Document Title:	**Semi-Automated Processing and Incorporation of 3D Geo-Coding Extensions to Commercial Buildings for Emergency Response Applications**
Author:	**Bill Ribarsky, Kalpathi Subramanian, Jianfei Liu, Kyle Lyons, Onyewichi Obirieze**
Document No.:	**237987**
Date Received:	**March 2012**
Award Number:	**2007-DE-BX-K010**

Semi-Automated Processing and Incorporation of 3D Geo-Coding Extensions to Commercial Buildings for Emergency Response Applications

Bill Ribarsky(PI)* Kalpathi Subramanian(Co-PI)† Jianfei Liu‡ Kyle Lyons§ Onyewichi Obirieze¶

NIJ Award 2007-DE-BX-K010:Phase 1 Final Report

Charlotte Visualization Center, Dept. of Computer Science
The University of N. Carolina at Charlotte, Charlotte, NC 28223, USA

Abstract

In this paper, we describe new automation tools to process 2D building geometry data, in order to facilitate emergency response to critical events in commercial buildings. Given the scale and complexity of commercial buildings, such tools are essential for effective response and communication during an emergency. Starting from CAD building files, our data processing pipeline consists of 3 major components, (1) build the adjacency graph that represents spatial relationships within a building (between hallways, ofces, stairways, elevators, etc), (2) identify elements involved in evacuation routes (hallways, stairways), (3) construct the 3D building graph by connecting the oor elements via stairways and elevators. Our methods have been applied a number of buildings on the campus of UNC Charlotte. To date, 21 academic buildings have been processed using our tools. We demonstrate and results and visual analysis on using these tools on two of the processed buildings and compare these to traditional manual processing. We also demonstrate some preliminary work on integrating an evacuation model into our system, as well as extensions of our system to a mobile platform.

1 Introduction

Effective management of natural and man-made disasters are becoming increasingly important in order to prevent or minimize loss of life and property. Emergency management, as currently defined consists of four phases: mitigation, preparedness, response and recovery. Of these mitigation and preparedness refer to activities prior to an emergency, while response and recovery are activities during or after the event. The use of Geographic Information Systems(GIS) is currently the technological tool of choice in managing large-scale disasters[10]. GIS systems centrally contain spatial and other attribute data that are relevant to an event and can be used by emergency responders for timely decision making. The extent to which these systems are used depends on a large number of factors, and most importantly, on available resources at the county level.

Traditional GIS based decision support systems have significant limitations[24]; spatial data is in 3D, while most existing systems do not have full support for 3D geometry structures, attributes and textures, and navigation capabilities. Ideally, the rich capabilities of graphics and visualization systems is yet to be translated into today's GIS systems. A very recent work by Lee et al.[13] on using triangle structures based on voronoi diagrams exemplifies this need. Here the authors propose this structure as a basis for efficient spatial queries, as well as a means to explore what-if style scenarios, in terms of locating infrastructure and other resources for emergency management. A good discussion of significant impediments to using GIS is presented by Zerger et. al.[22] for emergency management, including limitations in spatial data accuracy and quality, limited numerical and real-time modeling capabilities.

In this work, our focus is on emergency management issues and constraints that relate to interiors of commercial buildings. Earlier work on this has focused on the larger problem of such environments to the rest of the transportation network[12, 14]. Buildings, also classified as micro-spatial environments, present some unique challenges in emergency management, including (1) evacuation of large buildings via limited paths through stairways/elevators, and (2) communication such as GPS is not reliable, presenting challenges to emergency responders. Additionally, emergency responders need to have a good understanding of the building geometry for fast and effective response. Given that there are hundreds of thousands of buildings, there is a need to build new infrastructure that can provide emergency responders with new tools for effective communication and response.

We present new tools to automatically process building data (from their original 2D CAD files) into a 3D graph representation. There are 4 steps to this process:

1. **Data Acquisition/Preprocessing.** This step starts with the raw CAD files, georeferences the data to a base map, and does some cleansing to ensure data accuracy and consistency.

2. **Adjacency Graph.** In this step, spatial adjacency relationships are derived from the raw polygon data in the CAD files, resulting in a 2D graph for each floor of the building. Hallways, stairways elevators and entrances may need to be manually identified, depending on whether the input file contains this information.

3. **Centerline Extraction.** A distance transform based algorithm is used to automatically extract the centerline (medial axis) of the hallway polygons, that typically make up the evacuation routes of the building. Some postprocessing is necessary to complete this process.

4. **Building Graph Construction.** The adjacency graph and extracted centerline are used to determine the rooms adjacent to the centerline and build a 2D network (also a graph representation) for each floor. These 2D networks are then combined with the stairways, elevators and building entrances, resulting in the 3D building network.

We also present initial work on extending our system to a *mobile*

*email:ribarsky@uncc.edu
†email:krs@uncc.edu
‡e-mail: jliu1@uncc.edu
§email:kclyons@uncc.edu
¶e-mail:krs@uncc.edu

platform, and related analytical tools. Finally, we report work on integrating an evacuation model into our framework.

We have used this methodology to process 22 academic buildings at UNC Charlotte[1]. We demonstrate both visualization and analysis tools using our 3D building network on two of these buildings, containing three and four floors respectively. We present results and the advantages of our new automation tools in reducing the time needed to build a 3D building network, by comparison to traditional manual processing. It is to be noted that current GIS systems such as ArcGIS lack the full interactive capabilities to maintain and represent 3D building networks, and at best provide limited 3D viewing tools.

2 METHODS

2.1 3D Building Network Construction

Fig. 1 shows the framework for constructing the 3D building network, starting from the building CAD files[2]. After some preprocessing, the adjacency relationships between different polygon elements in each floor is determined and represented as a 2D graph. Elements relevant to determining evacuation routes such as hallways, stairways, elevators and entrances are interactively identified. Next, centerlines of hallway polygons are extracted. Using the adjacency graph, all rooms adjacent to the hallway are connected to the nearest centerline point, resulting in a 2D network for each floor. Finally, the 2D networks from each floor are linked by the stairways, elevators and the building entrances, resulting in a 3D building network. All processed data is maintained in a PostgreSQL/PostGIS database[2, 1].

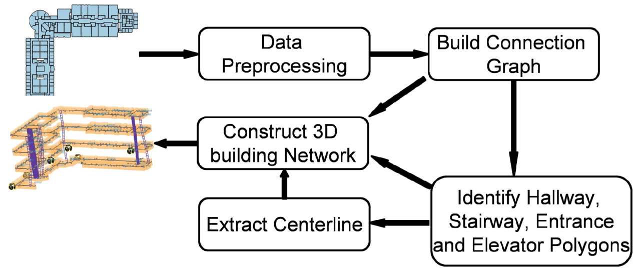

Data Preprocessing
Build Connection Graph
Construct 3D building Network
Identify Hallway, Stairway, Entrance and Elevator Polygons
Extract Centerline

Figure 1: The process of constructing 3D building network.

2.1.1 *Data Acquisition and Preprocessing*

Data acquisition begins with the CAD building files that will ultimately be incorporated into our PostGIS database. These drawing files are are read by ArcGIS(ESRI) software. Since these files contain data in 2D (each floor is on a separate file), and miss information on the connections between the floors(staircase, elevators), some manual processing is needed to prepare the data for 3D processing and graph construction. Additionally, elimination of data errors and consistency checking is necessary to avoid problems down the line.

Since the CAD files contain no spatial reference information, they must first be *geo-referenced* to a base map so as to line up with other datasets. We use the *Spatial Adjustment* tools in ArcGIS to define control points on the base map and point them to the same areas on the CAD building file. These are usually corners of the building or other defining features that will allow the algorithm to line up the data correctly with minimal distortion. Using a building footprint file or a rectified orthophoto allows a georeference to a

[1]We expect to complete the entire campus by December 2009.

[2]This work will be presented at the SPIE meeting in early 2010[11] and in a journal[15]

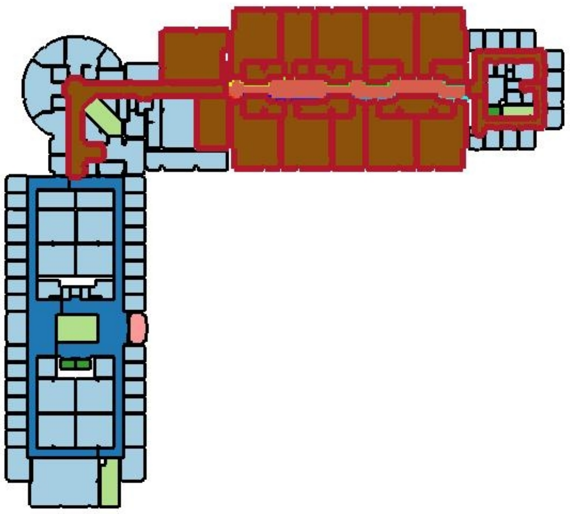


Figure 2: Adjacency Graph construction. A floor of a building is illustrated with a hallway polygon selected (light red). The automatically selected adjacent rooms and hallway polygon are illustrated (dark orange polygons). Building stairways are shown in green, and entrance to the building is in light red(next to the stairway, middle left)

coordinate system (in our case, NAD1983) and "spatially enable our data.

Once georeferenced, ArcGIS is used to read the files in their native format and convert to Shapefile format. Four shapefiles are produced for each CAD file, based on geometry type: point, line, multipatch, and polygon. Currently we only use the polygon data, which contain rooms, stairways and elevators. Next, all unneeded data are removed, and the files are "cleansed. A tag in the attribute table of the shapefile identifies the room, stairway, and elevator polygons. These are labeled as RM$, and a simple SQL query is used to select and insert them into their own separate file.

The "cleaned files are uploaded into a central PostgreSQL with PostGIS database. This database serves as the central data server for the application, and allows updates to be propagated down the line to any device reading from it. We use the database as the input source for the 3D building network construction, visualization, and other analysis tasks of the building.

2.1.2 *Adjacency Graph Construction*

Construction of the 3D building network requires knowledge of the spatial relationships between the elements that make up each floor and the links between the floors. However, the input database contains simply a collection of polygons that represent these elements at each floor. Thus we begin by analyzing the spatial relationships between the polygonal elements. For the sake of spatial analysis, we consider all polygonal elements to be rooms, except for hallways, stairways, and elevators. The goal is to recover and represent the relationship between these elements in an *adjacency graph*, where nodes represent polygonal elements and edges connect neighboring polygons.

Once the input polygonal data is read from the PostGIS database, each polygon's vertex ordering is checked, so as to be in counter-clockwise order, ensuring consistent orientation. Duplicate vertices, if any, are removed. Then the polygons (which may be concave) are converted into triangles using the Delaunay triangulation[3], to simplify geometry analysis. An interactive tool is also provided to specify polygons, if the triangulation fails.

Let $\mathbf{T^i_k}(\mathbf{v^k_1}, \mathbf{v^k_2}, \mathbf{v^k_3})$ define triangle k within polygon i, with vertices $v_n^k, n \in (1,3)$. Similarly, a polygon is represented as $\mathbf{P_k}(\mathbf{v_1^k}, \mathbf{v_2^k}, \ldots, \mathbf{v_n^k})$. Suppose the current polygon $\mathbf{P_i}(\mathbf{v_1^i}, \mathbf{v_2^i}, \ldots, \mathbf{v_n^i})$ is being processed to determine its spatial relationship with polygon $\mathbf{P_j}(\mathbf{v_1^j}, \mathbf{v_2^j}, \ldots, \mathbf{v_m^j})$. There are two steps to constructing the adjacency graph.

- **Triangle-Polygon Relationship.** In the first stage, each boundary vertex $\mathbf{v_k^j}$ of polygon j is evaluated as inside or outside of triangle $\mathbf{T_k^i}(\mathbf{v_1^k}, \mathbf{v_2^k}, \mathbf{v_3^k})$ in polygon i. For computational efficiency, we circumscribe each triangle, with $\mathbf{o_k^i}$ as the center and r_k^i as the radius. A boundary point is inside a triangle if

$$\sqrt{\|\mathbf{o_k^i} - \mathbf{v_k^j}\|^2} < r_k^i + \alpha \tag{1}$$

  where α is a constant value.

  If all boundary points of polygon j fail Eq. 1, it can be safely removed. Otherwise, it could be adjacent to polygon i and is retained.

- **External Triangle-Triangle Relationship.** In the second stage, the triangles of polygon j are analyzed for adjacency. Following the same strategy, circumcircles are constructed for each triangle of polygon j. Adjacency relationship is similarly defined between two triangles $\mathbf{T_k^i}$ and $\mathbf{T_k^j}$ as,

$$\sqrt{\|\mathbf{o_k^i} - \mathbf{o_k^j}\|^2} < r_k^i + r_k^j + \beta \tag{2}$$

  where β is a constant value. Based on Eq. 2, the accuracy of neighborhood relation between two polygons can be further improved by reducing β, resulting in identifying additional pairs of adjacent triangle pairs.

Fig. 2 illustrates an example output from our method. Here we see a hallway polygon (in light red) that has been selected. The neighboring polygons (in deep orange) that were identified are displayed, including an adjacent hallway polygon.

2.1.3 *Identification of Hallways, Stairways, Elevators, Entrances*

After the adjacency graph is constructed, we next identify polygons that represent hallways, stairways, and elevators. This is needed prior to centerline extraction, as well as for building network construction. Hallway polygons are typically adjacent to a large number of rooms(for example, corridors leading to offices in a commercial building); thus, its corresponding node has more edges(high degree) than other nodes and can easily be segmented from the rest of the graph. In addition, we provide interaction tools to let a user modify the results, if needed. Stairway and elevator polygons are similar to room polygons. If these are part of a building CAD file, they can be automatically identified and marked appropriately; otherwise, our system will let the user specify these types of polygons. Fig. 3 illustrates the process. In 3a, automatic process identifies most of the hallway polygons. In 3b, stairways, elevators and entrances are interactively specified, as this information is missing from the input CAD files. elements

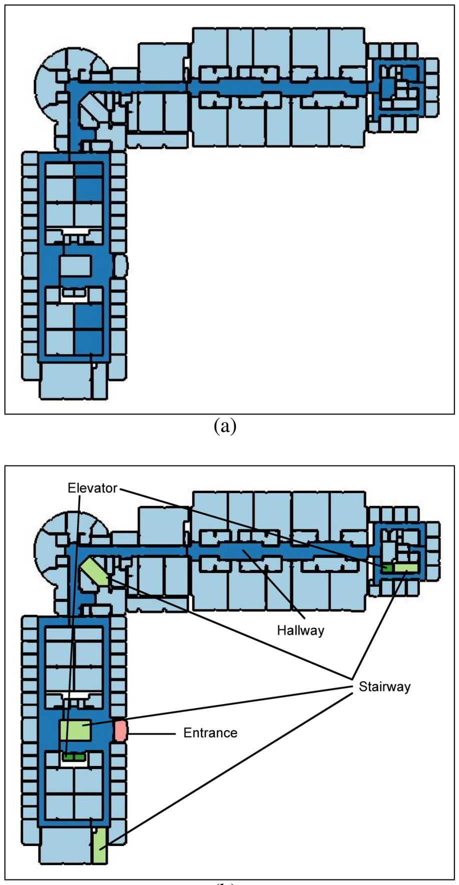


Figure 3: Identifying hallways, stairways, entrances, (a) Hallway polygons in dark blue have automatically been identified, (b) stairways (light green), elevator (dark green) and entrance (light red) were interactively identified, since this information is not part of the input CAD files.

2.1.4 *Centerline Extraction*

A key application of emergency management within buildings is the determination of evacuation routes. These typically involve hallways, stairways, elevators and entrances/exits. Automatic construction of evacuation routes is facilitated by centerline extraction algorithms (also proposed in [12]). Relevant methods for centerline extraction include those based on Voronoi diagram, potential field and distance field.

Voronoi Diagram. The Voronoi diagram represents a subdivision of space into regions whose points are closer to a generating vertex than any other element. Centerline algorithms using the Voronoi diagram[6, 17, 14] begin by sampling the polygon boundary and constructing the Voronoi diagram. The intersections between the Voronoi edges converge to the polygon centerline, as the boundary sampling rate is increased. One problem with this method is difficulty in joining centerline segments from separate but adjacent *hallway* polygons.

Potential Field. Potential field based methods begin by uniformly subdividing the space underlying the polygon. The pixels that intersect with the polygon edges are the sources of potential forces. First proposed by Chuang et al.[7], these methods evaluate the forces at each pixel, using an integration of the incoming forces from all visible pixels. Critical points, where the force sums to zero(or a minimum) are located and tracked along the force direction. These points are connected to make up the centerline[9, 8]. Our experiments with this method displays instabilities, due to difficulties in visibility determination, especially among the more complex concave *hallway* polygons.

Distance Transform. Similar to the potential field method, these approaches use a *distance function*, a signed function from certain source points. Points with maximum distance(local maxima) are extracted and organized into the centerline by using shortest path[23, 4, 5] or minimum spanning tree [21] algorithms.

We chose to use the distance transform based method, for the

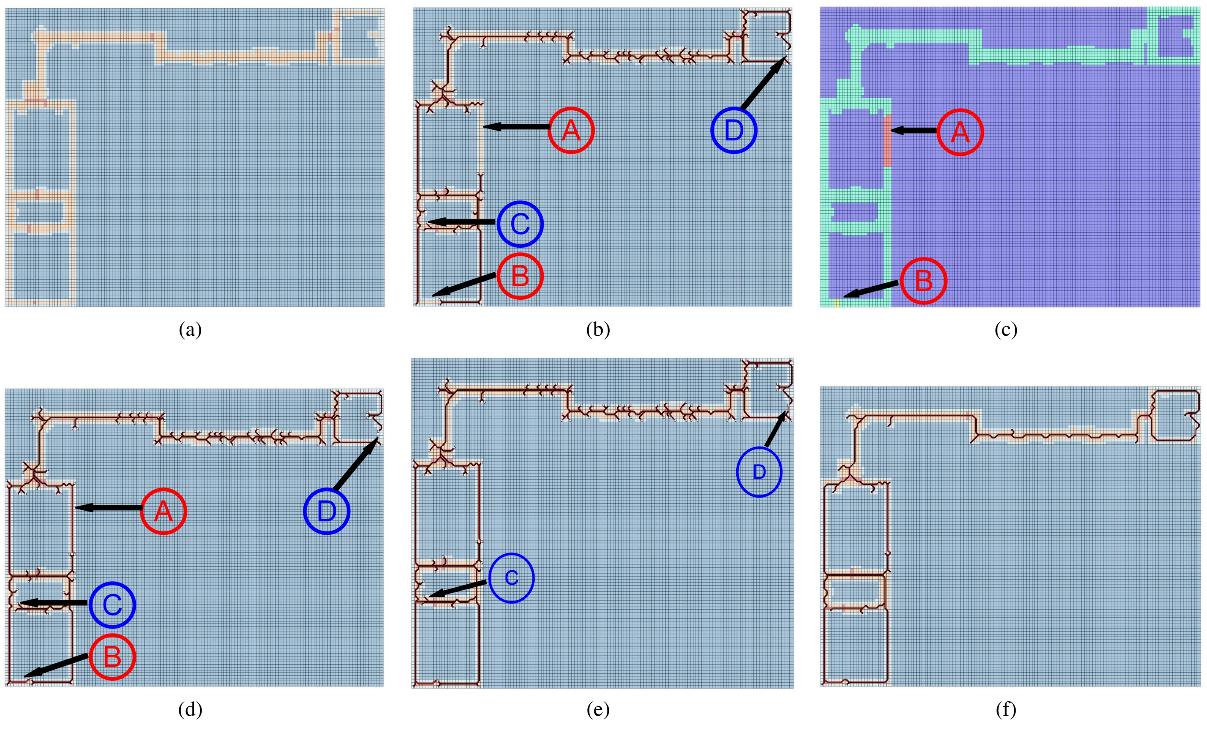


Figure 4: The process of centerline extraction. (a) Construct grids covering hallway polygons, (b) Centerline extraction by using minimum-spanning tree; (c) Identify missing segments through "circle-rolling" method, (d) Recover centerline segments in uncovered regions, (e) Manually repair small gaps, (f) Fulling centerline after pruning spurious branches. Red circles (A) and (B) denote large missing segments which can be detected by "circle-rolling" method, and blue circles (C) and (D) represents small gaps, which can be interactively repaired.

following reasons, (1) outside of the distance field calculation, the centerline extraction algorithm is quite efficient, (2) the centerline is guaranteed to be inside the structure, (3) it avoids visibility determination after pixelization and segmentation of object and background pixels, and (4) Separate but adjacent polygons can easily be joined by detecting shared boundary points. Our approach is illustrated in Fig. 4.

1. **Grid construction.** The space containing the polygons is first pixelized by uniform subdivision. Boundary pixels are identified (Fig. 4(a)) as the intersection between grid and the polygon edges. Seed points are selected inside the polygons and a region growing algorithm is used to identify the object pixels (yellow). the boundary pixels are changed into object pixels (red) if they belong to both (adjacent) polygons. Thus, the adjacent hallway segments can easily be merged.

2. **Centerline-Tree Construction.** Distance transform is performed on object pixels by considering the boundary pixels as source points. Then, the minimum-spanning tree can be built by using the inverse distance value as weight, and Wan's[21] algorithm is used to output the centerline-tree.

3. **Recovering Missing Centerline Segments.** Parts of the centerline segments maybe disconnected (see the marked circles in Fig. 4(b)) in the centerline-tree because the hallway is a loop, while the centerline structure is a tree. These gaps need to be identified and reconnected. To do this automatically, we use a *rolling-circle* scheme. A circle is centered at a centerline point $\mathbf{c}$ and radius equal to $r = 1.414 \times DFB(\mathbf{c})$ where $DFB(\mathbf{c})$ is the distance value of the current point. This circle is rolled along the centerline and its radius is dynamically varied, based on its current location. The regions not covered by the rolling circle are potentially broken segments (Circles A,B in Fig. 4(c)).

   The missing segments are repaired by determining two centerline points on either side of the uncovered region. Ideally, a shortest-path connecting these two lines will be the optimal route. However, it is likely to traverse the same path extracted using Wan's[21] algorithm. We propose two metrics to fix this:

   - Rather than directly extract missing centerline segments from the two selected points, we identify a point in the uncovered region with a local maximum in distance value, as well as being the farthest from one of the selected points. Then Wan's algorithm is used to connect these two points, followed by similarly recovering the gap with the second selected point.
   - The weight of minimum-spanning tree is modified. We add a penalty value to ensure that the selected path is always the shortest, by computing the distance from the seed point (DFS) to build the minimum-spanning tree. Thus, the weight is defined as

   $$weight = 1/DFB + DFS \tag{3}$$

   Fig. 4(d) shows the recovery of two centerline segments(circles A, B).

4. **Interactive Centerline Repair.** Small gaps (circles C, D) cannot be detected through the circle-rolling method. At present, we provide an interactive tool to let a user manually fill these gaps. It is efficient since these are small gaps and the user is only required to click two points; our system automatically searches for the two extracted centerline points closest

to the user's choice and the centerline gap is filled. Fig. 4(e) shows the repaired results, where two short centerline segments fill the gaps in circle C and D.

5. **Spurious branch removal.** In this step, a threshold is used to eliminate spurious branches associated with centerline extraction. The threshold is based on the point count of the branch segments. However, this process will only affect the centerline branches originally extracted (not from steps 3 and step 4) as well as having no offset branches. Fig. 4(f) illustrates the final centerline by pruning spurious branches.

2.1.5 *Network Construction*

To construct the 3D building network, we first build the 2D networks for each floor. This is accomplished by using the adjacency graph and the extracted centerline points. Rooms connected to hallway polygons are determined from the adjacency graph. Their centroids are then connected (they become graph edges) to the nearest centerline point.

After the 2D networks are constructed for each floor, the 3D network is constructed by linking corresponding stairways and elevators in different floors(represented by graph edges). These are then linked to the building entrances, enabling analytic tools to compute evacuation routes between any two nodes in the network.

Fig. 5 illustrates two building networks resulting from all of the automated and interactive processing. The blue points represent the centerline, while the red segments are connections(graph edges) to the adjoining rooms. Red segments between floors represent stairways and elevators.

3 Results

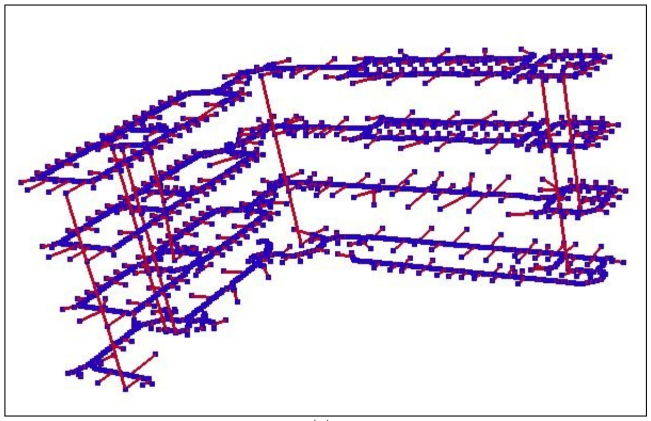
We have implemented our system in C++. We use the open source tools, including PostgreSQL[2] and PostGIS[1] as the main database server. ArcGIS was used for initial preprocessing. The desktop application uses the Visualization Toolkit[19] and FLTK[20] for the graphical interface. Future work on the mobile application will use more low-level tools, to scale better with the limited resources available on hand-held devices.

3.1 Visualization Tools

We provide interactive visualization tools to aid the user for both navigation, communication and some analytical tasks. We support a level of detail(LOD) representation of the urban environment. At the coarsest level, the landscape of the region of interest is represented, as seen in Fig. 3.16(a) of a collection of buildings. Buildings are represented as simple polygons at this level. If the user selects a particular building, the system will switch to the next level, which displays the selected building, as seen in Fig. 3.16(b). At this level, important features such as the entrances, stairways, elevators, etc, are highlighted. If the user continues to zoom into a particular part of the building, the system will transition to the next level, displaying the hallways in the interior of the building (Fig. 3.16(c)). We use transparency to illustrate the critical building structures. Finally, further zooming will let a user to change to the room-level view (Fig. 3.16(d)). Detailed information about the hallway, contained rooms can be displayed at this level.

3.2 3D Evacuation Model

In collaboration with Dr. Bala Ram, Professor, NC A&T University we are currently working on incorporating an evacuation model into our system. The evacuation model is a refinement of the model developed by Lu et. al.[16]. It also incorporates some earlier work on a multi-commodity network flow optimization model relevant to evacuation[18]. Some initial visualizations of an example simulation on two floors of an academic building is illustrated in Fig. 7. In this example, there are 374 occupants on the first floor of the building. Walking speed is assumed to be about 4.35 feet per sec(3 miles per hour), and occupancy area is about 3 square feet. In Fig. 7 the occupants are indicated by green filled spheres and moving towards two different exits.


(a)

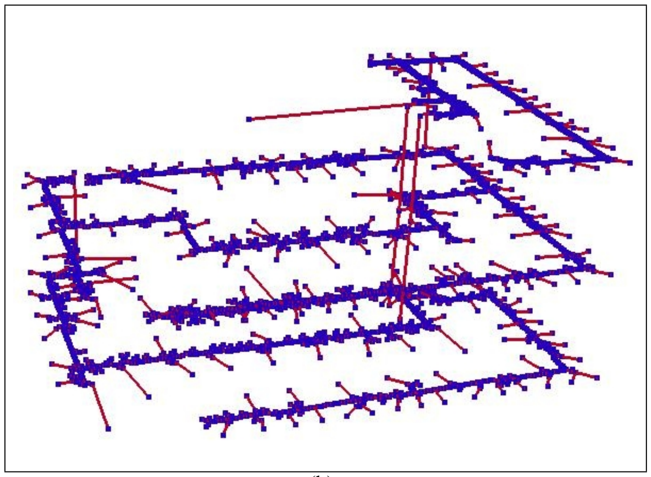

(b)

Figure 5: Example building networks of 2 buildings. Blue line segments represent the hallways, while the red segments represent connections to the rooms (offices) as well as the connections between the floors via stairways and elevators

3.3 Extension to Mobile Devices

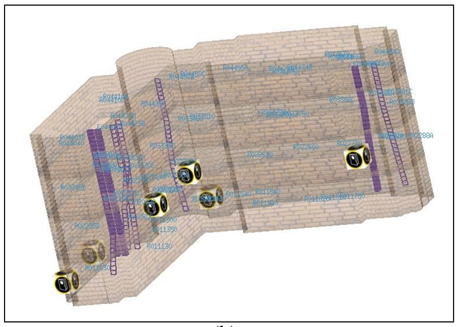
We have begun buiding new capabilities on a mobile application that will be used for onsite routing, remote data gathering, and to communicate between emergency responders. Our mobile application can communicate with the central PostgreSQL server(via PHP scripts), and display individual floors of building geometry. Selection of rooms within the floor has also been completed. A successful demonstration of two iPods communicating with the PostgreSQL server, as illustrated in Fig. 8 was also accomplished during the project period. Currently, work on adding additional interaction and query capabilities is ongoing, as well as limited 3D display and navigation capabilities. We currently use Apple iPhone/iPod technologies.

4 Testing and Evaluation

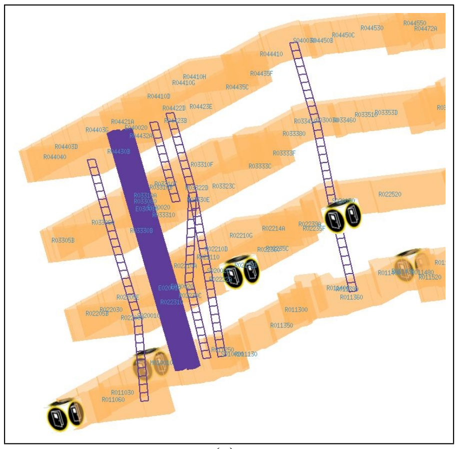
An important consideration of emergency management within buildings is the ability to quickly evacuate the occupants in an orderly and safe manner. Thus, automatic routing should be a critical component of any GIS assisted evacuation system. In our system,

Cameron
Poplar Terr
Cameron Bv
Phillips Rd
Craver Rd
Woodward Hall
Cameron Bv
Mary Alexander Rd
University Rd
Broadrick Bv
High Rise Rd
University City Bv
Sandburg Av
Vance Cr

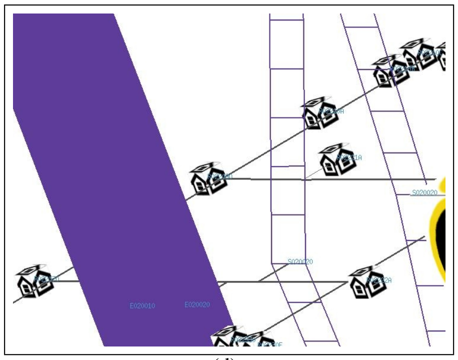
(a)



(b)



(c)



(d)

Figure 6: Level of Detail representation. (a) Coarsest level shows network of buildings, (b) building view, (c) Hallways within the building, and (d) Room view

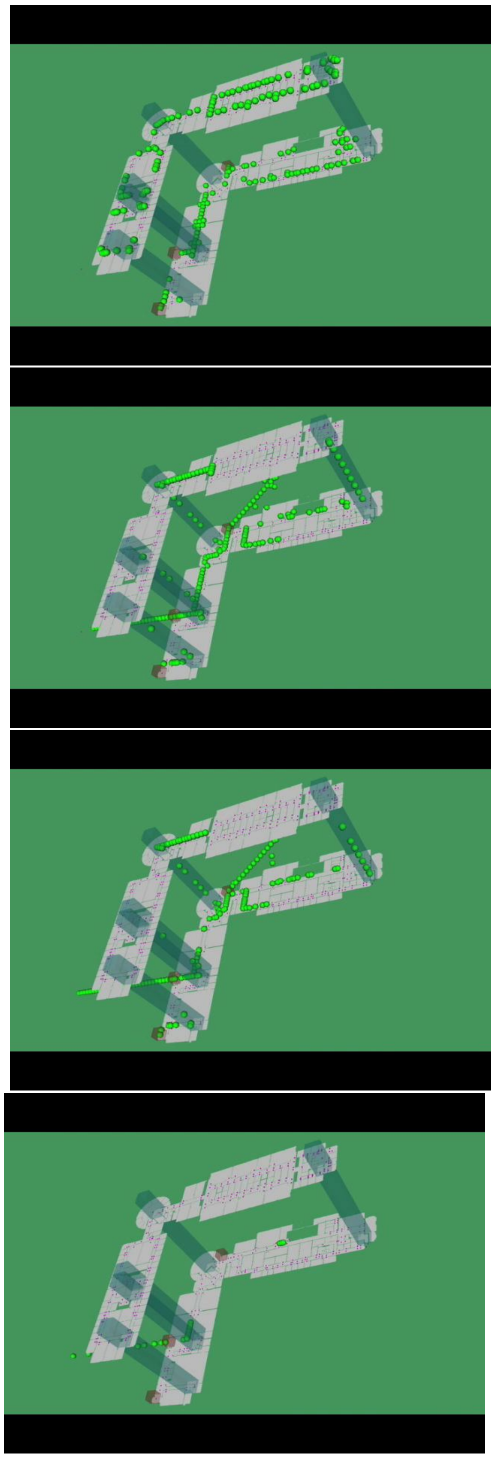


Figure 7: Evacuation Simulation Example on a Two Floor Model. Entrances and exits are on the lower floor.(Top Left) At the beginning of simulation, occupants are shown at the start positions, indicated by green spheres, (Top Right) Occupants can be seen exiting through four stairwells and two exits, (Bottom Left) Most of the top floor occupants have exited to the lower floor, (Bottom Right) Evacuation is almost complete.

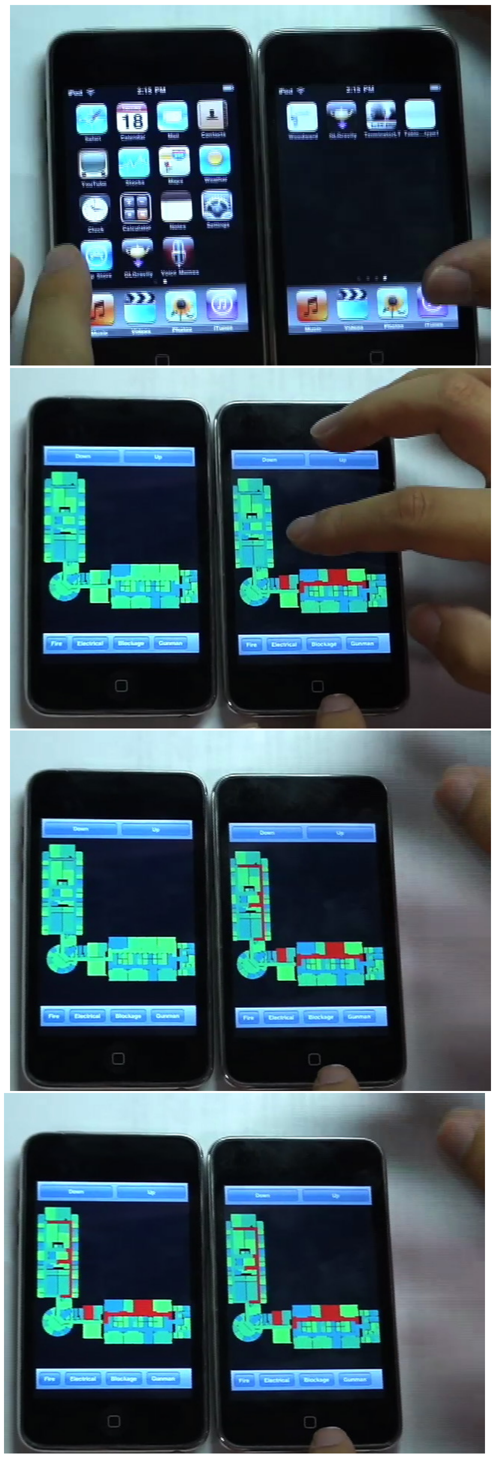


Figure 8: Mobile Application. (Upper Left) Two mobiles, with the application being started, (Upper Right, Lower Left) Shows one floor of Woodward Hall loaded and two hallways selected(in red) on one mobile, (Lower Right) Selections have been communicated to the server, and the second mobile is updated.

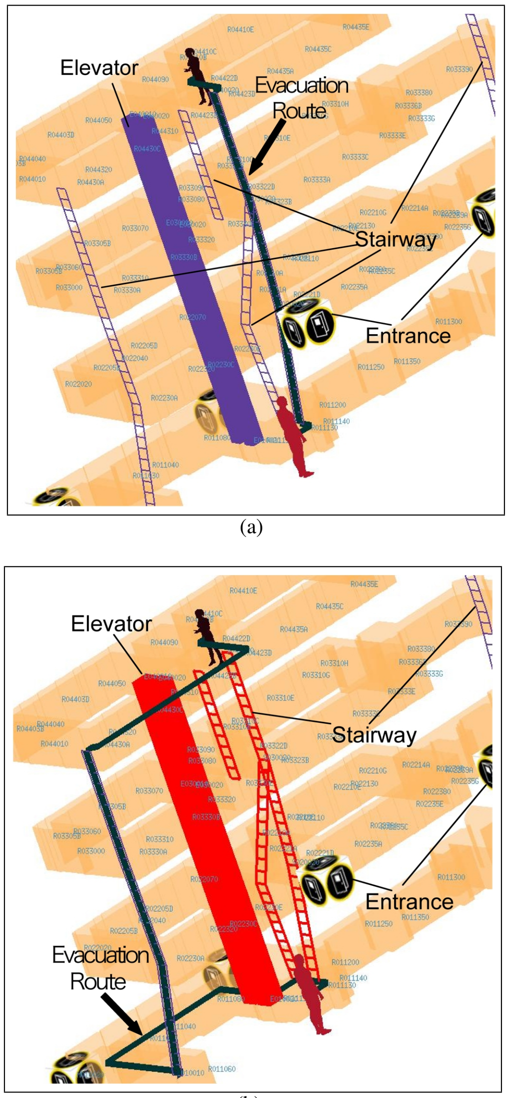


Figure 9: Test Scenario. An example situation where a part of the the buiding is blocked due to an emergency. (a) Evacuation rout between two points in the building under normal conditions, (b) Stairway and elevator has been blocked and an alternative evacuation route is displayed.

we support this capability to determine these routes under constraints. Fig. 4 shows an example scenario. In Fig. 4blocka we display the evacuation route constructed between two points(users) within a building. Dijsktra's shortest path algorithm is used to compute routes. In Fig. 4blockb shows a new path that was computed after the elevator and stairway was blocked (for instance, an emergency such as a chemical leak) from use. The stairway and elevator are now highlighted (in red) indicating the emergency.

In our system, the above situation can be controlled from multiple users. Thus, the responder on the fourth floor will evaluate the situation and block off the elevator and stairway. In our system, this is accomplished by interactively selecting the particular elements. This is then communicated to the database server and all other users will see the updated view immediately. Thus, Fig. 4blockb will be the view presented to the user at the entrance, or the commander on the ground, managing the emergency when certain hallways are blocked.

4.1 Performance

Tables 1 and 2 show a comparison of performance using our automation tools for 3D network construction. Processing time ranges from 10-20min using our tools. Majority of the processing time is due to interactive processing. Doing the entire processing manually takes anywhere from 2-4 hours. For more complex buildings (the buildings in our experiments contain 3 and 4 floors respectively) manual processing will be impractical.

Bldg. Name	Adj. Graph Constr.	Centerline Extraction	Bldg. n/w Constr	Manual Proc.
Cameron	4.0s	0.22s	7.1 s	540s
Woodward	5.82s	1.04s	0.34s	1080s

Table 1: Performance: Semi-Automatic Method

Building name	Centerline drawing	Network creation	Attributes assignment
Cameron	60 minutes	5 minutes	60 minutes
Woodward	90 minutes	5 minutes	120 minutes

Table 2: Time usage:Manual Processing

5 Conclusions

In Phase I of this project, we have constructed new automation tools to process building data from 2D CAD files, for use in emergency management of commercial buildings. We support a full-scale 3D representation, and more specifically a geo-referenced 3D building network that is structured for spatial querying and visualization. We support an LOD representation that permits large complex buildings to be viewed at different scales. This will become important when we adopt this application to work on hand-held mobile devices. We also demonstrate a test scenario for updating evacuation routes, depending on the situation, in an asynchronous manner. The design of the system is based on an intuitive interactive interface. Preliminary work on integrating an evacuation model and extension to a mobile platform has been presented.

This work reveals aspects of and an approach to handling a common problem in visual analytics: how to manage data for an application or domain and provide semantics and meaning in such a way that the data can be effectively used in an interactive visualization approach. A key issue for real-world applications is making sure that the data management and analysis can be carried out by the people and entities that will use them. For police and fire departments, it is very important that these processes be mostly automated and apply to data formats that they have access to. Otherwise too many person-hours or too expensive specialized tools will be needed, and the visual analytics approach will never be used at scale. In our approach, we strive to automate as much as possible, keeping in mind that the problem cannot be completely automated, so it is important to have interactive tools to finish the job quickly. In addition, standard CAD formats and GIS approaches are used, which have the added advantage that other GIS data can be merged with the building data and the whole system can be contributed to and perhaps even maintained by a city GIS and planning department. Finally, the data management and analysis issues addressed here also elucidate issues of interest in the FODAVA (Foundations of Data and Visual Analytics) efforts.

6 Acknowledgements

This work is supported by a grant from the National Institute of Justice under the auspices of the Department of Justice.

References

[1] Postgis. Refractions Research, http://postgis.refractions.net.

[2] Postgresql. http://www.postgresql.org.

[3] M. Berg, O. Cheong, M. Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, third edition, 2008.

[4] I. Bitter, A. Kaufman, and M. Sato. Penalized-distance volumetri skeleton algorithm. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):195–206, 2001.

[5] I. Bitter, M. Sato, , M. Bender, K.McDonnel, A. Kaufman, and M. Wan. Ceaser: A smooth, accurate and robust centerline-extraction algorithm. In *Proceedings of IEEE Visualization 2000*, pages 45–52, Oct. 2000.

[6] J. Brandt and V. Alazi. Continuous skeleton computation by voronoi diagram. *CVGIP: Image Understanding*, 55:329–338, 1992.

[7] J. Chuang, C. Tsai, and M. Kuo. Skeletonization of three-dimensional object using generalized potential field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1241–1251, 2000.

[8] N. Cornea, D. Silver, and P. Min. Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):530–548, 2007.

[9] N. Cornea, D. Silver, X. Yuan, and R. Balasubramanian. Computing hierarchical curve-skeletons of 3d objects. *Visual Computer*, 21(11):945–955, 2005.

[10] A. E. Gunes and J. Kovel. Using gis in emergency management operations. *Journal of Urban Planning and Development*, 126(3):136–149, Sept. 2000.

[11] J.Liu, K.Lyons, K. Subramanian, and W. Ribarsky. Semi-automated processing and routing within indoor structures for emergency response applications. In *Proceedings of SPIE Conference on Defense, Security, and Sensing, April 2010**(To Appear)***, Apr. 2010.

[12] M.-P. Kwan and J. Lee. Emergency response after 9/11: the potential of real-time 3d gis for quick emergency response in micro-spatial environments. *Computers, Environment and Urban Systems*, 29(2):93–113, 2005.

[13] I. Lee and K. Lee. A generic triangle-based data structure of the complete set of higher order voronoi diagrams for emergency management. *Computers, Environment and Urban Systems*, 33(2):90–99, 2009.

[14] J. Lee. A spatial access-oriented implementation of a 3-d gis topological data model for urban entities. *GeoInformatica*, 8(3):237–264, 2004.

[15] J. Liu, O. Obirieze, K. Subramanian, and W. Ribarsky. A 3d interactive system for effective response and communication within large urban structures. *National Crimemapping Journal*, 2010. In Preparation.

[16] Q. Lu, Y. Huang, and S. Shekhar. Evacuation planning: A capacity constrained routing approach. In *Proceeding of the First NSF/NIJ Symposium on Intelligence and Security Informatics (ISI 2003)*, 2003.

[17] R. Ogniewicz and M. Ilg. Voronoi skeletons: theory and applications. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 63–69, 1992.

[18] S. Osman, B. Ram, P. Standfield, F. Samanlioglu, L. Davis, and J. Bhadury. Optimization model for distributed routing for disaster

area logistics. In *The Fifth IEEE International Conference on Service Operations, Logistics, and Informatics*, pages 278–283, July 2009.

[19] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Prentice Hall Inc., 4th edition, 2006. www.vtk.org.

[20] B. Spitzak. The fast light toolkit. http://www.fltk.org.

[21] M. Wan, Z. Liang, I. Bitter, and A. Kaufman. Automatic centerline extraction for virtual colonoscopy. *IEEE Transactions on Medical Imaging*, 21(12):1450–1460, 2002.

[22] A. Zerger and D. I. Smith. Impediments to using gis for real-time disaster decision support. *Computers, Environment and Urban Systems*, 27(2):123–141, 2003.

[23] Y. Zhou and A. W. Toga. Efficient skeletonization of volumetric objects. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):196–209, 1999.

[24] S. Zlatanaova, A. Rahman, and M. Pilouk. Trends in 3d gis development. *Journal of Geospatial Engineering*, 4(2):1–10, 2002.