Visual Analysis of Situationally Aware Building Evacuations

Jack Guest and Todd Eaglin and Kalpathi Subramanian and William Ribarsky

Charlotte Visualization Center, Department of Computer Science
The University of North Carolina at Charlotte, Charlotte, NC 28269, USA

ABSTRACT

Rapid evacuation of large urban structures (campus buildings, arenas, stadiums, etc.) is a complex operation and of prime interest to emergency responders and planners. Although there is a considerable body of work in evacuation algorithms and methods, most of these are impractical to use in real-world scenarios (non real-time, for instance) or have difficulty handling scenarios with dynamically changing conditions. Our goal in this work is towards developing computer visualizations and real-time visual analytic tools for building evacuations, in order to provide situational awareness and decision support to first responders and emergency planners. We have augmented traditional evacuation algorithms in the following important ways, (1) facilitate real-time complex user interaction with first responder teams, as information is received during an emergency, (2) visual reporting tools for spatial occupancy, temporal cues, and procedural recommendations are provided automatically and at adjustable levels, and (3) multi-scale building models, heuristic evacuation models, and unique graph manipulation techniques for producing near real-time situational awareness. We describe our system, methods and their application using campus buildings as an example. We also report the results of evaluating our system in collaboration with our campus police and safety personnel, via a table-top exercise consisting of 3 different scenarios, and their resulting assessment of the system.

Keywords: Evacuation, visual analysis, situational awareness, emergency response

1. INTRODUCTION

In any emergency or incident involving large urban structures (arenas, stadiums, college campuses), the safety of the occupants is of paramount importance. Thus every building has a set of passive safety features (sprinkler systems, fire extinguishers) and evacuation plans or routing maps posted at various points within the building. However, in recent years, more active approaches to studying evacuations from large urban structures or street networks have gained importance, and have been based on mathematical and algorithmic approaches. These methods study the problem of evacuating the occupants from the structure in the shortest possible time, also known as the egress time. What is lacking in these methods is the ability to handle real-world scenarios involving *large and complex structures that may involve multiple buildings*, and more important, the *ability to react in real-time to dynamic changes in the scenario*, such as blocked stairwells or hallways, and provide useful recommendations to responders who can mitigate damage, injury, or loss of life. For a commander overseeing the evacuation, the ability to clearly and unambiguously understand the dynamically changing situation is very important, being useful for optimal allocation of limited resources and personnel, and for making other timely decisions.

In this work, we address these challenges with the primary goal of responding to dynamic events in real-time during an emergency. We begin with an existing heuristic based route planning algorithm,[1] make certain modifications, and embed this as part of a visual analytic system that permits complex real-time interactions during the event. This then permits dynamic changes in the building accessibility to be incorporated, and alternative routing assessed in real-time. To accomplish this, we employ an LOD graph representation of the underlying urban structures that permits real-time recommendations to be presented to the emergency planners

---

Further author information: (Send correspondence to Kalpathi Subramanian)
Kalpathi Subramanian: E-mail: krs@uncc.edu, Telephone: 1 704 687 8579
Jack Guest: E-mail: jguest@uncc.edu
Todd Eaglin: E-mail: teaglin@uncc.edu
William Ribarsky: E-mail: ribarsky@uncc.edu

Visualization and Data Analysis 2013, edited by Pak Chung Wong, David L. Kao, Ming C. Hao, Chaomei Chen,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 8654, 86540G · © 2013 SPIE-IS&T
CCC code: 0277-786X/13/$18 · doi: 10.1117/12.2001535

and responders for informed decision-making. This is combined with realistic models for building occupancy and traffic flow. Interactive visual analytic tools permit quick exploration of possible impacts of the current situation, and the increased situational awareness for emergency commanders and responders permit more optimal use of scarce resources, for instance, dispatching responders to areas of need in congested sections of a building, handle casualties, etc.

We begin with a review of current work on evacuation algorithms, followed by a description of the techniques we use to create our 3D models and building networks. This is followed by a description of our modified capacity based route planner, followed by a detailed description of the major features of our interactive visual analytic system and its reporting and recommender functions. We describe the use of this system on typical scenarios using our campus buildings as a test case. We evaluated our system and its performance via a table-top exercise, consisting of 3 scenarios: gas leak in building, active shooter, and an explosion in an adjacent structure. Performed in collaboration with our campus police and safety personnel, we report on their assessment and recommendations.

2. PREVIOUS WORK

The study of computer based evacuation modeling has evolved with both mathematical and algorithmic approaches. These approaches are categorized as *macroscopic* and *microscopic*.

2.1 Macroscopic Models

Macroscopic approaches focus primarily on minimizing egress time. The evacuees are treated as a unit or group of units and moved from source to destination. Interaction between these units are defined by capacity/congestion rules. Linear programming methods based on Network Flow were one of the earliest approaches, yielding optimal solutions but at high algorithmic cost, and often impractical to use in real-world scenarios. The Maximum Flow Problem is one network solution that is implemented with costs as high as $O(n^3)$[1] yielding a best known cost of $O(nm \lg n^2/m)$. Naoyuki Kamiyama et al.[2] apply two initial conditions to the network flow problem. For each vertex the sum of transit times of arcs on any path takes the same value, and for each vertex the minimum cut is determined by the arcs incident to it whose tails are reachable. These assumptions resulted in a 2d grid network and they solved the transhipment problem in $O(n \lg n)$ time. Shekhar and Yoo[3] compare models relevant to the study of nearest neighbor paths. Also Kim, et.al[4] discuss contraflow in reconfigured networks for emergency route planning. This work gives insight into reconfiguring a network that is damaged and therefore relevant to our work.

Our model is based on a heuristic approach, the Capacity Constrained Route Planner algorithm proposed by Shekhar et al.[1] This approach attempts to find lower cost algorithmic solutions at the expense of the detail of each evacuee's egress. These approaches are interesting because they can be evaluated quickly from a user perspective as the network-flow problem is reduced to a generalized shortest path problem. The inputs are a graph structure and evacuee populations. The output is a route plan with start times, and a location matrix for each evacuee for each defined time segment. We have adapted this algorithm to meet the more challenging requirements for real-time decision making in large urban environments, as well as the ability to inject situational changes during an emergency.

2.2 Microscopic models

Microscopic approaches use agent based modeling, where each evacuee is governed by unique rules of behavior. Interaction between individuals and their environment are defined based on spatial and social parameters. These approaches do not exclude the deterministic path or goal that is the foundation of the route planner, but they rely on behavior rules applied to each evacuee to overcome the "lack of detail" inherent in network flow or heuristic planners. These methods are also referred to as Agent Based Models (ABM). The goal remains to minimize the time to evacuate individuals to safe zones.

Because of the adaptive parameterization of ABM, neural networks and fuzzy-logic[5] approaches can be adapted to building evacuation simulation. Discrete Particle Swarm Optimization,[6] use of velocity and spatially based rules of interaction[7] are other possible approaches, as detailed by Castle et al.[8] The most important aspect of ABMs are the rules applied to each evacuee. Castle et. al.[8] describe a detailed list of rules and attributes. We believe our system captures sufficient detail using a congestion model. However, because agents can be trained, we are evaluating Q-Learning/SARSA techniques to add human factors to future work.

2.3 Visual Analytics

Visual analytics involves effectively combining interactive visual displays with computational transformation, processing and filtering of large data.[9] One focus of visual analytics is real-world problems involving situationally-aware decision support. Andrienko et al.[10] focused on automatic generation of transportation schedules for evacuation from a disaster zone; visual analytic tools were used for verification by human experts. Campbell and Weaver[11] used interactive visualization tools for hospital evacuation scenarios that involved training first responders. The work of Kim et al.[12,13] focused on use of mobile devices for situationally aware emergency response and training, and thus their approach is similar to our work. They demonstrated their system with an evacuation simulation of the Rhode Island club fire of 2003. Our system is considerably more general and is scalable to large urban buildings and provides the means to interrupt the simulation based on new situational information or dynamic changes.

The use of linked views is an important technique to connect different representations of information within a single visualization, with applications specific to urban structures.[14,15] Sensor networks are used within buildings to help create interactive visual analytics. The work of Ivanov et. al[16] is not specific to building evacuations, however their use of data graphs to interact with maps provided inspiration for our cross platform bar chart displays that interact with our simulation. Visual analytics tools such as Jigsaw[17] are available for integrating process output data from an application.

3. METHODS

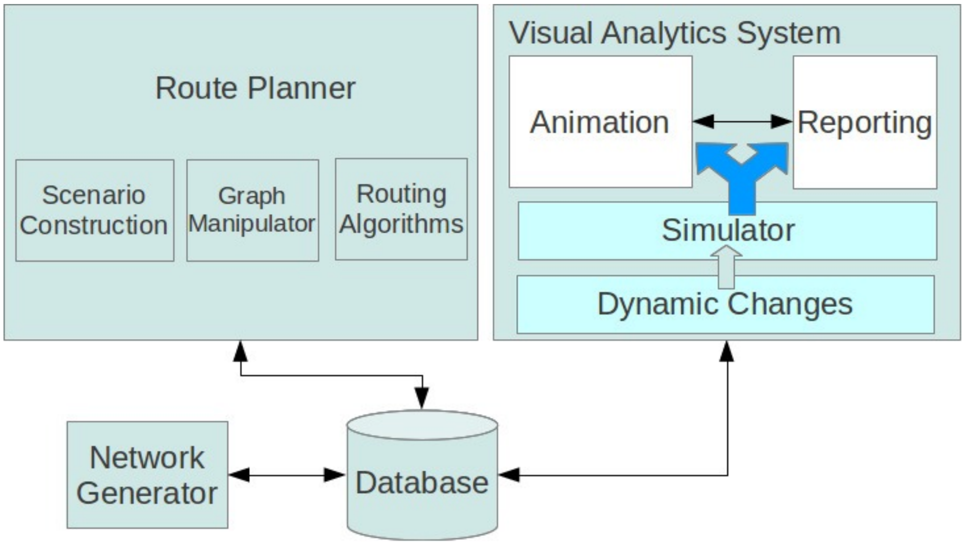


Figure 1. Visual Analytic System Architecture. Route planner involves preprocessing and routing calculations that serve to initialize the system, with processed results stored in a database. The visual analytic system uses visual abstraction and scalable representations that permit real-time interaction, injection of dynamic situational changes and visual analysis.

Fig. 1 illustrates the major components of our visual analytic system for situationally aware evacuations. There are two major components to the system. Functions in the Route Planner involve a significant amount of apriori processing and initialization. The visual analytic system is a highly interactive system that is user driven and can inject and respond to dynamic changes during evacuations. It also consists of reporting and visual analysis functions that can assist an emergency planner on exploring different scenarios, as to the use and deployment of resources, dispatch responders, effect of rerouting occupants, etc.

In our earlier work,[18] we described a semi-automatic system that constructed a *building graph*, incorporating key elements of a georeferenced urban structure critical to evacuations, such as hallways, stairways, elevators and entrances/exits. This building graph generator is used to process urban structures and is stored in a PostgreSQL database. We have successfully processed over 70 buildings of our campus using the graph generator.

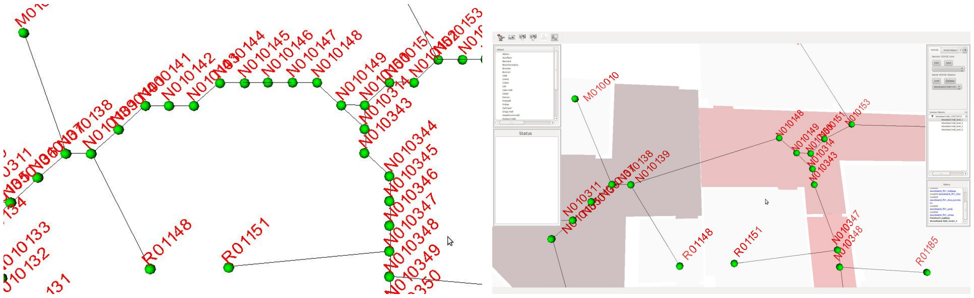
We next describe the main components of our visual analytic system.

3.1 Route Planner

Our route planner provides facilities for loading evacuation objects, that are combinations of urban structures, pathways, streets etc. Evacuations are built as combinations of structural objects and route planner objects and saved in the database. The Route Planner consists of the following components:

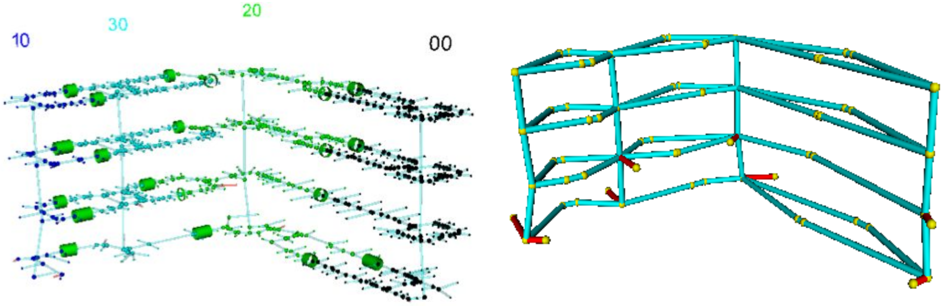
3.1.1 Scenario Construction

Scenario construction includes loading single or multi-building evacuation objects, selecting a route planning algorithm (currently limited to our modified capacity constrained route planner[1]), and setting visualization modes and evacuation parameters, such as building capacity, egress width, evacuee speed and density, and stairway up resistance.



(a) (b)

Figure 2. Building graph Simplification. Removing redundant nodes. (a) a section of a building floor with labeled building elements, (b) simplified graph.



(a) (b)

Figure 3. Building graph simplification to a zone graph. (a) building graph of a campus building, (b) transformation to zone graph representation. Yellow spheres are nodes and cyan tubes represent edges. The red tubes represent paths to exits

3.1.2 LOD Graph Construction/Manipulation

All computed egress paths are saved in the database as node to node connections. Nodes are defined based on their function, and can be rooms, hallways, stairways, elevators, and exits. The evacuation application also creates muster points, that represent locations where evacuees are ordered to congregate during an emergency. During the process of extracting centerline points the geometry of hallways is sampled at approximately two foot intervals. This process is necessary to insure accuracy, particularly at corridor elbows. However, this raises computational issues with extremely large buildings or multi-building graphs, and hinders real-time performance (expensive routing calculations) when dynamic changes need to be accommodated during an event. We address this problem by simplifying the graph with two different levels of detail.

Level 1. Remove Redundant Nodes. In this step, we simplify the original building graph by removing nodes that do not impact routing, for instance, shortest path calculations. The larger sampling rate used for accurate centerline calculations results in nodes that can be removed for use in building routes. Nodes and edges are collapsed in the process. Beginning with any node with three or more edges (or any arbitrary node), edges are followed until a node with 3 or more edges is encountered. This becomes a node of the simplified graph and a new edge is created connecting to the previous node, as can be seen in Fig. 2. The process is continued until all nodes have been visited. Our route planner is executed on these simplified graphs for scenarios in which all original egress paths are available. In our experiments, we see a factor of 5-8 reduction in the number of nodes in the simplified graph.

Level 2. Zone Graphs. For rapid computation of paths when dynamic changes are injected during an event, the simplified graphs can still be large, especially in multi-building evacuations. In these circumstances, we further simplify the building graphs into *zone graphs*, by segmenting the building into evacuation sensitive zones: for instance, stairwells, elevators and exits form the critical elements of any egress path. An example zone graph is illustrated in Fig. 3, involving stairwells, hallways and exits (in red).

To compute the zone graph, we use the precomputed evacuee paths to first associate each node with a zone that is closest to it, using an iterative procedure. In the second pass, the graph connectivity is established by keeping track of the zone of related objects that are encountered in these paths (adjacency lists are maintained). Intermediate nodes (the yellow spheres in Fig. 3(b)) are also identified by paths that cross multiple zones and are further used to complete the graph construction. Zone objects span floors in multi-storey structures, with appropriate floor identification for proper path determination during routing calculations. The number of nodes in the resulting zone graph depends on the number of zones and the number of floors in the building. In our experiments, a further factor of 5-7 reduction in the number of graph nodes was seen.

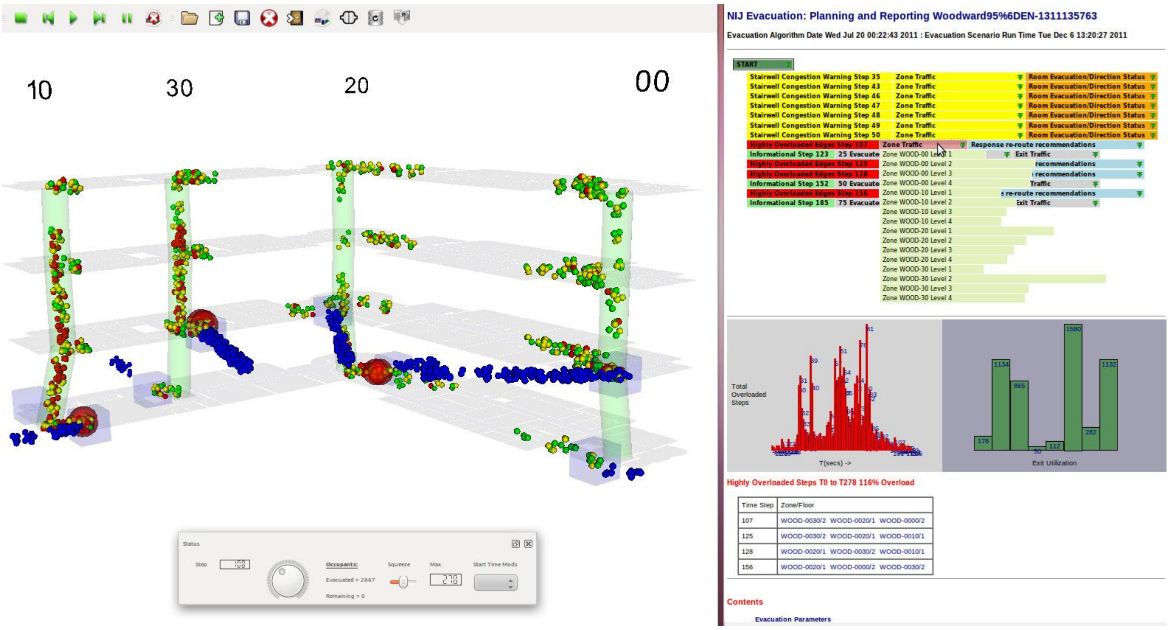
3.1.3 Routing Algorithms

We have implemented a modified version of the Capacity Constrained Route Planner (CCRP),[1] which is illustrated in Algorithm 1 (See Appendix A). Given a directed graph with node and edge capacities, the algorithm repeatedly computes the shortest path for each evacuee with available capacity. If a path is found, then it assigns as many evacuees as possible through that path, i.e., until the capacity of any node or edge along the path is exceeded. This is followed by moving all the evacuees at that time step. The process repeats until all evacuees have found paths to exit the structure. The final step is to evacuate the remaining evacuees in the building (who already have paths, but not exited the building).

We have augmented the CCRP algorithm by specifying the movement of the evacuees (in addition to finding the paths) at each iteration. Secondly, the algorithm is modified to work with our simplified graphs; in the simplified graphs, weights of the collapsed edges are accumulated and assigned to the new simplified edges. Running the routing algorithm on the simplified graphs makes it more scalable to larger urban structures as well as facilitating dynamic changes to the graph that will require rerouting occupants around blockages or other hazards caused by the emergency event. Finally, although each evacuee has a set average speed (3 ft. per sec.), evacuees cannot exceed the set density threshold. Thus, as congestion builds up, evacuee movements are naturally slowed down. Additional data structures are maintained to make these computations efficient.

3.2 Visual Analytics System

The visual analytic system (Fig. 1) consists of a simulator that accepts user input during an emergency, a 3D interactive animated display of the ongoing evacuation, and reporting and analytic modules. All these views accept direct input and the views are linked to update automatically, resulting in presenting the user with the most current information.



NIJ Evacuation: Planning and Reporting Woodward95%6DEN-1311135763

Evacuation Algorithm Date Wed Jul 20 00:22:43 2011 : Evacuation Scenario Run Time Tue Dec 6 13:20:27 2011

Highly Overloaded Steps T0 to T278 116% Overload

Time Step	Zone/Floor
107	WOOD-0030/2 WOOD-0020/1 WOOD-0000/2
125	WOOD-0030/2 WOOD-0020/1 WOOD-0010/1
128	WOOD-0020/1 WOOD-0030/2 WOOD-0010/1
156	WOOD-0020/1 WOOD-0000/2 WOOD-0030/2

Contents

Evacuation Parameters

Figure 4. Visualization Design. The upper left panel is the 3D view of the building undergoing an evacuation. Spheres encapsulate evacuee population densities, permitting easy identification of congestion points in the building. Vertical tubes (light green) represent stairways/elevators. Cubes on the first and second floor indicate exits. Lower left indicates the status bar for animation control. The upper right panel is for displaying reports of significant events, that can be drilled down. Lower right panels (bar graphs) show aggregate information on exit occupancy as well as events arranged on a timeline. The report views and the 3D views are linked for immediate updates.

3.2.1 Simulator

The simulator loads evacuation plans that were saved for scenarios under normal static conditions. Since dynamic changes affect only a small part of the structure, a *large amount of preprocessed data can be reused, contributing to our real-time performance*. The simulator accepts user defined dynamic changes (specification of blockages or casualty reports through the 3D display or the report modules) and modifies computed paths, and reroutes occupants. In addition it updates the generated reports and responder recommendations or action plans.

3.2.2 Visualization Design

Fig. 4 presents our interactive visualization system. Almost all of the interaction are via *direct manipulation*. There are three components that make up the design. On the left is a 3D animated view of the urban structure, where the user can load and play evacuation simulations. Evacuees are represented as spheres and colored green, yellow, or red based on low to high congestion. Partially transparent light green tubes represent stairwells, while purple cubes are exits. Blue polygons represent areas that can be occupied. Large red spheres of varying opacity represent edge (capacity) congestion. On the bottom left is the *status tool widget*, that allows moving around in the animation with a slider, indicating current step, evacuee counts and total simulation time. The top right panel is the *significant event* window. The rectangular bars are menus with varying levels of detail of the simulation report. The bottom right panel is a scrolling widget with interactive charts and graphs for interacting with the simulation and visual analysis.

- **Congestion Representation.** Congestion is the primary concern of each evacuee and predictions of future congestion and mitigation is of concern to the emergency response teams. In our system, congestion levels range from green to red (low to high).
- **Temporal Cues.** The color and size of spheres is modified at significant event times. For example sphere size is enlarged when the simulator starts moving evacuees from a source location. The resultant pulsing

in the animation yields important spatial and temporal information to the user about where a new source of traffic will originate and likely areas of future congestion.

- **Details on Demand.** The *Significant Event* window in the reporting tool uses a colorized layered menu so that the visualization of significant information is presented as needed by the user. This allows a maximum amount of reporting while allowing for quick event scanning in the event report. A user sees a limited top level distribution of data unless there is a reason to drill down deeper into the event for details. In the top right of Fig. 4 the user has selected a *Heavy Zone Congestion* item (in red) with its time step. The user can explore further via a mouseover operation to reveal a bar graph that shows the relative congestion of each zone in the building.
- **Interaction On Linked Views.** The simulation is manipulated by direct interaction over the 3D animation and report views. For instance, a blockage can be introduced via the 3D view, and simulation rerun to generate new (rerouted) paths for impacted evacuees; the report view is updated to reflect the situational change. Similarly the interaction with the reports menus, charts, etc. temporally updates the 3D animation view. All such operations are performed in near real-time as all computation are performed on simplified graphs, promoting interactive visual analysis.

4. IMPLEMENTATION

Our system is built utilizing open source toolkits on a Linux system, using Python 2.7.[19] 3D rendering is done with the Visualization Toolkit(VTK)[20] and the GUI is produced with QT4.[21] We use a PostgreSQL database with the PostGIS extensions. The database is accessed running on the local machine with direct package calls. The reports section is an HTML/Javascript window inside a QT4 widget. This allows easy porting to mobile device browsers.

5. EXAMPLE SCENARIO

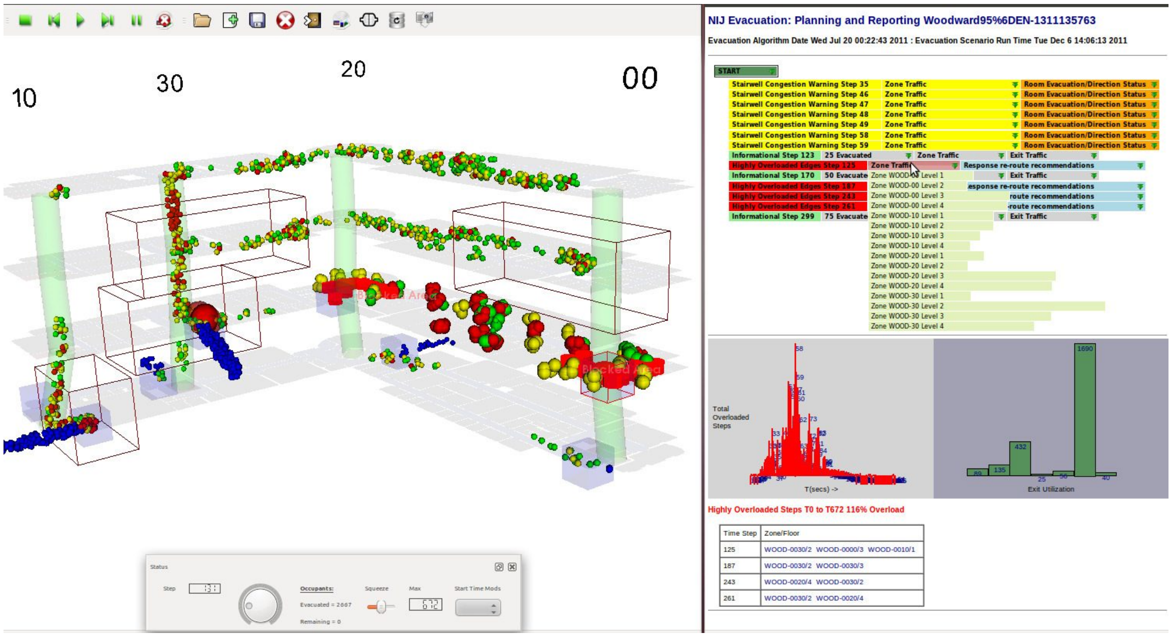
Next we describe three experimental scenarios to illustrate the use of our system. In this scenario, there are 3500 evacuees. The maximum egress capacity is set to 1 evacuee per cubic foot and navigation speed is set at 3ft/sec (congestion can slow down or halt evacuees during a simulation). The building is loaded to 95 percent capacity.

5.1 Situation 1: No Blockages

Figure 4 is a screen shot of our visual analytic system, loaded with a campus building with no inaccessible areas. Evacuees are represented by spheres, clustered visually using an algorithm which arranges them based on egress width. A small red sphere indicates an evacuee cluster on a congested step. Large red spheres of varying opacity indicate congestion greater than 116 percent of rated capacity. In the timestep shown in Fig. 4, the user has clicked on the reporting panel (upper right), representing a highly loaded event at timestep 108 sec. The user has also rolled over the zone congestion bar to reveal the detailed zone congestion graph. As indicated by the bar at Zone 30 Level 2 this is the most traveled and congested route. The application suggests that a responder be dispatched to this area. As the user rolls over the associated 'Response Reroute Recommendations' menu bar in the list the recommended action can be made visible.

Access for responders can be found by looking at the green exit utilization bars (lower right bar chart of Fig. 4), and choosing a low utilization exit. The zone exits in this scenario that are not utilized are on the 1st level. This type of information can be a powerful dispatch tool for the emergency commander to make an informed decision.

Each bullet in the significant event list (upper right panel in Fig. 4) serves as a visual clue to the overall execution of the scenario. The list covers the entire evacuation. The yellow stairway warnings can be drilled deeper to see which areas are becoming congested. These cues are important for responders to quickly react during the beginning of an event or if a campus lock down has been released. The room evacuation, direction status options can be rolled over to indicate the direction from which the traffic is proceeding, which in turn could result in congestion at a later point.



NIJ Evacuation: Planning and Reporting Woodward95%6DEN-1311135763

Evacuation Algorithm Date Wed Jul 20 00:22:43 2011 : Evacuation Scenario Run Time Tue Dec 6 14:06:13 2011

Stairwell Congestion Warning Step 35 | Zone Traffic | Room Evacuation/Direction Status
Stairwell Congestion Warning Step 46 | Zone Traffic | Room Evacuation/Direction Status
Stairwell Congestion Warning Step 47 | Zone Traffic | Room Evacuation/Direction Status
Stairwell Congestion Warning Step 48 | Zone Traffic | Room Evacuation/Direction Status
Stairwell Congestion Warning Step 49 | Zone Traffic | Room Evacuation/Direction Status
Stairwell Congestion Warning Step 58 | Zone Traffic | Room Evacuation/Direction Status
Stairwell Congestion Warning Step 59 | Zone Traffic | Room Evacuation/Direction Status
Informational Step 123 | 25 Evacuated | Zone Traffic | Exit Traffic
Highly Overloaded Edges Step 125 | Zone Traffic | Response re-route recommendations
Informational Step 170 | 50 Evacuate | Exit Traffic
Highly Overloaded Edges Step 187 | esponse re-route recommendations
Highly Overloaded Edges Step 243 | route recommendations
Highly Overloaded Edges Step 261 | -route recommendations
Informational Step 299 | 75 Evacuate | Exit Traffic

Zone WOOD-00 Level 1
Zone WOOD-00 Level 2
Zone WOOD-00 Level 3
Zone WOOD-00 Level 4
Zone WOOD-10 Level 1
Zone WOOD-10 Level 2
Zone WOOD-10 Level 3
Zone WOOD-10 Level 4
Zone WOOD-20 Level 1
Zone WOOD-20 Level 2
Zone WOOD-20 Level 3
Zone WOOD-20 Level 4
Zone WOOD-30 Level 1
Zone WOOD-30 Level 2
Zone WOOD-30 Level 3
Zone WOOD-30 Level 4

Highly Overloaded Steps T0 to T672 116% Overload

Time Step	Zone/Floor
125	WOOD-0030/2 WOOD-0000/3 WOOD-0010/1
187	WOOD-0030/2 WOOD-0030/3
243	WOOD-0020/4 WOOD-0030/2
261	WOOD-0030/2 WOOD-0020/4

Status
Step 131
Occupants:
Evacuated = 2667
Remaining = 0
Max 672

Figure 5. Two blockages have been placed in the building at 45 seconds into an emergency evacuation. (1) Floor 2 at zone 20 stairwell and (2) floor 2 at zone 00 stairwell. Individuals are shown trapped between them by enlarged spheres. Wireframe cubes indicate traffic flow areas, obtained by rolling over the bar chart at the bottom of the report window.

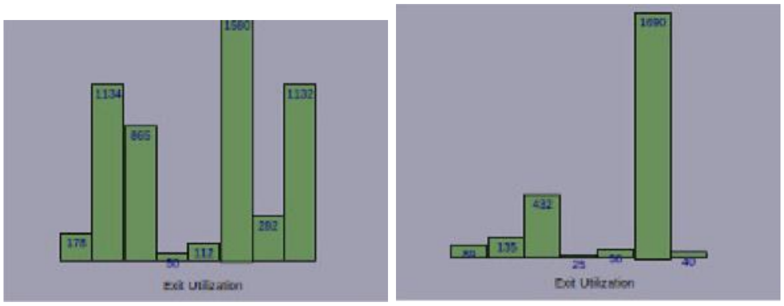


Figure 6. Before (left) and after (right) blockage exit utilizations. Here each green bar represents a specific exit with counts of people at that exit. On the left, there are people at all 8 exits, while on the right only 7 exits are being utilized.

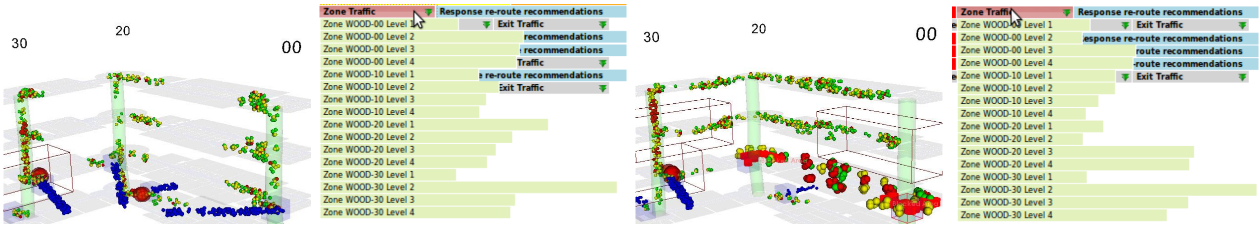
5.2 Situation 2: Induced Blockages

As shown in Fig. 5, two blockages have been introduced into the scenario of Fig. 4. The blocked areas represented by red squares spread out over several square feet on the second floor at zone 00 and zone 20. In this example, a total of 3100 evacuees were rerouted and all reporting recalculated in 2.8sec. Note that in the control bar at the bottom of the 3D animation view the maximum evacuation time has increased to nearly 7 minutes from less than 3 minutes.

Fig. 6 shows the drastic shifts in the movement of people from a standard evacuation of the building. This example serves to show that evacuation modeling of normal (non-blocked) scenarios is considerably different than when blockages are introduced. In particular, notice the difference in the utilization of the exits. In the blocked case, the congestion occurs earlier and is steeper, resulting in longer times for all evacuees to exit the building.

Mouse rollover on the significant event list view in Fig. 5 shows that the third and fourth floors are getting backed up above the exit at zone 20 floor 2 which is loaded heavily even during a normal scenario. Because the event occurred early there were a number of evacuees occupying the upper floors.

Fig. 7 further contrasts the blocked and non-blocked cases. Here Figs. 7a,b illustrates the unblocked case 108 sec. into the evacuation, and Figs. 7c,d for the blocked case at 131 sec. We compare the zone traffic via the light



(a) (b) (c) (d)

Figure 7. Contrast between normal (unblocked) vs. blocked scenarios. Two blockages have been introduced. (a,b) 3D view at 108 seconds into the evacuation, with top floors mostly evacuated, (c,d) 3D view at 139 sec. Floors 3 and 4 are still heavily occupied. Zone traffic (right panels) confirm and illustrate the aggregate picture of the traffic across the entire evacuation.

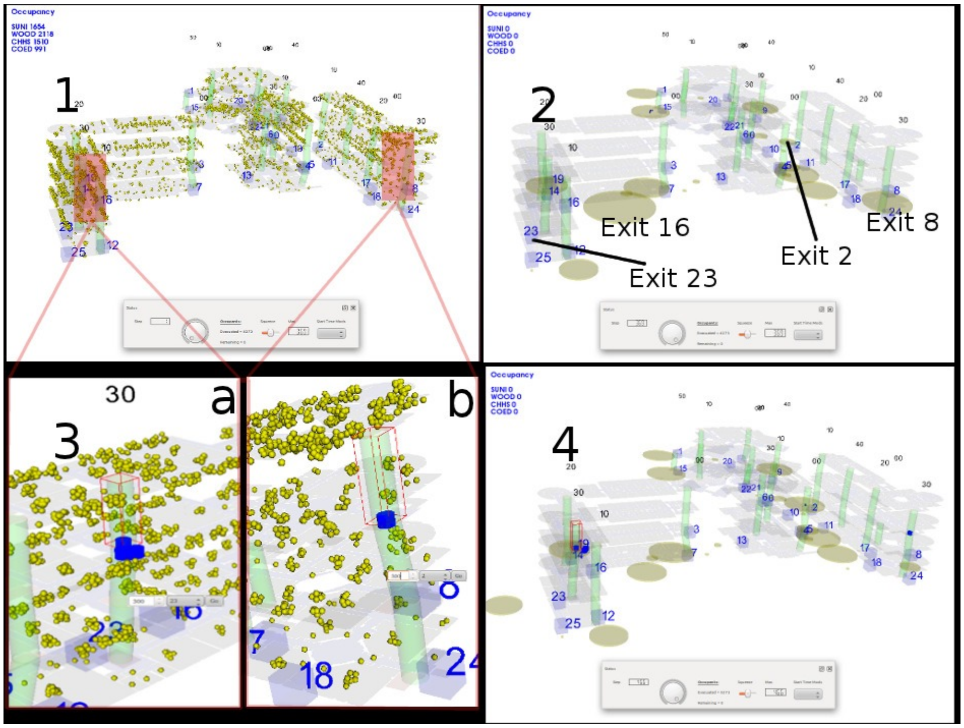


Figure 8. Rerouting evacuees from congested areas in a 4 building evacuation simulation. **1.** Four building cluster. **2.** Resulting evacuee density circles after simulation. **3a and 3b.** User added rerouting flags as indicated by the blue discs and associated with the opaque red rectangles in 1. **4.** Resulting evacuee density circles after modified simulation, changing the routes of evacuees to exit 2 and exit 23 in their respective buildings.

green bar charts. The bar charts show total zone traffic from the beginning to the end of the evacuation scenario. Even though the bar chart for the blocked case is the total picture it is still clear from both the building and zone traffic charts that the traffic on the top two floors is heavier in the blocked case, and in particular, is shown by the size of the bars labeled Zone WOOD-20 Level 3, Zone WOOD-20 Level 4, Zone WOOD-30 Level 3, and Zone WOOD-30 Level 4. The total picture of the bar chart and the single step picture of the building reinforce each other.

5.3 Situation 3: Rerouting Evacuees

When certain parts of a building are blocked, we can reroute evacuees in that area to other nearby less utilized exits. Also, our system permits a selected number of evacuees to be rerouted to reduce congestion at a stairwell or exit. This operation is performed in real-time and the simulation played through to evaluate the traffic or congestion patterns resulting from such an intervention, and thereby result in dispatching a responder to the affected area.

Figure 8 illustrates an example evacuation from a cluster of 4 academic buildings. The original evacuee densities are illustrated in panel 2. Exits 8 and 16 are heavily used, as indicated by the area of their exit circles. The red cubes in panel 1 have been interactively selected (panel 3 shows a zoomed-in view of these areas) for rerouting occupants within those areas. This is followed by specifying the number of evacuees to be rerouted to specific exits (here exits 2 and 23 were chosen). Panel 4 shows the results of these actions, leading to reduced densities at exits 8 and 23.

This function has value for planning and training. Building lockdown and release operations can benefit from such 'what-if' style scenarios that brings together rich spatio-temporal information into the hands of first responders. In this example, running the entire scenario from initiation to results and analysis took approximately 2 minutes. When large collections of buildings are involved with traffic routed to the adjacent street networks, such tools can be invaluable for effective and timely evacuation as well as optimal asset deployment.

6. EVALUATION: TABLETOP EXERCISE

The development of our application has included regular feedback and demonstrations with campus emergency and safety personnel, including the chief of police, other safety officers, and campus business continuity staff. As part of evaluating the system, we conducted a table top exercise with our campus police. We ran the application through three different scenarios to determine our system's usability, effectiveness, and need for improvements. A business continuity office staff member designed the scenarios. The campus police chief, a senior police officer, and the software team participated in the exercise.

All three scenarios involved a cluster of four campus buildings and a base scenario for the evacuation of approximately 5000 evacuees. The preprocessing step was timed at approximately 8 minutes. All simulations used this base evacuation object. Video of each of the 3 exercises (excerpts in attached video) were recorded for analysis, followed by feedback from the emergency personnel. The system was operated by a member of the software team while commands were received from the police chief.

6.1 Scenario 1. Gas Leak in Building.

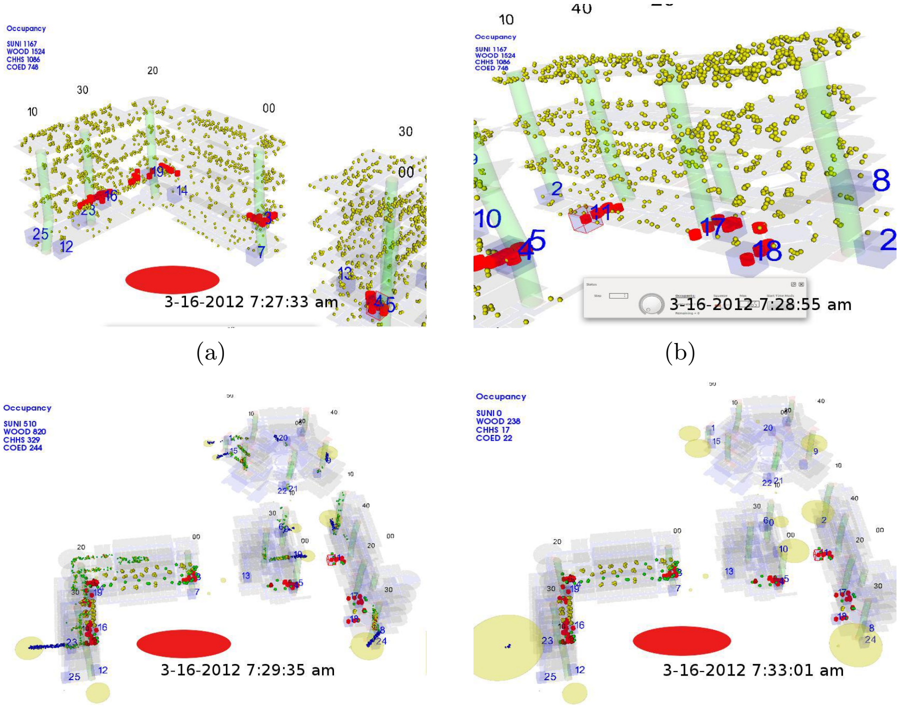
Figure 9 shows time sequenced snapshots of a simulated gas leak somewhere in the exercise area. Initially the gas leak was reported as "near Woodward Hall". The police chief requested a simulation start. As seen in Fig. 9(a) the buildings are being evacuated as expected with all exits being utilized. Several seconds into the simulation (sim time: 7:26:38) a report is received that the leak is in the "courtyard", as shown in the red ellipse in the figure. The simulation is halted and reset. The police chief instructed first responders to be dispatched to the building exits facing the courtyard. Also, entrance/exits into the courtyard were to be blocked from further use.

The simulation was restarted based on the new situation. We interacted with the software by placing blockages at the requested areas from 7:27:27 until 7:28:19 (Figs. 9(a), 9(b)). At this point, the software began to recalculate the 5000 evacuee paths. At 7:29:08 calculations were completed and the reporting process rebuilt, including the scenario timeline and the temporal congestion and exit utilization charts. The police chief requested to see the simulation based on the new situation.

Evacuees are confirmed to be exiting the buildings away from the hazard, as seen in Figs. 9(c), 9(d). The simulated time to exit all buildings increased from 318 seconds to 687 seconds. There were large evacuee populations in the areas of Woodward hall opposite the hazard and it was noted that due to the blockages in the second floor, some of the evacuees were trapped.

6.2 Scenario 2. Active Shooter in Building.

Figure 10 shows time sequenced snapshots of a simulated active shooter exercise in the Woodward hall. First, the police chief ordered a campus lock down and the building to be evacuated. At this point we switched from the base evacuation scenario of the lower quad (building cluster) to a base scenario of Woodward hall. The reason for this is that the scenarios are built as objects and run apriori. We could have placed blockages in the locked down buildings but chose to open a single building scenario for the purposes of the exercise.



(a) (b)

(c) (d)

Figure 9. Tabletop Exercise: Gas Leak in Building. At approximately 7:20 a gas leak is reported. Lower quad campus possibly affecting four buildings. Evacuation simulation begins. At **Sim. Time: 7:26:38.**, leak confirmed near red ellipse. Simulation suspended, assets are deployed to prevent evacuation into the hazard. Views modified to simulate asset activities at building exits. (a)**Sim. Time; 7:27:33.** Blocking building exits into quad (b)**Sim. Time; 7:28:55.** Blocking building exits from adjacent buildings into quad complete. Signal sent for application to perform situationally aware rerouting, (c) **Sim Time; 7:29:35.** Visualization of new simulation, evacuation in progress, simulating responders interaction at exits to affected areas, (d) **Sim. Time; 7:33:01.** Evacuees are avoiding hazard and and exiting to safe zones, evacuee densities are indicated by areas of yellow circles.

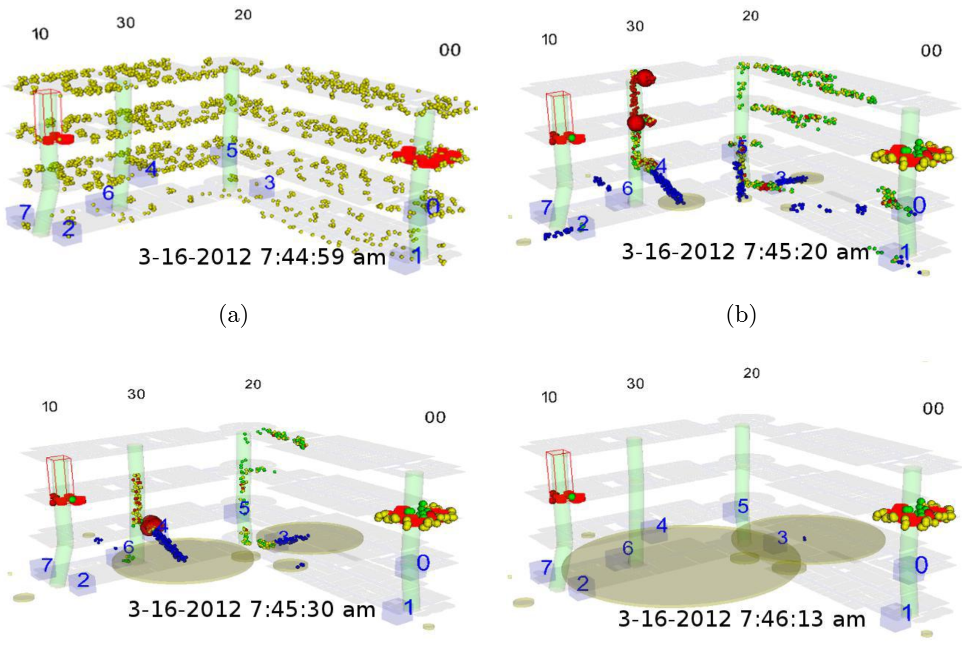
Some highlights in this exercise include: building rerouting and reporting occurs in 49 seconds (3 seconds for evacuee rerouting and 46 seconds for report generation). The total time here is similar to the multi-building evacuation because our base scenario included 3600 evacuees. This simulates a highly overloaded building to exercise the software for testing.

6.3 Scenario 3. Explosion in Utility Plant

An explosion in the RUP (regional utility plant) building created a scenario where the four building evacuation simulation of Figure 9 was also used. This scenario also found evacuees blocked in the upper floors and the explosion created a hazard in the building courtyard. As reports were received the building floors were blocked and the simulation was started. As more reports were received it became obvious that the personnel would exit toward the hazard in the courtyard. The exit density circles alerted the police chief to this problem and emergency personnel were dispatched to redirect these evacuees. At this point the police chief requested the exits facing the courtyard to be blocked. The simulation was restarted and evacuation times and exit results were evaluated as in previous scenarios.

6.4 Analysis and System Assessment

We detail below both the observations from first responders as well as the important features and current limitations of our evacuation system, as noted from the table top exercise. Overall, the feedback from the chief of police (who played the role of incident commander) and his officers was positive and consisted of the following observations:



(a) (b)

(c) (d)

Figure 10. Tabletop Exercise:Active Shooter. At approximately 7:40 a shooter is reported at Woodward hall. Campus is locked down. Reports are received that 3rd floor stairwells are blocked at each end of the building. At 7:44:10 blockages are placed in Woodward by the operator and the simulation is recalculated. (a)**Sim. Time; 7:44:59.** Processing for new evacuation simulation is complete, commander orders visualization of new simulation, (b) **Sim. Time; 7:45:20.** Trapped evacuees noted at 3rd floor zone 00 and zone 10, (c)**Sim. Time; 7:45:30.** Extreme congestion noted at stairwells in floors 2, 3, and 4 at zone 30. (d)**Sim. Time; 7:46:13.** Simulated evacuation complete. Evacuee populations are indicated by approximate area occupied circles.

- The ability to see the 3D layout of the buildings and surrounding areas and get a sense of the current situation was considered the most valuable. The ability to see the evacuation unfold, the buildup at congestion points and the ability to direct evacuees away from a hazard were considered critically important.
- The near real-time responsiveness of the system and the ability to see the evacuation under blockages was valuable for assessment and taking appropriate action, such as dispatching first responders.
- In the gas leak exercise, a review of the evacuation helped the commander quickly size up the situation (number of evacuees, exit routes, etc) and order a building evacuation. Once the hazard was located, evacuees were routed away from it by injecting suitable blockages at key points in the building.
- In the active shooter scenario, the police chief noted that the exit utilization and congestion reports would be an invaluable tool for first responders to analyze the condition of a building and dispatch personnel.
- Additional work on the user interface will be needed to ensure minimize delays during a dynamically changing situation; for instance, blockages are specified one at a time; a 'lasso' style interface to specify multiple blockages was considered more intuitive and efficient.
- A limitation of the current system is its inability to localize blockages to the exits, trapping evacuees in the vicinity.
- A visualization issue that is common to visual analytic systems is visual clutter and the ability to unambiguously visualize critical information. As we extend our system to incorporate tens of buildings in evacuation scenario, these issues will require careful design and representation choices, with input from responders.

7. CONCLUSIONS

In this work we have presented a visual analytic system for situationally aware evacuations of large urban structures. The goals of this work are to provide visual analytic tools that can be used in real-world scenarios (large urban structures, dense collections of buildings) and more importantly, be able to run evacuation scenarios in the context of dynamically changing conditions. We have developed and used an LOD representation of building graphs that can be used as part of a visual analytic system for near real-time response. This in turn permits situational changes to be incorporated into the underlying models and evacuees rerouted. Finally, our visual analytic system provides recommendations through the reporting functions that can be used for effective use of scarce resources in dispatching responders to areas of need during the emergency. We evaluated our system with first responders, including the campus police chief, a senior police officer and public safety and business continuity/planning personnel. Input from these experienced personnel is invaluable. A tabletop exercise was performed with three different scenarios (gas leak in building, active shooter, and explosion) overseen by the police chief, acting as the situation commander. Overall, the system performed well, as evidenced by direct feedback from the first responders, with valuable suggestions to improve the system. We are currently looking into making our evacuation model more realistic, by incorporating building occupancies as a function of time; in many instances, this is feasible, via employee records, university course schedules, etc. This also presents responders with more realistic information during the emergency.

8. ACKNOWLEDGMENTS

This work was supported by a grant from the National Institute of Justice, Office of Justice Programs.

APPENDIX A. MODIFIED CAPACITY BASED ROUTE PLANNER

Algorithm 1: Modified Capacity Based Route Planner

```
Input: ;
(1) G(N,E): Directed Graph, N nodes, E edges;
(2) Node Properties: capacity, occupancy;
(3) Edge Properties: capacity, travel time;
(4) Set of Source Nodes;
(5) Set of Destination Nodes;
(6) Set of evacuee objects;
Result: Evacuation Plan : Routes with schedules of evacuees on each route
foreach evacuee i at each source node s do
    path_found = find shortest path p from s to all destinations    with available capacity;
    if path_found then
        while p < max_capacity do
            /* can route evacuee via p */ evacuees[i].path = p;
        end
    end
    move all evacuees;
end
while evacuees not at destination nodes do
    move all evacuees;
end
```

REFERENCES

[1] Lu, Q., George, B., and Shekhar., S., "Capacity constrained routing algorithms for evacuation planning: A summary of results.," *Springer-Verlag Berlin Heidelberg 2005* **3633**, 291–307 (Sept. 2005).

[2] Kamiyama, N., Katoh, N., and Takizawa, A., "An efficient algorithm for the evacuation problem in a certain class of networks with uniform path-lengths.," *Discrete Applied Mathematics* **157**, 3665–3677 (2009).

[3] Shekhar, S. and Yoo., J. S., "Processing in-route nearest neighbor queries: a comparison of alternative approaches," in [*Proceedings of the 11th ACM international symposium on Advances in geographic information systems*], (2003).

[4] Kim, S., Shekhar, S., and Min, M., "Contraflow transportation network reconfiguration for evacuation route planning," *IEEE Trans. on Knowl. and Data Eng.* **20**, 1115–1129 (August 2008).

[5] Lo, S. M., Liu, M., and Yuen, R. K. K., "An artificial neural-network based predictive model for pre-evacuation human response in domestic building fire," *Fire Technology* **45**, 431–449 (Sept. 2009).

[6] Fang, G., "Swarm interaction-based simulation of ocucupant evacuation.," in [*2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*], (2008).

[7] Guy, S. J., Chhugani, J., Kim, C., Satish, N., Lin, M., Manocha, D., and Dubey, P., "Clearpath: Highly parallel collision avoidance for multi-agent simulation," in [*Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2009)*], Grinspun, E. and Hodgins, J., eds. (2009).

[8] Castle, C. J. E. and Crooks., A. T., "Principles and concepts of agent-based modelling for developing geospatial simulations," in [*Centre for Advanced Spatial Analysis. University College London*], **110**, 1–52 (2007).

[9] Thomas, J. and Cook, K., [*Illuminating the Path: The Research and Development Agenda for Visual Analytics*], IEEE Press (2005).

[10] Andrienko, G., Andrienko, N., and Bartling, U., "Interactive visual interfaces for evacuation planning," in [*Working Conference on Advanced Visual Interfaces(AVI) 2008 Proceedings*], 472–473, ACM Press (2008).

[11] Campbell, B. and Weaver, C., "Rimsim response hospital evacuation: Improving situation awareness and insight through serious games play and analysis," *Journal of Information Systems for Crisis Response and Management* **3**, 1–15 (Jul-Sept 2011).

[12] Kim, S., Maciejewski, R., Ostmo, K., Delp, E., Collins, T., and Ebert, D., "Mobile analytics for emergency response and training," *Information Visualization* **7**(1), 77–88 (2008).

[13] Kim, S., Yang, Y., Mellama, A., Ebert, D., and Collins, T., "Visual analytics on mobile devices for emergency response," in [*IEEE Symposium on Visual Analytics Science and Technology (VAST)*], 35–42 (2007).

[14] Meiguins, B. and Meiguins, A., "Multiple coordinated views supporting visual analytics," in [*Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration*], 40–45, ACM, New York, NY, USA (2009).

[15] Ivanov, Y., Wren, C., Sorokin, A., and Kaur, I., "Visualizing the history of living spaces.," *IEEE Transactions on Visualization and Computer Graphics* **110**, 1153–1159 (November 2007).

[16] Andrienko, N., Andrienko, G., and Gatalsky., P., "Towards exploratory visualization of spatio-temporal data.," in [*3rd AGILE Conference on Geo-graphic Information Science*], (2000).

[17] Stasko, J., Gorg, C., Liu, Z., and Singhal, K., "Jigsaw: Supporting investigative analysis through interactive visualization," in [*Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology*], 131–138, IEEE Computer Society (2007).

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

[19] van Rossum, G. and Drake, F., [*An Introduction to Python*], Network Theory Ltd. (2003). WWW: www.python.org.

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

[21] "Qt: Cross-platform application and ui framework." http://qt.nokia.com.