for us are the Quad-Edge (QE) and (its extension) the Augmented Quad-Edge (AQE) [6] data structures. These structures are suitable to construct models and their duals at the same time. We use dual space to connect cells in cell complexes and to navigate between them. Navigation and data structures are the same in both spaces. Eventually both spaces are connected together and we do not need any additional pointers for this connection. Other data structures like Half-Edge or Winged-Edge used widely in CAD systems do not provide for management of the duality. The only similar structure which preserves primal and dual 3D subdivisions is the Facet-Edge. In a cell complex modelled with this structure common faces are not stored twice – faces are shared by two cells. This is less memory consuming than the AQE structure [6]. However we decided to use AQEs in our project as we need to keep common faces separate because of the nature of the modelled objects. We believe that our solution is more suitable for real-time simulations and computation.

2.1 Quad-Edge

The Quad-Edge (QE) was introduced by Guibas and Stolfi [10]. Each quad consists of three pointers: R, N and V. Four quads connected together in an anticlockwise (CCW) loop by the R pointer form a full Quad-Edge. The Next pointer N points to the next edge (with the same shared vertex or face). All edges connected by this pointer form a loop. This is a CCW connection as well. The pointers R and N are directly used in Rot, Sym and Next standard navigation operators [10]. Rot uses R and returns the next quad from a loop of four. Sym calls Rot twice. Next uses N and returns the next edge from the loop (Fig. 1). The V pointer is used to point to vertices in the structure.

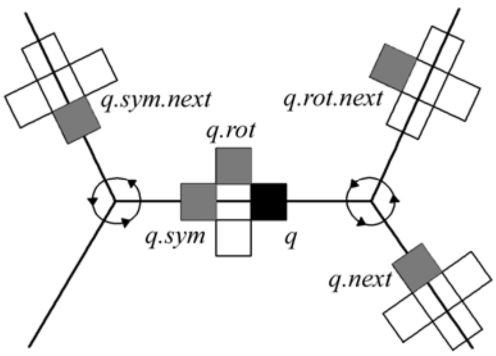

Fig. 1. Navigation in a structure with Quad-Edges; q represents origin quad.

To construct and modify graphs only two operations are needed – MakeEdge to create a new primal/dual edge pair, and Splice to connect/disconnect it to/from the graph under construction. Tse and Gold showed that Euler Operators could easily be developed on the basis of QEs rather than half-edges or winged-edges [14].

The QE structure was originally used to describe both the primal and dual graphs simultaneously. The particular example was triangulation modelling, showing both the primal Delaunay triangulation and the dual Voronoi tessellation. Either graph may be navigated by simple pointer-following, using Rot and Next pointers, with "Vertex" pointers to primal and dual graph nodes.

Duality of structures is very important for many purposes. As stated in the previous section, many applications demand that 3D models include geometry and a graph of connections between elements at the same time. Having dual graphs at the same time are also advantageous for 2 and 3 dimensional electromagnetic simulation [15].

2.2 Augmented Quad-Edge

In 3D, primal and dual graphs may also be defined: the dual of a face is an edge, the dual of an edge is a face, the dual of a node is a volume and the dual of a volume is a node. Ledoux and Gold defined an extension of the Quad-Edge - the Augmented Quad-Edge (AQE) [6]. In this approach the "Vertex" pointer to a face (usually unused) was assigned to the dual edge of that face. This was called "Through" and allowed navigation between cells, via the dual edge. "Adjacent" is the next operator that uses the dual structure. It is a combination of Through and Next operators. It is used to reach the adjacent cell (e.g. the next room) in the structure (Fig. 2). If we are in one room/cell and want to go to the next one we have to find a wall/face between those two rooms/cells (this is quad q_f in fig. 2) and use the Adjacent operator (q_f.adjacent). The result gives an adjacent wall/face in the next room/cell. The original QE operators were restricted only to a single cell.

The target application of the AQE was the 3D Voronoi/Delaunay structure. The difficulty with this model was that the construction operators were complex, in particular the implementation of the Through pointer maintenance.

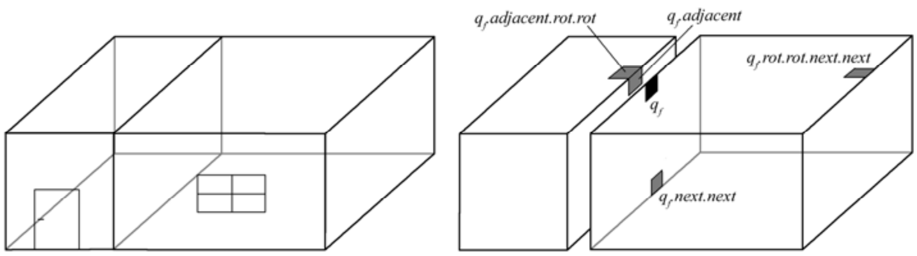

Fig. 2. The AQE structure is a suitable structure for modelling 3D objects – e.g. building interiors. The set of operators allow for navigation inside one cell as well as between cells; q_f represents the origin quad. [6]

2.3 GML/ISO19107

In GIS systems the ISO 19107 standard is used widely. It uses the GML schema to describe objects. For example, a simple building composed of rooms consists of a collection of Solid components, each bounded by an ExteriorSurface composed of SurfacePatches which have one exteriorRing each. This LinearRing has a sequence of coordinate positions, or a sequence of Point components each with a coordinate position. There is no navigation mechanism between Solids, or between Surfaces/Patches on the same Solid. This must be done by matching coordinates, or Point IDs, for the LinearRing defining the SurfacePatch.

Our data structure has a “Next” pointer permitting navigation within the ExteriorSurface (“Shell”) of any Solid. By using the Next pointer of the dual we may navigate between adjacent Solids. Our Solid is a volumetric component distinct from its Shell. Our structure handles lower-dimensional components, e.g. partly connected walls or edges. It can easily be exported to GML by navigating the graph structure and outputting the various components as they are encountered.

GML currently assumes matching coordinates or Point IDs. To import from GML we must make these matches between GML components to reconstruct the adjacency structure. Where coordinates are only approximate and matching points have approximately similar coordinates some tolerances must be used. This is ongoing research.

3 Current work

The work reported here is the continuation of a larger project. Ledoux and Gold [6] saw the need for a data structure for the simultaneous maintenance of both the primal and dual spatial subdivisions. Although the dual can always be obtained from the primal, there are many applications where it is needed directly: an obvious example is for the navigation from primal cell to primal cell. Especially for disaster management, where repeated traversals are needed for real-time shortest-path planning through a changing network, such processes are basic. They demonstrated that, with a space-filling set of primal cells and another space-filling set of dual cells (where each cell is a simple 2-manifold shell), it was possible to provide a mechanism for navigation throughout. It was based on the observation that the dual-entity of a face in one space is an edge in the other. Thus transforming the pointer to a face within a cell in the first space into a pointer to the corresponding edge in the second space gave a link between the two – and hence a way to navigate. This Augmented Quad-Edge was a direct modification of Guibas and Stolfi's [10] Quad-Edge structure.

We were already familiar with the Voronoi-Delaunay duality in 3D, and so this was our test-bed to validate the principle and develop functional code. It was entirely successful and, for example, drawing a single Voronoi cell or a single Delaunay tetrahedron was greatly simplified. The algorithms used were the standard incremental insertion and flipping methods of Shewchuk [16] and others. There was a great deal of local searching and sewing needed to maintain the primal and dual structures simultaneously. An effect of a simple tetrahedralization is shown in the Fig. 3.

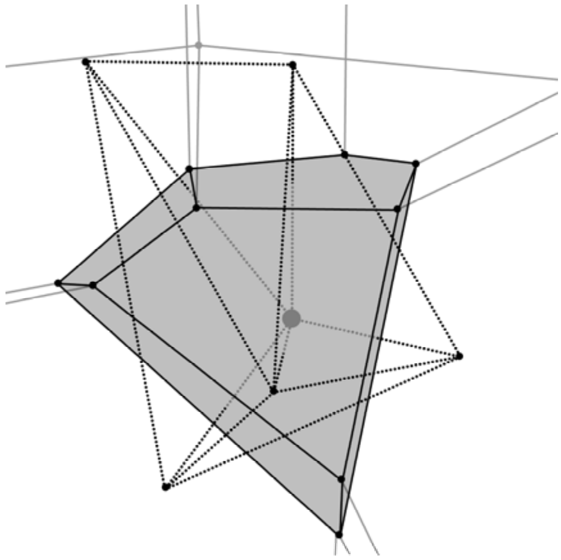

Fig. 3. Tetrahedralization. Gray cell with black edges is a Voronoi cell. This cell enclosed one Delaunay point (big gray vertex). Dotted black edges form Delaunay tetrahedral cells.

However, rooms in buildings, or subsurface catacombs, are not easily made directly from Voronoi cells – so the point-insertion and flipping algorithms were inappropriate. The current work attempts to produce a set of local, and general, construction operators that allow the maintenance of the dual structure as a by-product of the (manual) construction of buildings from floor-plans, stairways, etc.

This is in principle very similar to the Euler Operators developed for CAD systems (e.g. Mantyla [7], Lee [17]) –, but with the ability to create and link multiple simple shells (as in non-manifold CAD systems) – and with the simultaneous maintenance of the dual. We are still working on the construction of exact equivalents of Euler

Operators, but we have developed an alternative approach, based on double-face elements, that can achieve similar results.

In the Quad-Edge structure there are four individual pieces ("Quads") associated with each edge of a simple oriented 2-manifold. These are linked (by pointers in our implementation) so that the "first" has a pointer to a vertex, the "third" points to the other vertex forming the edge, the "second" (anticlockwise) points to the right-hand face and the "fourth" points to the left-hand face. An additional "Next" pointer in each Quad points to the next Quad-Edge in anticlockwise order around the associated vertex or face. The (2D) dual is represented, and one may navigate directly around e.g. Voronoi cells or Delaunay triangles. Frequently the face pointers remain unused, unless an explicit face object is desired, e.g. to assign an attribute such as colour.

We observed above that in 3D each face of a cell is penetrated by a dual edge. Indeed, it is penetrated by one edge for each of the vertices of the face – each of these is surrounded by its own dual shell. Thus if the face pointer is re-assigned to point to a dual edge, then this need never be changed. (Inversely, the related face pointer of the dual edge points back and also remains unchanged.) Construction is thus simplified if these are created together, already linked. However, an edge has two sides, with associated faces, and during construction low-level elements need to be linked to form the final structure. Initially our atomic elements were complete edges – but this necessitated the updating of the re-assigned face pointers, which is undesirable.

Our final choice was to create an atomic element from a "half edge" (essentially two of our four Quads) together with the linked half-edge in the dual. This reduced our construction process to the snapping-together of the appropriate half edges, to form complete edges. This atomic element is shown in Fig. 4.

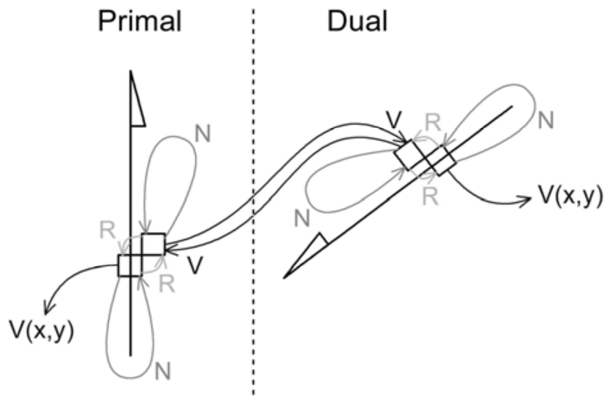

Fig. 4. Half Edges are created in the primal and dual. They are linked permanently.

Our current construction method starts by building one side of a complete "double face" – a "half-face", then the other, and then joining them to form a "sandwich". Fig. 5 shows the process to connect half-edges together in primal space, and to make the necessary connections in the dual. The result is a half-face based on the building design. This is repeated for the matching half-face.

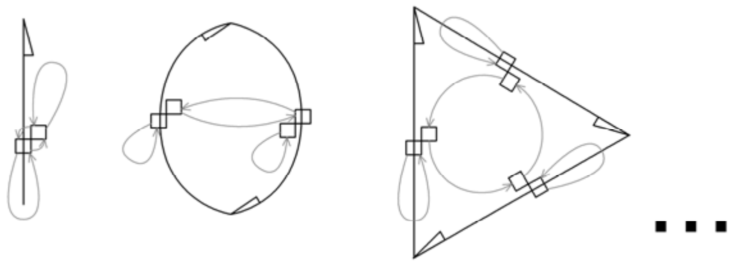

Fig. 5. Construction of a half face in the primal. Half edges are added one by one to the loop. The Dual is not shown.

These two are combined to form the completed “double face” by snapping the matching half-faces together (Fig. 6). The result is a completely defined shell – although of zero volume.

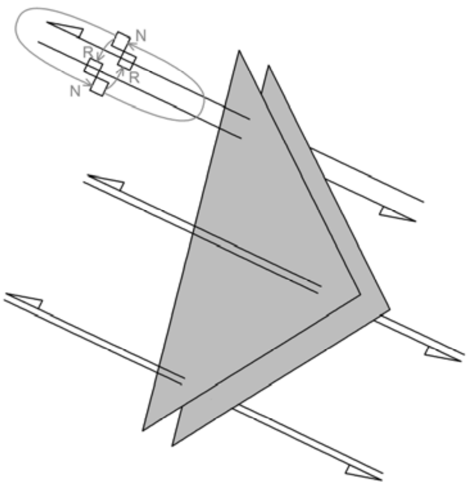

Fig. 6. Construction of a double face. Half edges in the primal and dual are linked together and now form full edges.

These “walls” (or “floors”, etc.) are then joined together as desired to give our final building structure. As all operations are defined to be reversible, one edge at a time of the double-face is un-snapped, the same is done for the receiving face, and then the half-faces are snapped back together in the new configuration (Fig. 7). This directly mimics the construction of a cardboard model using sticky tape!

Visualization of a model is very easy. To get all points, edges or faces of a cell we use breadth –first traversal of the graph. All cells are connected by a dual structure which is also a graph.

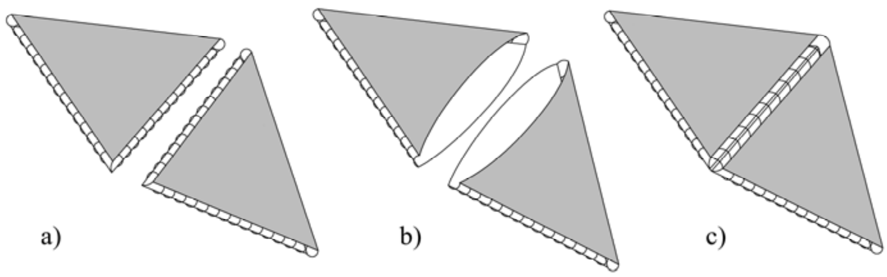

Fig. 7. Snapping two faces: a) two separate faces; b) unsnapping edges; c) snapping edges into new configuration.

The process of construction was first validated using the previous Voronoi-Delaunay model (see Fig. 3). It satisfactorily replaces the complex stitching process previously used. It was then used to construct both simple and more complex building structures (see Fig. 8). Note that the sequence of “manual” operations is relatively straightforward.

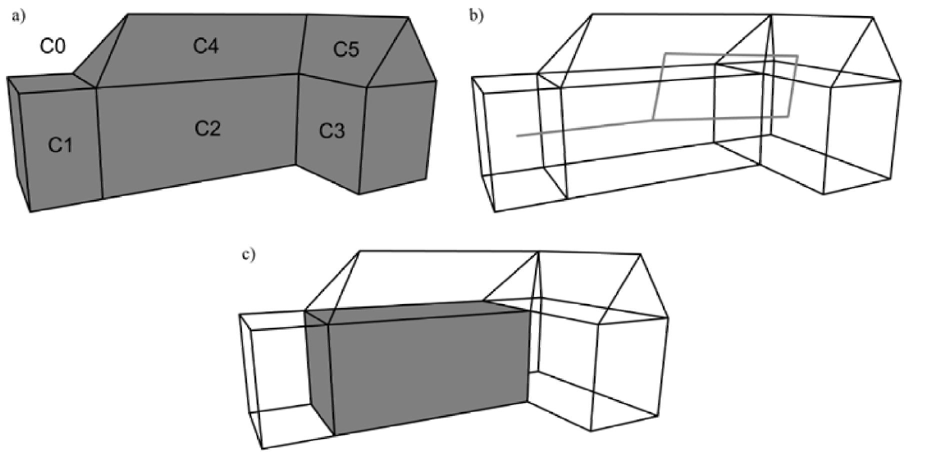

Fig. 8. Objects constructed using AQEs and the new construction operators; a) model with faces shown; C1-C5 – cells of a model, C0 – outside cell, b) mesh of edges; black – primal and grey – dual structure and c) one volume/room selected.

There are 98 edges used in this model of a building grouped in six cells of one cell complex. Five cells (C1-C5) form the building and one (C0) is its outside. Outside is an environment of a building. It can be air, ground etc. We need this outside cell for navigation and to have properly built dual graphs. Each edge in the model consists of four quads and each edge in the primal has been assigned two half edges in the dual. This gives us 784 quads. To obtain the number of pointers we have to multiply the number of quads by three. In total we have 2352 pointers in the model. There are also 26 vertices (21 in the primal and 5 in the dual).

The number of elements in this model is too high to show the process of construction, so we present only a middle part of the building to make this example clear (Fig. 9). All steps in Table 1 are high level operators used to construct the model. The MakeFace operator takes a list of points (Pxx) to create a face. The SnapFaces operator takes two quads (Qxxx) from faces to snap them together. Finally we have three cells C2, C4 and C0. C0 is outside of the model.

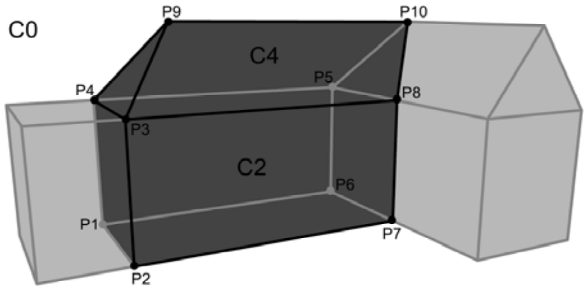

Fig. 9. Process of construction is presented only for a middle part of a building.

Table 1. Process of the model construction. Only high level operators are shown.

Cell 2	**Cell 4**
MakeFace(P1,P2,P3,P4) – left face	MakeFace(P4,P3,P9) – left face
MakeFace(P1,P4,P5,P6) – back face	SnapFaces(Q193,Q153)
SnapFaces(Q33,Q1)	MakeFace(P3,P8,P10,P9) – front face
MakeFace(P2,P1,P6,P7) – bottom face	SnapFaces(Q217,Q149)
SnapFaces(Q65,Q5)	SnapFaces(Q217,Q201)
SnapFaces(Q65,Q33)	MakeFace(P8,P5,P10) – right face
MakeFace(P3,P2,P7,P8) – front face	SnapFaces(Q249,Q145)
SnapFaces(Q97,Q9)	SnapFaces(Q249,Q225)
SnapFaces(Q97,Q65)	MakeFace(P5,P4,P9,P10) – back face
MakeFace(P4,P3,P8,P5) – top face	SnapFaces(Q273,Q157)
SnapFaces(Q129,Q13)	SnapFaces(Q273,Q193)
SnapFaces(Q129,Q97)	SnapFaces(Q273,Q229)
SnapFaces(Q129,Q41)	SnapFaces(Q273,Q257)
MakeFace(P6,P5,P8,P7) right face	
SnapFaces(Q161,Q45)	
SnapFaces(Q161, Q141)	
SnapFaces(Q161, Q109)	
SnapFaces(Q161, Q77)	

We do not have to construct the bottom face of C4 explicitly. The faces of C4 are connected to C2. The top, two-sided face from C2 is shared to give the bottom face of C4.

In all cases the dual graph – a necessity for building navigation and real-time escape route planning – is constructed on-the-fly.

We believe that this construction process is both satisfactory and intuitively clear when using the higher-level commands. It is not presented in this paper but each operator has an inverse version or is self-reversible, although no formal proofs have yet been developed. We are exploring the relationship with conventional CAD Euler Operators, and we hope to provide equivalent operations in the near future – with automatic dual construction as the major requirement. Overall, the ready availability of the dual graph should provide new insight into spatial relationships – for navigation, sound propagation, the solutions of Maxwell's Equations [15] and many other applications.

4 Future work

Euler operators The next issue we want to undertake is to develop a set of Euler operators. This will open the door to CAD system development. Many CAD applications implement a boundary representation using Winged-Edge and Half-Edge structures [17, 18]. These structures do not include a dual graph, which is essential in emergency management systems. We think that it is possible to implement duality in Euler operators for cell complexes: it works for single cells. Future work will show if more complex operations on many connected cells are possible.

Storage Many GIS 3D applications are based on a tetrahedral mesh stored in a database such as Oracle [19]. Modern large database engines support 3D geometry storage and operations. Our approach comes from graph theory rather than database design. In one solution in [19] only tetrahedral shapes are allowed – ours is more general. Our future task is to develop methods for saving our graph structures in a database or on disk.

We can assign unique IDs to points and quads in a structure. Each cell in one space encloses exactly one point in dual space. Thus a solid can be identified by the ID of the dual point, and its boundary is associated with the dual point. In addition faces have dual penetrating edges and thus take the edge ID.

Overlapping Currently coordinates of the vertices of adjacent cells in the same cell complex must be identical. In many models objects penetrate each other due to data errors. We would like to include mechanisms in the future that can manage this issue.

5 Applications

To prove the correctness of our algorithms we developed a simple application. We used Delphi to write it and an in-house graphics engine based on OpenGL for visualization. This engine was written for previous projects. To be sure that the presented methods are suitable for modelling building interiors we reconstructed the simple project (Fig. 10a) originally presented in [5].

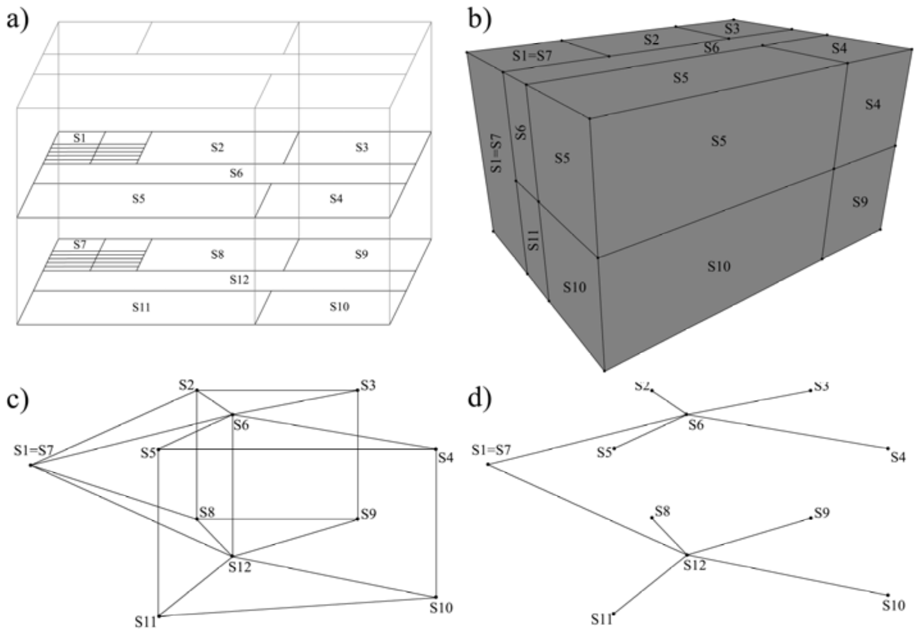

Fig. 10. Example structure representing a 3D model of a building interior. S1, S7 – staircases, S2-S5, S8-S11 rooms, S6, S12 - corridor; a) spatial schema – original version [5], b) volumetric model of rooms, c) complete graph of connections between rooms, d) graph of connections between rooms – only passages.

In our approach the geometry of the building is in the primal (Fig. 10b) and the topology in the dual space – a connection graph (Fig. 10c). Each point (S1-S12) in this graph represents a separate room, corridor or stair cage. Connections between these objects are represented by edges. Edges have weights as attributes. The decision whether there is a passage or not depends on this weight. A strong connection means that there is a passage like a door, window, etc. A weak connection means no passage, but the possibility to go to the next room still exists. This model predicts such situations as making holes in walls and the creation of new passages. When weak connections are removed from the graph, we have the same logical network as in the case of [5] (Fig. 10d). The only difference between the models is that we built the staircase as one room (originally the stair case was divided into two levels).

6 Conclusions

The 3D structure and construction operators presented here create a solid base for 3D models where both geometry and topology are essential. It seems to be valid for modelling and navigation within spatial objects like multilevel buildings in urban areas. The structure consists of two graphs. The first one represents the geometry i.e. rooms in a building and the second one indicates the connectivity between objects (rooms). We do not need to manipulate the dual space directly. It should be noted that we have achieved this by creating and joining individual faces only in the primal space. We achieved a straightforward mechanism for 3D volumetric modelling which preserves the dual space for navigation, useful for example in disaster management, with no additional effort. Other construction operators are also possible – e. g. standard CAD Euler Operators. We are exploring these.

Real-time systems for emergency response are currently very important. They can save many human lives and much money in hazardous situations. The lack of a good 3D model has been a problem. Perhaps our structures and methods will be useful in real-time applications in the future, and they will improve existing solutions.

Acknowledgments This research is supported by the Ordnance Survey and EPSRC funding of a New CASE award.

References

1. Raper, J., Slingsby, A.: Navigable Space in 3D City Models for Pedestrians. Advances in 3D Geoinformation Systems. Peter van Oosterom, Sisi Zlatanova, Friso Penninga, Elfriede Fendel (Eds.) Springer, pp. 49-64 (2007)
2. Kwan, M-P., Lee, J.: Emergency response after 9/11: the potential of real-time 3D GIS for quick emergency response in micro-spatial environments. Computers, Environment and Urban Systems, vol. 29, pp. 93-113 (2005)
3. Pu, S., Zlatanova, S.: Evacuation Route Calculation of Inner Buildings. In: Geo-Information for disaster management. Eds P. van Oosterom, S. Zlatanova and E. M. Fendel, pp. 1143-116. Springer Verlag, Heidelberg (2005)
4. Meijers, M., Zlatanova, S., Pfeifer, N.: 3D geo-information indoors: structuring for evacuation. In: The First International Workshop on Next Generation 3D City Models, G. Groeger & T. H. Kolbe (Eds.), pp. 11-16 (2005)
5. Lee, J.: A Three-Dimensional Navigable Data Model to Support Emergency Response in Microspatial Built-Environments. Annals of the Association of American Geographers, Blackwell Publishing, vol. 97, pp. 512-529 (2007)
6. Ledoux, H., Gold, C. M.: Simultaneous storage of primal and dual three-dimensional subdivisions. Computers, Environment and Urban Systems 31 (4), pp. 393-408 (2007)
7. Mantyla, M.: An introduction to solid modelling. Computer Science Press, New York, USA (1998)
8. Muller, D. E., Preparata, F. P.: Finding the intersection of two convex polyhedra. Theoretical Computer Science, vol. 7, pp. 217-236 (1978)
9. Baumgart, B. G.: A polyhedron representation for computer vision. In: National Computer Conference, AFIPS (1975)
10. Guibas, L. J., Stolfi, J.: Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, ACM Transactions on Graphics, vol. 4, pp. 74-123 (1985)
11. Lopes, H., Tavares, G.: Structural operators for modelling 3-manifolds. In: Preceedings 4th ACM Symposium on Solid Modeling and Applications, Atlanta, Georgia, USA, pp. 10-18 (1997)
12. Lienhardt, P.: N-dimensional generalized combinatorial maps and cellular quasi-manifolds. International Journal of Computational Geometry and Applications, vol. 4 (3), pp. 275-324 (1994)

13. Dobkin, D. P., Laszlo, M. J.: Primitives for the manipulation of three-dimensional subdivisions. Algorithmica, vol. 4, pp. 3-32 (1989)
14. Tse, R. O. C., Gold, C. M.: TIN Meets CAD - Extending the TIN Concept in GIS. Future Generation Computer systems (Geocomputation), vol. 20(7), pp. 1171-1184 (2004)
15. Sazanov, I., Hassan, O., Morgan, K., Weatherill, N. P.:Generating the Voronoi –Delaunay Dual Diagram for Co-Volume Integration Schemes. In: The 4th International Symposium on Voronoi Diagrams in Science and Engineering 2007 (ISVD 2007), pp. 199-204 (2007)
16. Shewchuk, J. R.: Delaunay Refinement Mesh Generation. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburg, USA (1997)
17. Lee, K.: Principles of CAD/CAM/CAE system, Addison-Wesley/Longman, Reading (1999)
18. Stroud, I.: Boundary Representation Modelling Techniques, Springer (2006)
19. Penninga, F., van Oosterom, P.: First implementation results and open issues on the Poincare-TEN data structure, Advances in 3D Geoinformation Systems. Peter van Oosterom, Sisi Zlatanova, Friso Penninga, Elfriede Fendel (Eds.) Springer, pp. 177-197 (2007)