Crowd modelling for quasi-real-time feedback during evacuation in a situational awareness system

Paul Simon Townsend

Abstract

Evacuation modelling has traditionally been limited to prior analysis of the geometrical design of spaces. This approach is limited by the number of scenarios that can be tested. A scenario-independent evacuation model would rely on real-time information processing. This approach allows the generation of an optimum evacuation strategy that minimises evacuation time and crowd density, and is tailored to the current situation at hand. This paper describes the requirements and development of such a crowd model. Graph theory and network flow heuristics are combined with agent movement, based on the individual determination of speed by meso-scale density assessment.

Keywords: evacuation; crowd modelling; situational awareness; optimisation; prediction; evacuation strategy

1. Introduction

In general, buildings often have to satisfy certain safety standards, for example guidelines such as BSI (2008), NFPA (2012), and IMO (2007). Different methods of assessing such compliance can be considered. At the very basic level, a number of measurements can be taken and hand calculations can determine the width required for corridors, doors or escape routes, based on fixed values. However, this simplistic approach often assumes an even distribution of exit usage that is unlikely to occur in reality. For a number of years, evacuation simulations have
been used to analyse the evacuation of building designs in much greater depth (Cabinet Office, (2010)) For example, they can determine the evacuation time of a given number of people in a building, predict congestion during such an evacuation and test different scenarios of use (Pauls (1987, 1988), Kuligowski and Peacock (2005)). However, these models are limited by the number of scenarios that can be run. Evacuation time, congestion etc. is dependent on many variables such as location and nature of threat (Still (2007)).

Recently, with the advent of smart spaces, sensor networks and greater computing power, the ability for such models to be able to simulate live situations has become a possibility (Aydt, Lees, Turner and Cai (2014)). Integrated with live data feeds of certain crowd parameters, such feedback can aid situational awareness and influence evacuation strategies.

eVACUATE1, is a holistic, scenario independent, situation awareness and guidance system for sustaining the active evacuation route for large crowds. It allows a forecast of crowd movement during an evacuation to be provided to decision makers. Data from sensors is processed and used by the crowd model to predict congestion and evacuation time. Time-dependent route optimisation is then undertaken, which minimises these variables to provide an optimum strategy for the specific scenario at hand. The generalised eVACUATE system concept is shown in Fig. 1.

This paper describes the research towards the crowd modelling element of eVACUATE, considering the requirements for practical use of the crowd models in the situational awareness system, contextual definition of quasi-real-time, model structure, validation and optimisation of route choices.

2. Requirements and Definitions

Within the eVACUATE project, a number of requirements have been developed, with influence from potential end users, and industrial partner expertise. These include the following:

1 The research leading to these results has received funding from the European Union’s Seventh Framework Programme (FP7/2007–2013) under grant agreement number 313161 (eVACUATE), http://www.evacuate.eu
The system architecture or details on provision of information or data fusion from sensors is not covered in this paper. The requirements for the modelling elements are addressed below.

2.1. Feedback Congestion Prediction

Here, it is assumed that initial parameters for the model, such as building geometry, flow rates and live data from sensors is available and has been used to initiate a simulation that runs to completion. Feedback from the model is required to show:

- Predicted evacuation time
- Crowd density periodically over evacuation
- Crowd speed periodically over evacuation

Predicted evacuation time is given per simulation run. Crowd density and speed are provided over a pre-defined period. Such a period is customisable in the models, but a baseline has been determined at 1 minute. This means that snapshots of density and speed as predicted by the model are provided for every minute during the simulated evacuation.

2.2. Feedback to be provided in close to real-time

In simulation terms, ‘real-time’ is defined to be the simulation speed matches real-time. i.e. 1 second of simulation time is completed in 1 second of real-time. However, this speed of computation is clearly not useful for providing a prediction that can increase situational awareness during a crisis. It is the output of the simulation that would ideally be provided in real-time to the control room operatives. Given the complexity of an evacuation simulation, there is an expected feedback period that is acceptable to the control room. Therefore, feedback is provided as quickly as possible in ‘quasi-real-time’. This means ‘almost real-time’, but should be defined specifically for each venue.

This period will differ for different venues, but a benchmark speed has been defined such that the simulation should run at a minimum of 30x real-time. This means that a 10 minute simulation can be computed in 20 seconds. In reality, faster computational speeds have been achieved. As a general benchmark for this paper and non-specific venues, quasi-real-time is defined to be providing feedback within 20 seconds of a request from the user.

2.3. Optimum Evacuation Routes

The simulation is required to optimise the evacuation strategy of the venue to the current situation. This means that a general evacuation strategy, where routes are pre-defined, often needs to be changed dynamically during an evacuation due to the unfolding situation. For example, a certain route is blocked due to a particular threat;
unevenly distributed crowds across the venue leads to blocked exits and re-routing is required etc. The evacuation is then directed by the best judgement of control room operatives and decision makers. A system that could optimise the evacuation given any particular situation would clearly be of benefit. Therefore, the model is required to output the ‘optimum evacuation route’ for each area in the venue.

It is noted that finding a truly optimum solution to the problem would be extremely difficult. In fact, many such problems like the Vehicle Routing Problem and Travelling Salesman Problem are known to be NP-complete in combinatorial theory. In practical terms, such a precisely defined optimum is not required. The solution found must be easily implemented by dynamic information provision and the direction of staff on the ground. Therefore discrete algorithmic methods are employed to find the ‘best available’ solution. It is this solution that we henceforth refer to as the ‘optimum’.

3. Model Structure

3.1. Model Definition

The modelling space is $\Omega \subseteq \mathbb{R}^3$. $\Omega$ is an arbitrary network, defined by a weighted graph $G^\Omega = (V, E)$ where $V$ is a set of vertices (1), $E$ is a set of edges (2) and there is an associated function $\alpha: E \rightarrow \mathbb{R}$ on the edges. This is hereby referred to as the ‘evacuation network’ or ‘network’ (Fig. 2.).

\[
V = \{(v, a)|v \in \mathbb{R}^3, a \subset \mathbb{R}^2\} \quad a = \begin{cases} \text{circle of radius } r, & r \in \mathbb{R} \\ \text{rectangle of height } h \text{ and width } w, & \text{rotation } \theta \end{cases}
\]  

(1)

\[
E = \{(u, v, w, l)|u, v \in V, \ u \neq v, \ w, l \in \mathbb{R}\}
\]  

(2)

Agents are represented by a singular three-dimensional location. There is no notion of physical body space of agents in the model. This means that the computational code that approximates the model can be threaded so that each agent can process their next move in parallel to a number of other agents.

Feedback from the eVACUATE system provides the location of crowds spread across the network. This data is gathered from combined sensor data and interpolated data where sensors cannot be located. The model then starts an evacuation procedure defined by a Finite State Machine (FSM). The FSM is the intelligence of the model that instructs agents to travel to certain vertices. Vertices will be normally be defined as exits, but, in certain circumstances, evacuees will travel via other nodes to a fixed or closest exit point. An example of this is found in
some cruise ship evacuations where passengers will travel to their cabins to retrieve their life jackets before heading to a muster station. An example of a finite state machine is shown in Fig.3.

The actual movement of agents to each decision point in the finite state machine is based on the mesoscopic movement and routing principles outlined below.

3.2. Mesoscopic movement

The speed of movement at any time is calculated by measuring density in a localised space. The space in which density is measured is created arbitrarily by defining ‘bins’ along the length of an edge. These bins are given an arbitrary minimum length and the edge is divided into bins of this length. Any ‘end’ bins that may be smaller than the minimum defined length are amalgamated into the conjoining bin along the edge. A vertex is classed as one bin. Therefore, it is clear that each bin has an area that can be calculated.

As agents move through the model space, they move from bin to bin. A count is kept of each bin as to how many agents are inside at any time. Therefore, the density of a bin can be calculated. In fact, the mesoscopic model calculates an agent’s current density, $\rho$, as a weighted representation of the density of an agent’s current bin, $\rho_0$, and the 3 bins in front of it, $\rho_1, \rho_2, \rho_3$ (3). The calibrated values are given in (4).

$$\rho = a \rho_0 + b \rho_1 + c \rho_2 + d \rho_3$$  \hspace{1cm} (3)

$$a + b + c + d = 1, \quad a > b > c > d, \quad a, b, c, d \in \mathbb{R}^+$$

$$a = \frac{8}{19}, \quad b = \frac{4}{19}, \quad c = \frac{2}{19}, \quad b = \frac{1}{19}$$  \hspace{1cm} (4)

Each agent is given a maximum speed, based on a normalised distribution. An agent will move at this speed in low density. As soon as the density reaches a threshold value, the agent calculates its speed based on a speed-density curve. In fact, any speed-density curve can be implemented in the model, but two main curves are used as defaults, derived from Fruin (1971) and Weidmann (1993). The empirically derived equations from these two data sets are shown below respectively in (5) and (6), where $s_{av}$ is the average speed of the agent’s current bin.

**Fruin:** \[ s_{av} = -0.07194\rho^3 + 0.2277\rho^2 - 0.53083\rho + 1.41895 \] \hspace{1cm} (5)

**Weidmann:** \[ s_{av} = 0.0022\rho^5 - 0.0386\rho^4 + 0.237\rho^3 - 0.548\rho^2 + 0.0341\rho + 1.3303 \] \hspace{1cm} (6)

The individual agent’s speed is calculated based on a normal distribution with $\mu = s_{av}, \sigma = 0.1$. 
3.3. Routing

Routing is achieved macroscopically before the simulation commences using Dijkstra’s algorithm (Dijkstra (1959)) and can be then be updated periodically through the simulation if required. α can be defined to be any cost function. However, the function used as a default to determine the path taken through the network is given in (7). A route map is created in the form of a lookup table from each node to each possible destination as defined in the FSM. The velocity of an agent is calculated using the speed determined by the mesoscopic density assessment and the direction towards the location of the next node.

\[
\alpha(E) = \frac{E_t}{1.36 - 0.41 \sum_{i=1}^{E_n} \tilde{\rho}(E_i)}
\]

(7)

\(E_t\) = edge length  
\(n\) = number of bins in edge  
\(\tilde{\rho}(E_i)\) = average density of bin over \(t\) seconds, \(t\) is an arbitrary constant over all bins

**Outputs**

The model is capable of producing statistics on flow rate across each bin, the density in each bin and aggregating data across multiple bins, which aids the display of complicated information to a control room operator. This is done in xml format across a pre-defined time period, \(t\). Every \(t\) seconds, a snapshot of current density and speed of all bins across the network is aggregated into proportional sections defined for each edge. An example heat map output of density across a network is shown in Fig. 4. Emergent behaviour is shown for the edges approaching the central cluster of stairs, such that higher density is seen approaching a bottleneck at the top end of the edge.

Further outputs from the simulation show the predicted time to hit certain stages of the finite state machine for all agents. For example, all agents have reached the exit points (this is overall evacuation time).
4. Calibration and Validation

The model has been validated against the empirical flow/density curves of Fruin and Weidmann using the flow rate output along each bin of the model. Flow rate is measured across the central edge which has a width of 10m. Varying entry flows to the edge were introduced to determine a spread of densities and flow rates. The flow rate and density are taken to be the average over 60 seconds of simulation time to mimic how the empirically derived curves are measured in reality (Fig. 5, Fig. 6).

To check that a smaller sample of average density still averaged around the speed/density curve, flow was averaged over \( N \) seconds of simulation time (\( N = 5, 10, 15, 20, 30, 60 \)). This is confirmed in the graph shown in Fig. 7, measured against a polynomial curve derived from empirical data. As the time period used to gather data shortens, the greater the dispersion is around the empirical curve. As the density increases, it is clear that the error values above and below the curve are less accurate for each test. Both these phenomena are to be expected when one considers the range of values in the empirical data sets used to construct such curves, for example in experiments performed by Zhang and Seyfried (2012).
4.1. Validation of quasi-real-time feedback

Agents are parallel processed in batches, with the batch size being defined per network from a single agent to all agents in the network at any time. This allows for the model to be calibrated for specific networks to increase the computational speed of the model.

Therefore, the speed of simulation could potentially be increased further through calibration of the batch size, bin size, simulation movement step time. Network simulations were run using a laptop with specification: Intel® Core™ i7 4700MQ 2.40GHz; 8GB 1600MHz DDR3; NVIDIA® GeForce® GTX 765M. Values of batch size 5, bin size 1.5m, and 1/3 seconds per movement step were used. Due to the highly parallel nature of the computational code for the mesoscopic model, an increase in RAM, speed of processor, and/or number of processing cores would increase the computational capability of the model even further. The requirement for quasi-real-time feedback is satisfied by the current models, without extensive calibration and on a high-end non specialist laptop. This is shown for varying numbers of agents constantly being simulated for a 10 minute period in Table 1.

Table 1. Computational speed of network simulation for varying agent numbers

| Number of Agents | Time taken for a 10 minute simulation (secs) | Equivalent average speed (x real-time) |
|------------------|-------------------------------------------|--------------------------------------|
| 1,000            | 1.9                                       | 316                                  |
| 2,000            | 1.9                                       | 316                                  |
| 5,000            | 2.9                                       | 207                                  |
| 10,000           | 3.9                                       | 154                                  |
| 25,000           | 7.1                                       | 85                                   |
| 50,000           | 13.5                                      | 44                                   |

5. Optimisation of Evacuation Routes

As previously defined, the ‘optimum evacuation route’ is not a strictly optimum, but a route that is calculated based on trying to minimise evacuation time and congestion over the routes of a network. Currently, a constraint is applied, such that only one direction is taken at any decision point. This means that, where a truly optimum solution
may require the use of two or more routes from a waypoint, only one is followed by the agents. This restriction may be lifted for certain venues if such a strategy can be implemented on the ground, although it is believed that this would require complicated management of staff to facilitate such a strategy and may be counter-productive to the overall evacuation.

This work is in its early stages, although an iterative process is currently being used to implement an ‘all or nothing’ assignment of routing where the cost function (7), uses the densities stored from the previous iteration to calculate the route costs. Future routing algorithms will seek to reduce the number of iterations before a suitable evacuation time is obtained by using a localised or global density avoidance heuristic, combined with the cost function from the previous iteration. Another possibility is to start the iterative process without the need to run agents through the model for the first \( N \) iterations. It is thought that a combination of these techniques will enable a faster convergence towards an optimum solution, providing quicker feedback to the control room.

6. Conclusions

Evacuation model development is progressing from the traditional pre-analysis of buildings in design stages. Simulation models have been built that can provide live feedback to control room operators and decision makers in quasi-real-time. Such models rely on the timely provision of sensor data, data fusion and a highly computationally efficient evacuation model. It has been demonstrated that the mesoscopic model presented meets the base criteria defined in terms of model validation of crowd movement and for quasi-real-time feedback.

It is intended that this model will be developed further. The next steps are to study how best to obtain optimised routes through the venue within the defined quasi-real-time feedback period; and the inclusion of other mesoscopic or microscopic models that provide a less coarse assessment in areas where the graph theory and mesoscopic movement approach is not ideal (for example large rooms with complex geometry). It is clear that the modelling system within eVACUATE has made excellent progress towards live simulation, fast computational feedback and that the software produced meets the requirements of a number of potential users of the system.

The model is set up such that it can be expanded to different modelling techniques, allowing the combination of macro-, micro- and meso-scopic models within one framework to potentially provide more accurate predictions, more customised network setups for any type of venue.

References

Aydt, H., Lees, M. H., Turner, S. J., and Cai, W., 2014. Toward Simulation-Based Egree Optimization in Smart Buildings Using Symbiotic Simulation, in Weidmann, U., Kirsch, U., Schreckenberg, M. (Eds.) “Pedestrian and Evacuation Dynamics 2012”, vol. 2, pp. 987-999.

Bellomo, Bellouquid, Knopoff, 2013. From The Microscale To Collective Crowd Dynamics, Society for Industrial and Applied Mathematics 11(3), 943-965.

British Standards Institute, 2008. BS 9999:2008: Fire safety code of practice for the design, management and use of buildings.

Cabinet Office, 2010. Understanding Crowd Behaviours Volume 1 – Practical Guidance and Lessons Identified, London, TSO, pp. 64-69.

Dijkstra, E. W., 1959. “A note on two problems in connexion with graphs”, in Numerische Mathematik 1, pp.269–271.

IMO, 2007. Guidelines For Evacuation Analyses For New and Existing Passenger Ships, International Maritime Organization, MSC/Circ. 1238.

National Fire Protection Association, 2011. “Chapter 7: Means of Egress” NFPA 101: Life Safety Code, 2012 Edition, pp. 101-144

Fruin, J. J., 1971. Pedestrian Planning and Design, Metropolitan Association of Urban Designers and Environmental Planners, New York, p. 74.

Kuligowski, E. D., Peacock, R. D., 2005. A Review of Building Evacuation Models, National Institute of Standards and Technology Technical Note 1471, pp. 13-18.

Pauls, J., 1987. Calculating evacuation times for tall buildings. Fire Safety Journal 12, 213-236.

Pauls, J., 1988. “Movement of people”, in SFPE Handbook of Fire Protection Engineering, SFPE/NFPA, pp. 246-268

Still, G. K., 2007. Review of pedestrian and evacuation simulations, Int. J. Critical Infrastructures, 3(3/4), 376–388.

Weidmann, U., 1993, Transporttechnik der Fussgänger, Transporttechnische Eigenschaften des Fussgängerverkehrs (Literaturauswertung), Schriftenreihe des IVT Nr. 90, Zweite, ergänzte Auflage, Zürich, p109.

Zhang, J. Seyfried, A., 2012. Empirical characteristics of different types of pedestrian streams. Procedia Engineering 62, 655-662.