Analyzing Egress Accuracy through the Study of Virtual and Real Crowds

Vincius Cassol, Jovani Oliveira and Soraia Raupp Musse*
Pontifical Catholic University of Rio Grande do Sul - Brazil

Norman Badler
University of Pennsylvania - USA.

ABSTRACT

A variety of scientific models and computational tools have been developed to improve human safety and comfort in built environments. In this work we discuss the use of crowd simulation to reproduce and to evaluate egress performance in specific scenarios. We present *CrowdSim*, a crowd simulation tool designed to automatically reproduce crowd behaviors during building egress. Associated with this kind of software, that aims to be used in real life events, are critical validation questions. Here we present quantitative and qualitative methods to evaluate our *CrowdSim* software, but that can be extended to any crowd simulator. To ground these methods a case study in a real night club evacuation was performed and results are discussed in the paper.

1 INTRODUCTION

The different ways in which a crowd can behave have been studied in disparate areas such as psychology, safety engineering, and entertainment. Various scientific models were developed in order to computationally simulate the behavior of a crowd in a specific environment [12, 13]. When simulating crowds, a set of parameters must be considered in order to reproduce coherent behaviors. Such parameters aim to represent: *i) Environment physical structure*: should provide information about building features such as dimensions, number of floors, number of rooms, number and localization of exits, stairs location; *ii) Environment functionality*: people can act in different ways according to the functionality of the place, such as office, hospital, school, airport, stadium, or arena; and *iii) Population data*: number of people in the environment, age, gender, relationships among them, knowledge about the environment, disabilities, etc.

These factors are just a small set of aspects that can impact in an evacuation process. The variation of human behaviors based on such a multiplicity of factors makes the reproduction and virtual simulation of a evacuation process complex and challenging. In this paper we present *CrowdSim*, a crowd simulation tool developed with its main goal to reproduce human motion behaviors in egress situations. In addition, we also present a *CrowdSim* validation according to international guidelines. To complement the validation result we performed a real life egress exercise which we discuss below. We believe that, by simulating different situations using *CrowdSim*, engineers, designers, and safety managers can study the performance of different evacuation plans and thereby improve such plans as well as safety procedures.

The paper is organized as follows: next we present some related work focusing on environment evacuation and egress simulation. Section 3 details *CrowdSim* while Section 4 describes its validation according to international guidelines. Section 5 describes the experiment performed on a night club and Section 6 presents some final remarks.

2 RELATED WORK

Studying crowd motion behavior may have different goals. In this section we discusses some important approaches which have focused on crowd simulation applied to building egress.

The work of *Fu et al.* [7] was developed with a singular goal: simulate the usual process of evacuation. The motivation was to reproduce pedestrian behavior in order to represent exit selection taking into account a least effort cellular automaton algorithm where the motions and goals used to guide the movements are defined by a probabilistic approach. Chu *et al.* [6] developed the platform SAFEgress (Social Agent For Egress), in which building occupants were modeled as agents able to take actions according to their knowledge of the environment, their interactions with social groups, and the neighboring crowd. According to the authors, results show that agent familiarity with the building and social influences can significantly impact evacuation performance. Huang *et al.* [10] added other environmental factors such as smoke and noxious fumes into an evacuation process. The authors developed MIMOSA (Mine Interior Model Of Smoke and Action) which integrates an underground coal mine virtual environment, a fire and smoke propagation model, and a human physiology and behavior model.

One of the critical points considered when simulating crowds is result validation. First of all, it is important to be sure that the employed simulator is able to produce accurate results. The validation of crowd simulators has been addressed in different scientific approaches and is itself an important field. The work of Kuligowski and Gwynne [11] presents a set of guidelines to be observed as general requirements of crowd modelling on simulation software. The authors attempted to aid users in the selection of an appropriate evacuation model by identifying key factors and explanations regarding project requirements, the background of the model, the current capabilities and characteristics of the model for comparison with other models, and the future prospects of a model for a specific application. In order to validate the *EvacSim* pedestrian model against real-world pedestrian data, the authors made a comparison of flow rates, density and velocity for corridor entry and for merging groups, considering data from simulations and the real world in a controlled environment.

Galea proposed an approach to validate evacuation models in [8]. The validation is described as an ongoing activity that must take into account four different aspects: component testing, functional validation, qualitative validation, and quantitative validation. This approach is applied here in order to evaluate our CrowdSim simulator, as explained in Section 4.

3 *CrowdSim*

CrowdSim is rule-based crowd simulation software developed to simulate coherent motion and behaviors in an evacuation process [5, 4]. It also presents data that are used to estimate human comfort and safety in an specific environment. During the design phase of *CrowdSim*, we endeavored to develop software specifically able to:

- Represent the physical geometry of a building in a 3D environment. Such a representation allows safety engineers to use the software in order to virtually simulate an occupation or

---

*e-mail: soraia.musse@pucrs.br

2016 IEEE Virtual Humans and Crowds for Immersive
Environments (VHCIE)
20 March, Greenville, SC, USA
978-1-5090-0829-2/16/$31.00

evacuation plan attending to real building physical constraints (doors, emergency exits, size of corridors).

- Define the spatial occupation of the population in the environment to reproduce initial conditions for an egress event.
- Model an egress plan in the context of emergency situations triggered by specific events over time: start evacuation, change route, etc.
- To produce a visualization of the simulation as well as to summarize data to be considered in statistical analyzes.

Two key components are considered in *CrowdSim*, organized in distinct modules: *Configuration* and *Simulation*. Figure 1 illustrates the software architecture including sub-modules, the necessary inputs from the user, and produced outputs. In the following sections we describe such modules of *CrowdSim* detailing their inputs, dependencies and work flow.

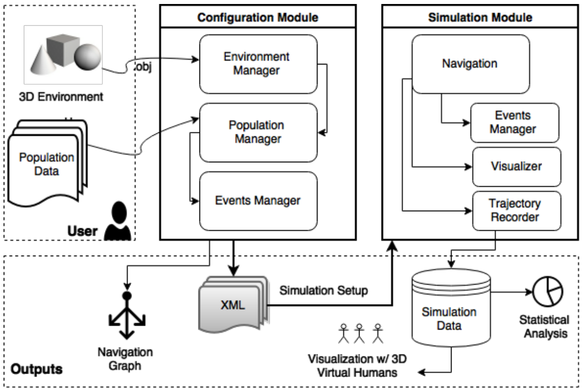


Figure 1: The representative architecture of *CrowdSim*.

3.1 Configuration Module

The configuration module requests, as a first input, the 3D representation of the environment that will be simulated. Such a 3D model will be considered by the *Environment Manager* in order to allow the user to define the walkable regions according to the building structure as well as physical restrictions and obstacles.

When mapping the environment, the user is also able to define *population data*. In order to define the scenario that will be simulated, the *Environment Manager* classifies walkable regions with different purposes. Such walkable areas are called *contexts*. We define three different types of contexts when specifying a simulation environment: *birth*, *motion* and *goal* contexts.

Birth Contexts are used to represent areas of the building where agents should be created during the simulation. In such areas the user is requested to supply the number of agents to be simulated that should be created in such context. Also the user defines the following information based on the total number of agents to be created:

- *Groups Size*: The agents can be created in different groups until they reach the total number that should be created in the context;
- *Creation Time*: Time that groups of agents start to be created after the beginning of the simulation;
- *Time among groups*: Interval of time to be taken into account when creating different groups; and
- *Goal*: The context (or set of possible contexts) to be considered as goals to be reached by an agent when moving.

Goal Contexts are regions of interest to be considered during agent motion (goals). When creating a goal context, the user is requested to define the percentage of agents that should be removed from the simulation when achieving the context, the percentage of agents that should stay moving in such a context, and the percentage of agents that should find another goal and move in that new direction.

The *Motion Contexts* are considered by the simulation algorithm as connection regions between birth and goal contexts. They are important when calculating the agents' motion routes. In addition, a connection graph is built as an output of the configuration module according to connections among contexts and their population specifications.

When the environment is coherently mapped and the user has defined all the *population data*, it is possible to specify in the submodule *Population Manager* how agents should behave when moving. Agent behaviors can be:

- *Goal Seeking*: The agents should seek their goals immediately or vaguely, by performing random motion;
- *Keep waiting*: The agents, when achieving some specific region of the environment, can spend some time in it before looking for another goal;
- *Perform random motion*: The agents can chose random destinations during a specific time, before trying to identify the best path to achieve the main goal.

In *CrowdSim* we can set up two different categories of behaviors *static* and *dynamic*. Static behaviors are always performed as defined by the environment and population manager specifications. On the other hand, dynamic behaviors can be configured in the same way, but these behaviors will be performed just when a trigger is reached. Thus, the responsibility of the *Event Manager* sub-module is to define triggers to perform a series of specific behaviors. An event trigger is composed of the time to occur, a set of dynamic behaviors to be performed at that time, and a time interval between event occurrences.

The correct definition of scenarios is critical in this work, because the combination and analyzes of information is responsible for producing acceptable and valid results. When the environment is totally verified with all walkable regions defined, all the parameters configured, and desired behaviors specified, the user is able to run the second module of *CrowdSim*: Simulation. The data transfer between configuration and simulation modules is currently performed by a scenario file (XML), able to store all the configurations to be observed when computing a new simulation. In addition to the XML file, the Configuration Module also generates a navigation graph with the initial distributed population in the nodes. We implemented a planner that runs offline from *CrowdSim* to read the graph and generate evacuation plans. The main difference between a navigation plan and an evacuation plan is that in the former we know where people start and in the latter we define the distribution of people at any graph bifurcation, i.e. we define the exits for each people/group. Of course, we adopt the important hypothesis that the shortest path is not always the best for crowd simulation.

3.2 Simulation Module

The simulation module of *CrowdSim* is responsible for computing the navigation of virtual agents in a specific environment. Such navigation should coherently take into account agent motion, collision control, speed variation, and other pedestrian behaviors.

A simulation setup, previously defined in the configuration module, is requested as input to the simulation. The simulation computes the routes of each agent to achieve a specific goal. Routes can be computed based on user specification (i.e., a graph determined by the user) or computed by the best paths considering only distance criteria. *CrowdSim* uses A* [9] in order to compute shortest paths. During motion simulation, *CrowdSim* avoids collisions among agents or obstacles using a simple local geometry method. Indeed, it is rule-based and local defined (based on distance proximity). Close agents and their speeds are used in the collision-test to detect a possible collision situation in a next frame. If this situation is going to happen, one of the involved agents (randomly defined) must take a decision: *i)* to change its direction vector (shifts of $+-40$deg are allowed) as a function of goal vector, or *ii)* to reduce its speed. The information about the pair of agents and the decision taken is saved in a list of past actions, which is lost each n frames (we used $n = 10$ in experimental results). If a new collision situation is detected for the same pair of agents and there is still an action in the list of past actions, the agent takes a different decision, i.e. if direction changing is saved in past actions, then a speed changing should be performed. Consequently, agents try to reach their goal, avoiding collisions with others. This method is not free-of-collision, but maximum error of 10% have been observed in all performed experiments.

The output of each simulation contains the following information:

- agent trajectories during the simulation;
- speed variation for each agent;
- agent simulation time;
- total time of simulation, and
- local density per time step – we compute the local density by counting the number of agents per square meter in each context, rather than the global density (i.e., number of people divided by the building area).

The output data is stored and can be used to produce different statistical analyzes. Agent trajectories can be easily visualized with articulated virtual humans in a virtual environment in order to provide a qualitative visual validation of the simulation.

4 *CrowdSim* VALIDATION

Validation & Verification are some of the most important software development activities [3], [2]. Their purpose is to guarantee that software is built correctly. In this section we present how the validation process is performed in *CrowdSim*. We assume that *validation*, for this work, is the systematic comparison of *CrowdSim* predictions with reliable information (usually from real data analyzes). The work of Galea [8] presents a set of different validations to be performed. We focus on three of them: *Component Testing*, *Qualitative and Quantitative Validation*. Such tests are already recognized and considered in the field of safety engineering in order to validate evacuation systems[1]. In London, the International Maritime Organization (IMO) developed *guidelines for evacuation analysis for new and existing passenger ships* IMO [1] based on Galea's work. The main goal of such guides [1] is to develop a methodology for conducting an advanced evacuation analysis in order to build systems coherently able to :

- identify and eliminate congestion regions which may arise during an abandonment, due to normal movement of passengers and crew along escape routes, taking into account the possibility that crew may need to move along these routes in a direction opposite to the movement of passengers;
- demonstrate that escape arrangements are sufficiently flexible to provide the possibility that certain escape routes, assembly stations, embarkation stations or survival craft may be unavailable as a result of a casualty.

In this section we relate a set of test cases suggested by IMO in order to validate *CrowdSim* in each category. In addition, we detail how we particularly validate the software in a qualitative way.

4.1 Component Testing

Component testing is part of the normal development cycle and involves checking if the various components of the software perform as intended. This involves running the software through a battery of elementary test scenarios. In the following, we present a list of adopted component tests extracted from [1] and applied in *CrowdSim*.

4.1.1 Maintaining set walking speed

Validates the speed of a single agent when moving in a specific known environment. We built a 2m wide and 10m long corridor (illustrated in Figure 2) and simulated one agent walking from left to right with speed of 1m/s. The success criteria of this test assumes that the agent should walk 10 meters in 10 seconds.

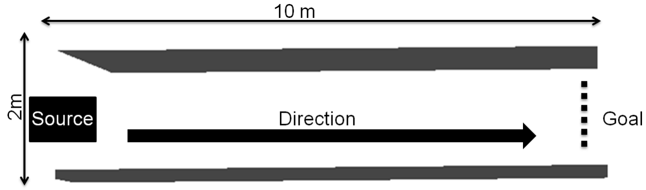


Figure 2: Environment of walking speed test.

After ten individual simulations we compute acceptable average values. The obtained average individual velocity was $1.08m/s$ with standard deviation of 0.09m/s. The average walked distance was $10.232m$ (standard deviation of 0.097m) and time of $9.506s$ (standard deviation 0.769s). According to IMO's specifications we observe that *CrowdSim* successful achieves this criteria.

4.1.2 Rounding Corners

This test evaluates the agent's ability to walk around a corner without colliding with walls and other agents. We simulated twenty people approaching a left-hand corner according to specifications illustrated in Figure 3(a).

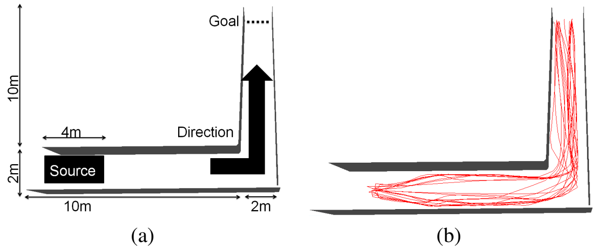


Figure 3: Setup of the experiment environment (a) and obtained trajectories of rounding corner simulation (b).

According to IMO's guidelines, this test aims to verify two specific points:

---

[1] This procedure has been highlighted in ISO document ISO/TR 13387-8:1999.

1. *The agents should successfully navigate around the corner without penetrating the boundaries.* Figure 3(b) illustrates the simulated trajectories of all twenty agents. A visual check shows that agents do not collide with the walls.

2. *The agents should successfully navigate without overlap at any time.* Figure 4 illustrates three situations for a typical simulation at different moments. While it is difficult to visually verify collision avoidance, we observe in the agents positions file (generated in the simulation) that there are no overlaps among agents (computed by their interpersonal distances).

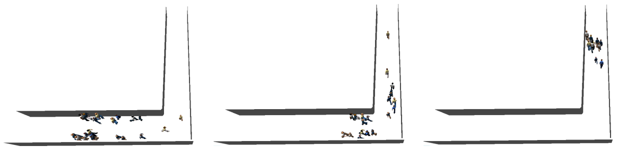

(a) time = 10 seconds (b) time = 15 seconds (c) time = 25 seconds

Figure 4: Simulation for rounding corner test.

4.2 Qualitative Validation

Qualitative Validation concerns the nature of predicted human behavior with informed expectations from observed situations. While this is only a qualitative form of verification, it is nevertheless important, as it demonstrates that the capabilities built into the model are able to produce realistic behaviors. The qualitative tests performed in order to validate the *CrowdSim* simulator are the impact of counter flow in evacuation time, crowd dissipation from a large public room, and exit route allocation. These tests are described in next sections.

4.2.1 *Counter flow* - impact in evacuation time in two rooms connected via a corridor

This test was performed according to the environment illustrated in Figure 5 populated by 100 individuals. The test was implemented in two steps as follows:

1. Agents move from room 1 to room 2, where the initial distribution is such that the space of room 1 is filled from the left with maximum density. The elapsed time until the last person enters room 2 is recorded.

2. Step one was repeated with an additional 10, 50, and 100 people in room 2. People from both rooms move simultaneously to the other room, and the time for the last person in room 1 to enter room 2 is recorded. The expected result is that the recorded time increases as the number of people in the counterflow increases.

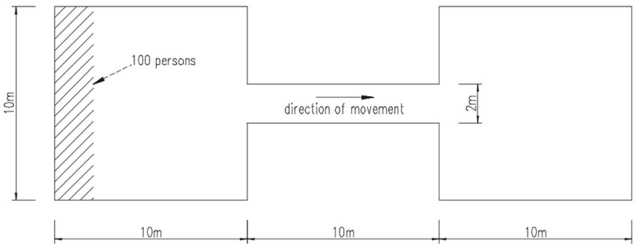


Figure 5: Counterflow scenario configuration according to IMO's specifications.

We repeated each of the scenarios described in steps 1 and 2 ten times, considering different seeds for the random number generator, which led to a test bank of 40 simulations. The expectation of increasing the time for evacuation of room 1 with the increasing number of agents in counter flow was observed, as shown in Figure 6. The graph in this figure illustrates the average time variation with the number of agents in counter flow. The black markers near to each point represent the standard deviation for the ten simulations in each case.

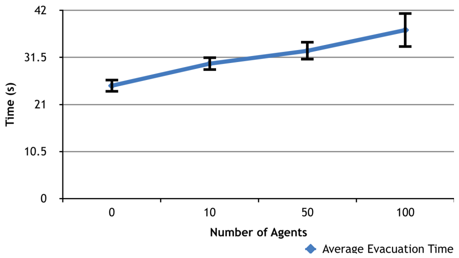


Figure 6: Average and standard deviation of time for evacuation from room 1 as a function of the number of agents in counterflow.

4.2.2 *Exit Flow* - crowd dissipation from a large public room

This test was performed in a public room populated by 1000 agents where 4 exits are available to be considered during evacuation as illustrated in Figure 7. According to IMO's instructions, the test should run according to two steps: first, simulate and record the time for the last person that leaves the room when 4 exits are available and second, the same situation but considering doors 1 and 2 as closed. The success criteria for this test is related to the amount of time for evacuation in the two cases. According to IMO, the elapsed time of the second case should be around 50 percent greater than in case 1. When such an experiment was performed with *CrowdSim*, we computed the time of 83.79s in the first case and 121.62s in the second. These values meet the requirement and, as a consequence, we can consider that *CrowdSim* is validated according to this criteria.

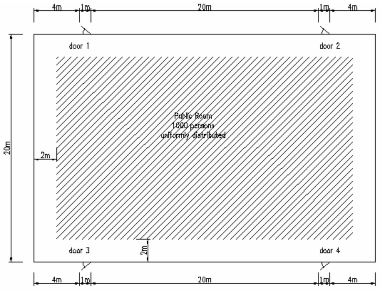


Figure 7: Exit flow scenario configuration according to IMO's specifications.

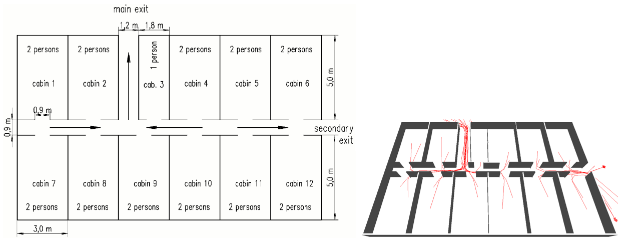
4.2.3 Exit route allocation

The IMO specification for this test has us build a cabin corridor section populated as indicated in Figure 8(a). The success criteria for the test assumes that:

1. The main exit was allocated as the goal for the people from cabins 1, 2, 3, 4, 7, 8, 9, and 10.

2. The secondary exit was allocated as the goal for all the remaining passengers.

We performed such a test in *CrowdSim* where the agents move to their assigned exits. Figure 8(b) presents the agents' trajectories in the 3D environments illustrating the success of the test.



(a) (b)

Figure 8: Exit Route Allocation: (a) IMO's specification for the test and (b) agents' performed trajectories.

4.3 Quantitative Verification

Quantitative verification involves comparing model predictions with reliable data generated from evacuation demonstrations. Galea's work [8] highlights two kinds of quantitative validation: *historic* and *prediction* based validation. In the first case, the user knows the results from previous simulations and real exercises. On the other hand, the second case refers to the usage of the model to perform predictive simulations prior to having actual experimental results. To the best of our knowledge, current IMO guidelines do not have any evaluated experimental data which would allow a thorough quantitative verification of an egress model. Therefore, in this work, we propose a method to quantitatively validate *CrowdSim*.

We propose that the quantitative validation should take into account other information besides the total evacuation time. In *CrowdSim*, such information is based on the simulation model outputs and include data about density, time, and velocities. Wherever possible, the simulations performed by *CrowdSim* have been quantitatively evaluated. Here we present a simulation performed in a night club.

5 APPLYING *CrowdSim* IN A REAL NIGHT CLUB

In this section we detail the application of *CrowdSim* to a night club. Our goal was to study how people perform an evacuation process in real life and thus obtain data to allow quantitative comparisons. The experiment was a shared experience developed in partnership with the night club owners and a safety company. On the day the experiment was conducted, the audience agreed to leave the club exactly at 2AM. Some days before the egress exercise in the club, *CrowdSim* was applied in order to provide different evacuation plans that could be used to estimate occupant behavior. The first step of the process was to reproduce the club environment in 3D. The environment has a total area of 1100 sqm and has 4 floors (see Figure 9 to see the door locations). A 3D representation is illustrated in Figure 10. In addition, the navigation graph was generated automatically based on environment geometry (see Figure 11). Notice that the graph nodes store the occupant distribution in the space, as estimated by the club owners and adjusted after the real simulation.

The safety company had generated different evacuation plans using *CrowdSim*. In Table 1 are the results obtained from three evacuation plans designed and tested in *CrowdSim* by the safety engineers. The difference among them (highlighted in Figures 12) is related to the number of people in the 3 bifurcations. Based on such distributions, the people who used the 4 different exit doors changed, as shown in the last four lines in Table 1. Indeed, it is easy to show that doors 3 and 4 received more people than the other two. This happened because doors 3 and 4 are larger and could accommodate more people.

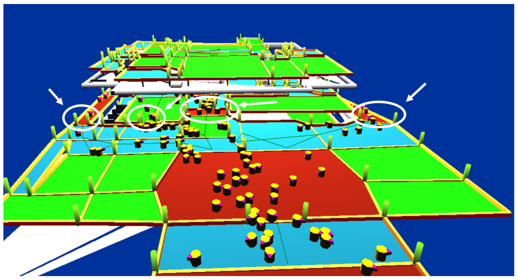


Figure 9: Images illustrating the 3D environment and doors. From left to right, the doors have IDs: 2, 3, 4 and 1.

Figure 10: Images illustrating the 3D environment and simulation.

In order to select the best plan to be executed in real life, we considered the global time (best achieved values are from $Simulation_1$ and $Simulation_2$), average time (best is from $Simulation_1$) and average density (best is from $Simulation_1$). So, based on these simple criteria, we selected $Simulation_1$ to be executed in real life.

Once the plan was selected, the safety company began to train individuals who work in the Night Club. The real evacuation was performed with 240 people who agreed to participate in the experience. During the real egress exercise, we were able to collect different data in order to evaluate results of this experience. Occupant data was obtained from security camera videos. The number of people in different parts of the club was obtained from infra-red technology. This information was very important in order to evaluate this work. Table 2 summarizes the comparison between real and virtual evacuation scenarios. Figure 13 provides an image captured during the evacuation that shows the people in stairs (2nd floor) at 40 seconds after the simulation started, and another image at the same place and time in the virtual simulation. When analyzing Table 2 there are clear differences in evacuation time. It can be explained by the fact that real people do not voluntarily behave the same as they would in a true emergency. That is, real people, not in panic, respect the space of others, and therefore do not achieve the higher densities apparent in the simulation data.

	$Simulation_1$	$Simulation_2$	$Simulation_3$
Global time (*sec*)	142	142	146
Average time (*sec*)	61	62	64
Average density ($people/m^2$)	0.1123	0.1138	0.1162
Average velocity (m/s)	0.80	0.80	0.80
Place of biggest density	2nd floor stairs	2nd floor stairs	2nd floor stairs
Time when biggest density was observed	Second 40	Second 39	Second 50
Biggest speed (m/s)	1.3	1.2	1.3
Smallest speed (m/s)	0.01	0.01	0.005
Biggest Local Density	5.4	5.4	5.0
Number of people in Door1	54	18	21
Number of people in Door2	12	41	50
Number of people in Door3	80	126	75
Number of people in Door4	100	61	100

Table 1: Quantitative data comparing simulated scenarios containing 240 people.

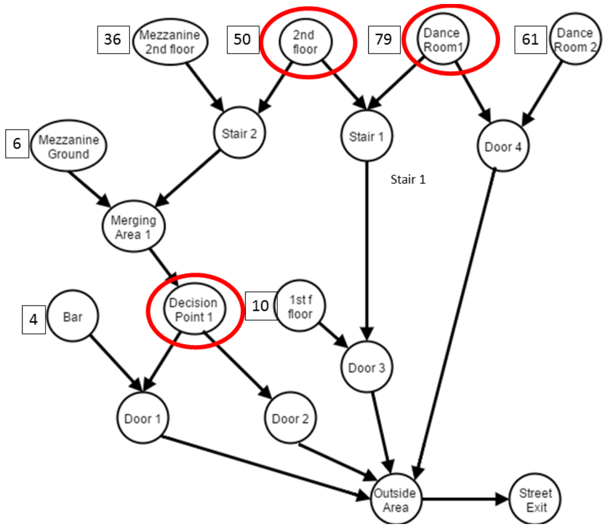


Figure 11: Navigation graph generated by *CrowdSim*.

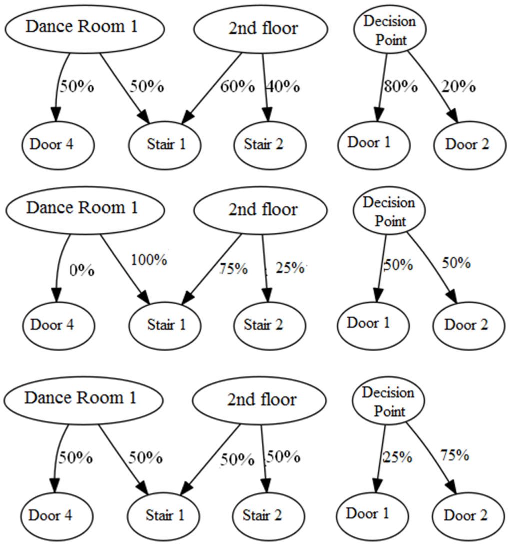


Figure 12: Three examples of evacuation plan tested by our model: Simulation IDs: 1, 2 and 3, as related in Table 1.

6 CONCLUDING REMARKS

In this paper we presented *CrowdSim*, a tool to simulate crowds in emergency situations. Our goals in this work were two-fold: to show the methodology used to validate the tool according to IMO specifications, and to include a qualitative validation based on a real life experiment. Results show that *CrowdSim* is accurate for various evaluation metrics. One important limitation is that for now, *CrowdSim* does not have functions to simulate well-structured behaviors, e.g., children evacuating from a school in a desired order. Our virtual humans act like individuals, but they cannot follow, rescue or be behind others because of this lack of structured behavior. This aspect compromises the simulation in specific environments, but we intend to address this aspect in future work.

ACKNOWLEDGEMENTS

This work was supported by Brazilian Research Agencies CNPq, CAPES and FAPERGS.

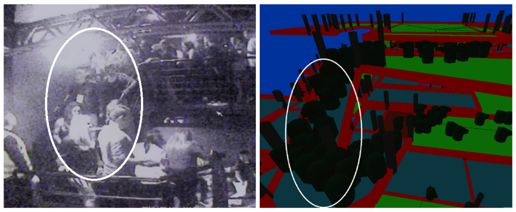


Figure 13: Images illustrating the stairs in the 2nd floor 40 seconds after the beginning of the simulation in real and virtual environments.

	Simulation$_1$	Real World Data
Total time for evacuation (seconds)	142	175
Highest Density (people/m^2)	5.4	4.5
Place of highest density	Stairs (2nd floor)	Stairs (2nd floor)
Time when highest density was observed	Second 40	Second 50
Highest speed (m/s)	1.3	1.5
Smallest speed (m/s)	0.01	0.2

Table 2: Quantitative data comparing real and simulated situations considering exactly the same evacuation plan.

REFERENCES

[1] *Guidelines for Evacuation Analysis for new and existing passenger ships*. International Maritime Organization, Marine Safety Committee, London, 2007.

[2] W. R. Adrion, M. A. Branstad, and J. C. Cherniavsky. Validation, verification, and testing of computer software. *ACM Comput. Surv.*, 14(2):159–192, June 1982.

[3] B. Beizer. *Software system testing and quality assurance*. Electrical computer science and engineering. Van Nostrand Reinhold Company, 1984.

[4] V. Cassol, C. M. Dal Bianco, A. Carvalho, J. Brasil, M. Monteiro, and S. R. Musse. An experience-based approach to simulate virtual crowd behaviors under the influence of alcohol. In *IVA'15: Proceedings of the 15th International Conference on Intelligent Virtual Agents*, Berlin, Heidelberg, 2015. Springer-Verlag.

[5] V. J. Cassol, R. A. Rodrigues, L. C. C. Carneiro, A. Silva, and S. R. Musse. Crowdsim: Uma ferramenta desenvolvida para simulação de multidões. In *I Workshop de Simulação Militar - SBGames2012*, 2012.

[6] M. L. Chu, P. Parigi, K. Law, and J.-C. Latombe. Modeling social behaviors in an evacuation simulator. *Computer Animation and Virtual Worlds*, 25(3-4):373–382, 2014.

[7] L. Fu, W. Song, W. Lv, and S. Lo. Simulation of exit selection behavior using least effort algorithm. *Transportation Research Procedia*, 2(0):533 – 540, 2014. The Conference on Pedestrian and Evacuation Dynamics 2014 (PED 2014), 22-24 October 2014, Delft, The Netherlands.

[8] E. R. Galea. A general approach to validating evacuation models with an application to EXODUS. *Journal of Fire Sciences*, 16(6):414–436, 1998.

[9] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *SIGART Bull.*, (37):28–29, 1972.

[10] P. Huang, J. Kang, J. T. Kider, B. Sunshine-Hill, J. B. McCaffrey, D. V. Rios, and N. I. Badler. Real-Time Evacuation Simulation in Mine Interior Model of Smoke and Action. 2010.

[11] E. Kuligowski and S. Gwynne. What a user should know when selecting an evacuation model. *Fire Protection Engeneering*, pages 600–611, Oct. 2005.

[12] N. Pelechano, J. Allbeck, and N. I. Badler. *Virtual Crowds: Steps Toward Behavioral Realism*. Morgan Claypool Publishers, 2015.

[13] D. Thalmann and S. R. Musse. *Crowd Simulation - Second Edition*. Springer-Verlag London Ltd, 2013.