COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2015; **26**:377–386
Published online 29 April 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cav.1652

SPECIAL ISSUE PAPER

Environment optimization for crowd evacuation

Glen Berseth[1]*, Muhammad Usman[2], Brandon Haworth[2], Mubbasir Kapadia[3] and Petros Faloutsos[2]

[1] University of British Columbia, Vancouver, British Columbia, Canada
[2] York University, Toronto, Ontario, Canada
[3] Rutgers University, New Brunswick, NJ, USA

ABSTRACT

The layout of a building, real or virtual, affects the flow patterns of its intended users. It is well established, for example, that the placement of pillars at proper locations can often facilitate pedestrian flow during the evacuation of a building. Such considerations are therefore important for architects, game level developers, and others whose domains involve agents navigating through buildings. In this paper, we take the first steps towards developing a simulation framework that can be used to study the optimal placement of architectural elements, such as pillars or doors, for the purposes of facilitating dense pedestrian flow during the evacuation of a building. In particular, we show that the steering algorithms used to model the local navigation abilities of the agents significantly affect the results, which motivates the need for a statistically valid approach and further study. Copyright © 2015 John Wiley & Sons, Ltd.

KEYWORDS
crowd simulation; optimization and analysis

***Correspondence**
Glen Berseth, Computer Science, University of British Columbia, Vancouver, Canada.
E-mail: gberseth@cs.ubc.ca

1. INTRODUCTION

Architects produce functional pieces of art through the planning, design, and construction of buildings. Humans and, in the near future, robots explore, interact with, and engage these environments and those who participate in them. It is of interest to the architecture, robotics, urban simulation, game development, and other communities to explore the configuration space of environmental elements for a variety of reasons and applications. For example, the elements that are present in an environment affect the flow of people during an emergency evacuation.

It is generally impractical for human experts to exhaustively search the entire space of environment configurations in order to select an environment layout that meets application-specific criteria. Thus, there is a growing practical relevance for computational approaches that can comprehensively analyze the space of all possible configurations of an environment and automatically derive optimal designs that satisfy user-defined objectives.

In this paper, we propose a computational framework for studying the configuration of architectural elements such as pillars or doors for optimizing dense pedestrian flow during building evacuations. Our goal is to understand the complexity of the problem by systematically evaluating variations of key factors in the simulation pipeline that influence the results.

In particular, we study the effect of local collision avoidance strategies (steering) on crowd flow patterns on representative evacuation scenarios. We use three different steering algorithms for this study, which include physically based models and optimal reciprocal methods. The scenarios include variations on the number and placement of pillars, exit door sizes, and corridor and crowd flow configurations (unidirectional, bidirectional, and four-directional flows were studied).

Our findings reveal that the choice of steering algorithm significantly affects the results. We also observe that placing one to four pillars often improves the flow of the crowd as is shown in Figure 1. In summary, our observations prompt the need for a statistically valid approach and further study.

The rest of the paper is organized as follows. Section 2 provides a brief overview of related work. Section 3 describes how we define and parameterize our domain.

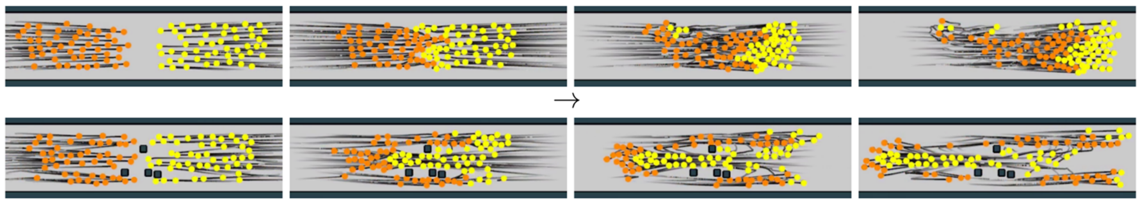


Figure 1. Snapshots in raster order of two similar scenarios where two groups of agents travel in opposite directions in a hallway. Placing four pillars at optimal locations (bottom) improves on average the flow for both groups compared with the case without pillars (top).

In Section 4, we present our optimization formulation, we describe our scenarios in detail in Section 5, and in Section 6, we discuss our results.

2. RELATED WORK

Crowd evaluation. There has been a growing recent trend using statistical investigation in the evaluation and analysis of crowd simulations. [1] adopts a data-driven approach of evaluating crowds by measuring its similarity to real-world data. [2,3] proposes a compact suite of manually defined test cases that represent different steering challenges and a rich set of derived metrics that provide an empirical measure of the performance of an algorithm. Recent extensions such as [4] propose a representative sampling of challenging scenarios that agents encounter in crowds to compute the coverage of the algorithm and the quality of the simulations produced. Density measures [1] and fundamental diagram-based comparisons [5] use aggregate metrics for quantifying similarity. The work in [6,7] measures the ability of a steering algorithm to emulate the behavior of a real crowd dataset by measuring its divergence from ground truth. [8] presents a histogram-based technique to quantify the global flow characteristics of crowds. Perceptual studies rely on human factors experiments to measure the variety in appearance and motion [9] or perceptual fidelity of relaxing collisions [10] in crowds.

Optimizing crowd simulation parameters. Researchers [11,12] have observed that the selection of a steering algorithm's parameters can dramatically influence the performance and behavioral patterns of the aggregate crowd dynamics. The work in [13] proposes solutions for automatically fitting a steering algorithm's parameters to minimize collisions, minimize evacuation times, or match recorded data. However, there is little work that studies the impact of environmental parameters on crowd flow patterns, which is the main focus of this work.

Game level optimization. The work in [14,15] use evolutionary approaches for procedural level creation and the placement of game level design elements. [16] optimizes platformer games to maximize "fun." [13] proposes a new method to improve the behavior of crowd simulations using combinations of objectives. [17] uses optimization to find interesting variations in the game level, while [18] searched for optimal play space configurations for platformer games. [19] formulates a parameterization of the game level and evaluates each game level's expected difficulty.

Architectural optimization. The work in [20] studies the optimal placement of pillars for specific evacuation scenarios using a single steering algorithm. In contrast, we take a step back and systematically observe the sensitivity of steering algorithms and its parameters and the parameters of environment elements (e.g., pillar shape) on the optimization results as well as evaluating additional environments.

3. SCENARIO CONFIGURATION SPACE

A scenario s is a specific configuration of obstacles and agents in an environment. A scenario may refer to the starting layout of obstacles and agents or to an intermediate snapshot of a dynamic simulation. More formally, we define a scenario, similar to [21], as $s = \langle \mathbf{O}, \mathbf{A} \rangle$, where $\mathbf{O}, \mathbf{A}$ are the sets of static obstacles and agents in the scenario. An obstacle $o \in \mathbf{O}$ at a particular position in the environment ($\mathbf{x}_o$) can be either a rectangular bounding box or a cylindrical pillar. An agent $a \in \mathbf{A}$ is defined as $a = \langle \mathbf{x}, r, \mathbf{g} \rangle$, where $\mathbf{x}$ is the current position, r is the collision radius, and $\mathbf{g}$ is the goal position of the agent.

Allowing a few or all of the parameters of a scenario, s, to range between finite and infinite bounds defines a configuration space for the scenario that we refer to as scenario subspace, $\mathscr{S}_{sub}$, from which we can draw arbitrary samples. In this work, we focus on the crowd flow of agents through corridors or during a building evacuation, and we construct our subspaces as follows. We first define a set of agent regions, within which we can randomly distribute agents, usually based on uniform sampling. The goal region and/or the desired velocity of each agent is set to a region or a direction that affects desired interactions with the other group(s) of agents and obstacles of interest. An obstacle region is placed at a location of interest. For example, we often set obstacle regions near doors or in the middle section of a corridor. Specific examples can be seen in Sections 5 and 6. For notation reasons, we indicate the parameters that the optimization can change to improve an objective as $\mathbf{p}$ and the associated bounds (constraints) on these parameters as $\mathscr{P}$.

3.1. Crowd Flow

There are a number of proposed measures to characterize the flow of a crowd [22,23]. We define crowd flow as the ratio of the number of agents that successfully reached their destination $|A_c|$ to the average agent completion time t_{avg}. Crowd flow for a specific parameterization of a scenario is computed as follows:

$$f(\mathbf{p}) = \frac{|A_c|}{t_{avg}}, t_{avg} = \frac{\sum_{a \in A} t_a}{|A|} \tag{1}$$

where t_a is the simulated time that the agent a needed to complete the simulation; otherwise, a scenario-specific upper limit A is the set of all agents and $|A|$ indicates the cardinality of set A. The parameters $\mathbf{p}$ are used to construct the scenario configuration, which affects the simulation time t_a of each agent. An agent has completed a simulation if the agent reaches its target location before the simulation terminates; A_c is the set of completed agents.

Given a reference or default parameterization $\mathbf{p}_d$ of the subspace, we can define the relative crowd flow as follows:

$$f_r(\mathbf{p}) = f(\mathbf{p}) - f(\mathbf{p}_d) \tag{2}$$

The sign of the relative flow reveals immediately if parameterization $\mathbf{p}$ improves crowd flow over the reference one. For our experiments, the reference flow corresponds to the case where no pillars are present.

4. OPTIMIZATION FORMULATION

Given a scenario subspace, $\mathscr{S}_{sub}$, a set of free parameters, $\mathbf{p}$, and their bounds (constraints), $\mathscr{P}$, we set up and solve a minimization problem to fit the parameters to an objective as follows:

$$\mathbf{p}^* = \arg\min_{\mathbf{p} \in \mathscr{P}} (-f_r(\mathbf{p}) + g(\mathbf{p})) \tag{3}$$

4.1. Objective Function

Our objective formulation consists of two terms: the opposite of the relative flow term, $f_r(\mathbf{p})$, defined in Equation 2, and a penalty term, $g\mathbf{p}$, which penalizes the violation of the constraints on the parameters. The reason for including a penalty term is that the method used to solve the problem, like many optimization methods, prefers constraints modeled using penalty terms rather than hard constraints. Intuitively, penalty methods allow the optimization process to compute smoother derivatives, which often improves the rate of convergence. For independent scalar parameters, the penalty function(s) can be formulated by the optimization method directly. However, in our case, the parameter vector contains the location of multiple obstacles, so we need to explicitly enforce non-overlapping constraints in the placement of obstacles.

Overlap penalty term. Let $ov(o_1 o_2)$ be the area of the overlapping regions of two obstacles, o_1, o_2, or zero if the obstacles do not overlap. We define a penalty term for all pairs of overlapping obstacles as follows:

$$g(\mathbf{p}) = \sum_{\forall (o_1,o_2) \in \mathbf{O} \times \mathbf{O}} g_{ov}(o_1, o_2) \tag{4}$$

where

$$g_{ov}(o_1, o_2) = (ov(o_1, o_2) + 1)(1 - f_r(\mathbf{p})))^2 \tag{5}$$

for distinct obstacles, o_1, o_2, whose overlapping area, $ov(o_1, o_2)$, is non-zero and zero in all other cases.

4.2. The Covariance Matrix Adaptation Evolution Strategy Algorithm

For most scenarios of interest, our minimization formulation results in a non-convex problem that we solve with the covariance matrix adaptation evolutionary strategy [24]. The covariance matrix adaptation evolution strategy algorithm is well suited to this domain for many reasons: it is straightforward to implement, it can handle ill-conditioned objectives with noise, and it is very competitive in converging to an optimal value in few iterations.

4.3. The Effect of Global Navigation

The aggregate dynamics of simulated crowds are governed by decision making at two levels: (i) global navigation: selecting the next target location in space to steer towards and (ii) local collision avoidance: steering towards the next target while avoiding obstacles and other moving agents.

The global navigation decision dictates local collision avoidance targets, producing a significant impact on the resulting flow patterns produced depending on the decisions that agents make. There are many computational methods for solving the global navigation problem including grid, interval, mesh, field, and sampling-based approaches, each with their own variations and parameters suited for different applications.

To demonstrate this effect, we used a grid-based search method for global navigation. We observe the following artifacts: (i) jarring discontinuities in crowd flow due to the discretization of the environment into spatial grid cells and (ii) crowd congestion because the static navigation strategy does not account for dynamic agents while selecting a navigation decision. These artifacts dilute the effects of the actual steering strategy used.

In order to study the impact of local collision avoidance strategies on crowd flow, we mitigate/nullify the impact of global navigation on flow by selecting the navigation decision as the desired goal location as provided by the scenario definition. Hence, the aggregate dynamics of the crowd is dictated purely by the steering algorithm.

A systematic study of the effect of global navigation strategies on crowd flow and the interrelation between local and global strategies is the subject of future work.

5. METHODOLOGY

In this section, we describe the steering algorithms, the specific scenarios, and the environmental features that we consider in our study. For all experiments, agents are represented by disks with a radius of 0.2286 m. We use this radius as it gives us more realistic results and has no negative effect on our simulation system.

5.1. Steering Algorithms

To study the effect of steering algorithms on the results, we chose the following three established steering algorithms that represent a range of different steering approaches: (i) ORCA: an efficient and widely used technique that uses reciprocal velocity obstacles for collision avoidance [25], (ii) PPR: a hybrid approach that uses rules to combine reactions, predictions, and planning [26], and (iii) SF: a variant of the social forces method for crowd simulation [27]. For each algorithm, we use the default parameters that are suggested by the algorithm's developers.

5.2. Benchmarks

We study crowd flow patterns using the aforementioned steering algorithms on a variety of scenarios that exercise unidirectional, bidirectional, and four-directional flows, using different configurations of corridors, pillars, and exit doors.

5.2.1. Unidirectional Hallway.

The configuration of this scenario is shown in Figure 2. A hundred agents are randomly placed in a 12.5×4 m^2 region (blue). Up to four pillars are placed in the optimization region (gray). Each agent has a target location in the goal region (green) outside of the hallway. The distance between the closed boundaries of the optimization and the crowd regions is 3.5 m.

5.2.2. Bidirectional Hallway.

This scenario is an extension of the previous one with two groups of agents, A and B, traveling in opposite directions in the hallway (Figure 3). Each group contains 50 agents that are randomly placed in the corresponding blue region of size 6.25×4 m^2. Up to four pillars are placed in the 4×4 m^2 optimization region (black). Each group must cross the optimization region to reach its corresponding target region (green).

5.2.3. Two-way Egress.

In this scenario, two groups of agents traveling from opposite directions in a hallway must exit the same door in the middle of the hallway (Figure 4). The arrangement of the agents and the optimization region are identical to those of the previous scenarios. A door of size 1.3716 m is in the middle of the 34-m hallway. The size of the door is in accordance with local standard building codes.

5.2.4. Four-way Hallway.

This is an extension of the previous bidirectional hallway scenario in all cardinal directions (Figure 5). Four groups of 25 agents each travel from opposite directions in two hallways that share a 4×4 m^2 optimization region (gray) in the center. The agents are randomly distributed in their corresponding region (blue) and must reach their target region (green) across the corresponding hallway.

These scenarios are selected based on the most common flow scenarios studied and cover many real-world situation. Although we do not rigorously evaluate our method's ability to generalize to any scenario, we suspect that the method has this property. This can be attributed to the optimization algorithm's robustness.

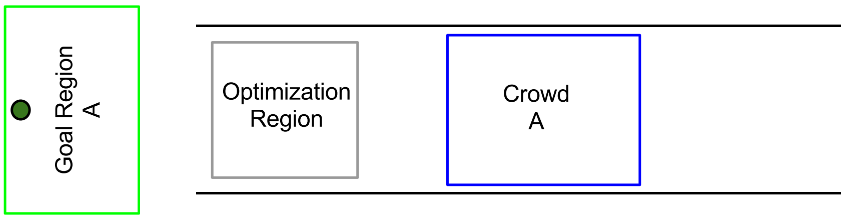


Figure 2. Unidirectional hallway scenario.

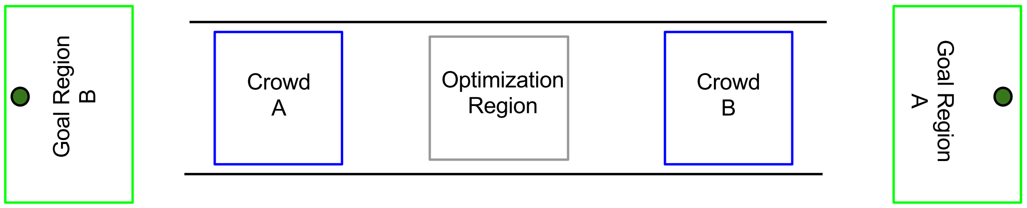


Figure 3. Bidirectional hallway scenario.

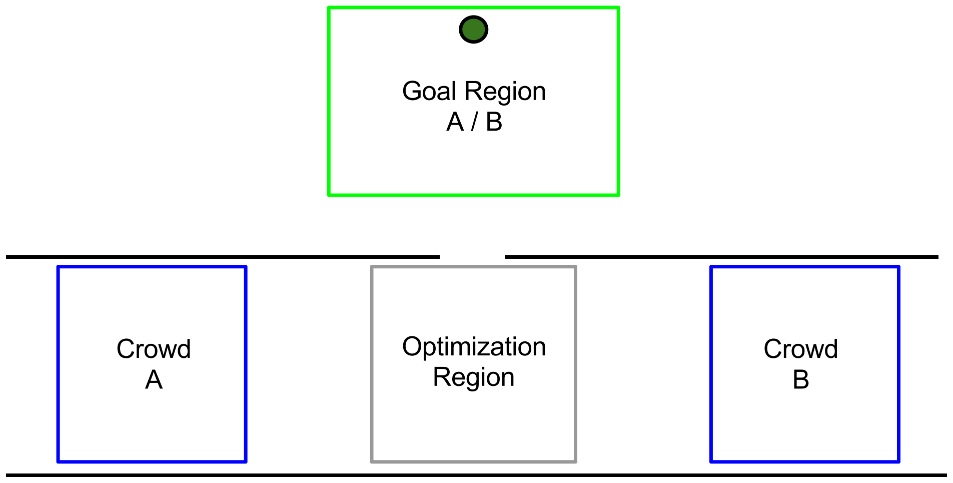


Figure 4. Two-way egress scenario.

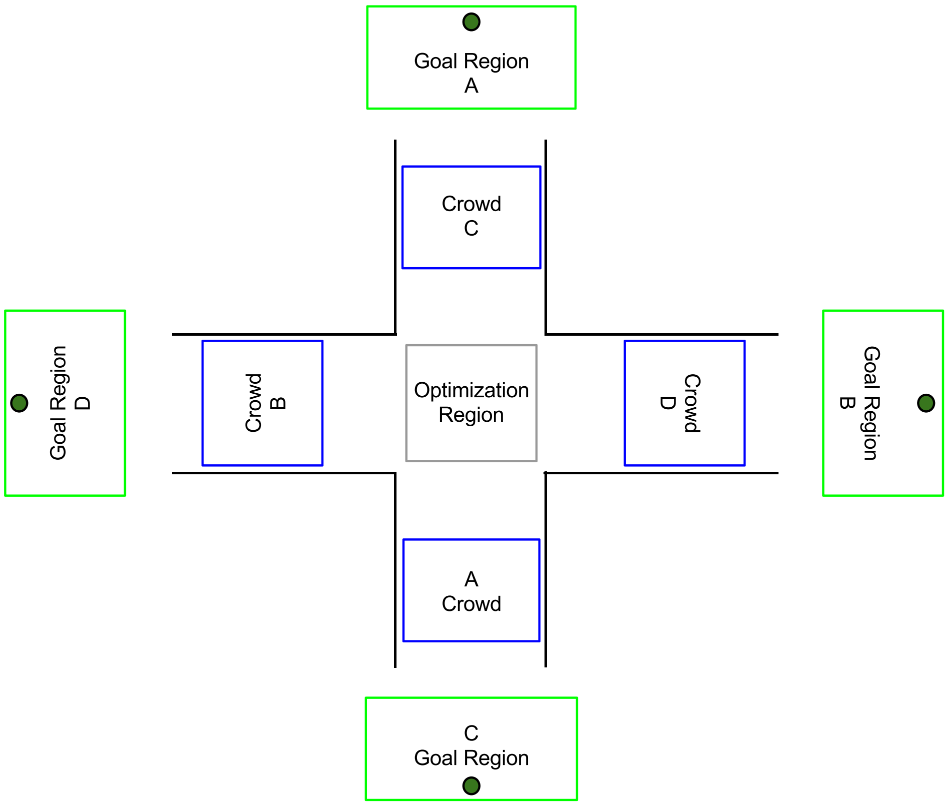


Figure 5. Four-way hallway scenario.

6. EXPERIMENTS

We apply our methodology to variations of the scenarios described in the previous section and discuss the results. As a proof of concept and to motivate the rest of the experiments, we first present an exhaustive characterization of the optimization landscape for a single scenario.

6.1. Characterizing a Scenario Subspace

For this experiment, we uniformly sample the unidirectional hallway subspace with a single pillar for all steering algorithms. The optimization region in Figure 2 is uniformly sampled at intervals of 2.5 cm, which produces 25 600 sample locations for the pillar. Figure 6 shows

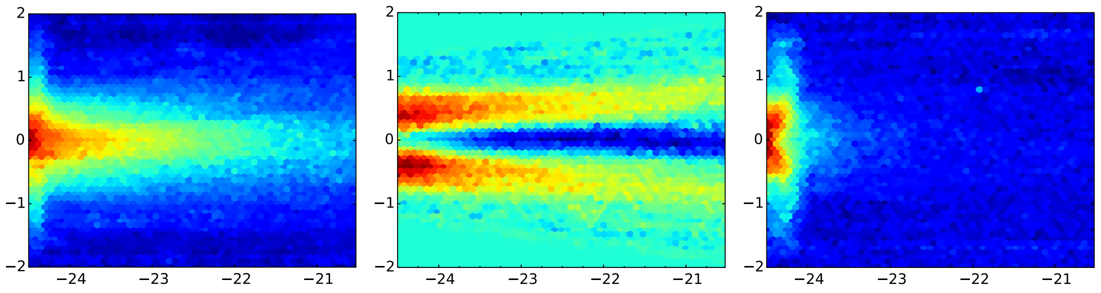


Figure 6. Uniform sampling of the optimization region for the unidirectional hallway scenario, where blue indicates better flow. Clearly, the effect of the pillar differs per algorithm.

the flow relative to the case where no pillars are present (inverse of Equation 2 in the form of heat maps for all three algorithms). Blue and red correspond to high and low values of the relative flow, respectively.

It is compelling to see the significant difference in the optimization landscape for the three algorithms. ORCA prefers the obstacle out of the way of the exit, while PPR clearly benefits from having an obstacle in the middle of the optimization region. It is also evident that SF has the largest blue region. In some sense, SF is the least sensitive to an obstacle that is not near the exit.

We also found some complex behavior with respect to pillar geometry. The ORCA algorithm had poor behavior for circular pillars, treating the pillar as if it was an agent, significantly impeding the crowd flow. However, the SF algorithm functioned much smoother with round pillars. The use of round pillars for the SF algorithm leads to tangential forces that help agents slide around pillars, and the axis-aligned boxes did not have this property. Lastly, the PPR algorithm was indifferent to the pillar geometry.

6.2. Unidirectional Hallway

Table I shows the crowd flow as defined in Equation 1 for the optimal placement of one to four pillars for all scenarios. The table includes the case of zero pillars for reference. Looking at the section of the table that corresponds to the unidirectional hallway scenario, we can see that for all steering algorithms, the crowd flow improves with the placement of pillars. Notably, three pillars produce the highest crowd flow for all three algorithms.

6.3. Bidirectional Hallway

Table I shows that for this scenario in almost all cases, the crowd flow improves with the placement of pillars. ORCA and SF show improved flow even with four pillars. In fact, SF achieves the best flow with four pillars. PPR shows approximately 30% improvement with the optimal placement of two pillars.

6.4. Two-way Egress

Table I shows that only ORCA benefits significantly from the placement of obstacles in the scenario. The largest improvement for ORCA is in the case of four pillars, for which crowd flow surprisingly more than doubles. SF benefits in all cases but only marginally, with the largest benefit in the case of two pillars. PPR had difficulties completing this scenario realistically without global planning.

6.5. Four-way Hallway

Table I shows that PPR has a very difficult time with this scenario. This is probably because the algorithm tends to make agents wait when they are unable to move in a range of forward directions. Because of this behavior, the four groups seem to reach a deadlock in the middle of the hallways. On the other hand, ORCA and SF both show

Table I. The optimal crowd flow values, $f(\text{p})$, for all experiments, where $n-p$ means n pillars.

Algorithm	$0-p$	$1-p$	$2-p$	$3-p$	$4-p$	Algorithm	$0-p$	$1-p$	$2-p$	$3-p$	$4-p$
		Unidirectional hallway						Two-way egress			
ORCA	6.12	6.61	6.60	6.63	6.62	ORCA	0.80	1.53	1.59	1.71	1.93
PPR	1.92	2.18	2.17	2.19	2.15	PPR			N/A		
SF	4.43	4.48	4.51	4.89	4.57	SF	4.29	4.62	4.36	4.48	4.36
		Bidirectional hallway						Four-way hallway			
ORCA	2.84	3.64	3.70	3.54	3.63	ORCA	3.01	3.79	3.64	3.51	3.63
PPR	1.69	2.16	2.22	2.09	2.11	PPR			N/A		
SF	3.34	3.80	3.57	3.65	3.93	SF	3.48	3.76	3.84	3.93	3.76

improved crowd flow with the addition of pillars. ORCA seems to perform better with one optimally placed pillar while SF with three.

6.6. Flow Rate Optimization

We found that the rate of convergence depended not only on the steering algorithm but also on the scenario. Convergence was the fastest for the unidirectional hallway scenario and the slowest for the two-way egress. This may be due to the change in flow direction that the crowds need to make, which is not present in the other scenarios. We show the optimization process for the SF algorithm on the four-way hallway scenario with two pillars in Figure 7. Comparing the convergence rates for each of the steering algorithms shows that no particular algorithm is easier or harder to optimize for all scenarios.

The placement of additional pillars followed two patterns. In the case when crowd flow was unidirectional or bidirectional, additional pillars would be aligned with the crowd flow. In other cases, the pillars were configured in arrangements that avoid alignment with the crowd flow, such as diagonal to crowd flows or in a triangular pattern.

6.7. Agent Completion Histogram

It is valuable to examine the agent completion rate over time. Figure 8 presents a histogram analysis for the three steering algorithms and for the optimal placement of zero to four pillars for three different scenarios. Each histogram shows the number of agents that reached their goal within uniformly spaced windows of time. The first row corresponds to ORCA and the unidirectional hallway scenario. Of particular interest is the case of three pillars, which seems to produce the best completion rate for all cases where pillars are used. The middle row shows the results for SF and for the four-way hallway scenario. The agent completion histogram for SF appears consistent for any number of pillars. The bottom row shows the histogram for PPR and the bidirectional hallway scenario. In this case, it suggests that the results change significantly with the number of pillars.

6.8. Varying Door Size

To study the effect of doorway size on crowd flow, we modify the two scenarios that include doors and experiment with door openings that are 1.5× and 2× the original size.

As expected, increasing the doorway size does improve the crowd flow for each of the steering algorithms (Figure 9). The improvement is significantly larger for the two-way egress scenario than for the unidirectional hallway scenario. SF surpasses ORCA in crowd flow for the largest doorway size in the unidirectional hallway scenario. For the two-way egress example, the crowd flow almost doubles for SF and almost triples for ORCA. The significant effect of adding just a single pillar for the ORCA algorithm can also be seen from these data. What we find from this is that the overall increase in flow is highly dependent on the arrangement of the door opening. It is more crucial to increase door openings that are perpendicular to the crowd flow.

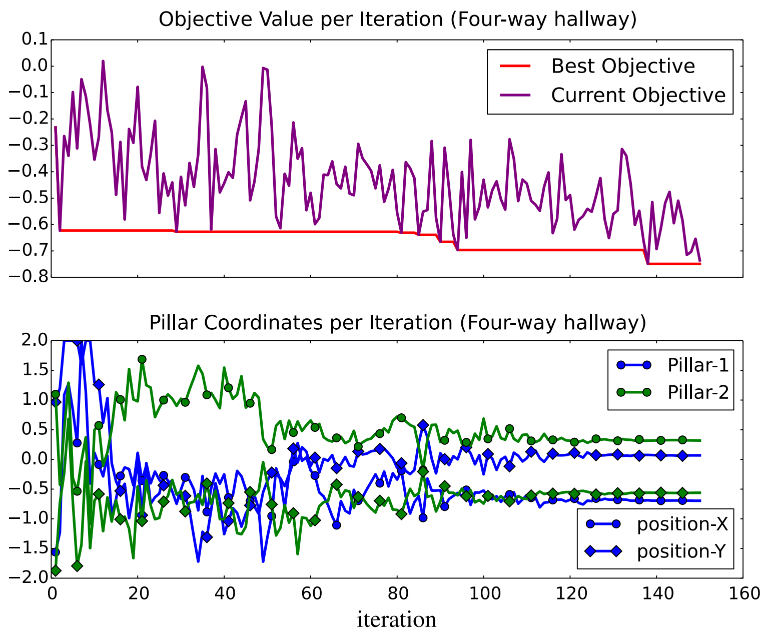


Figure 7. A visualization of the parameter selection process while optimizing the SF steering algorithm for the four-way hallway example with two pillars.

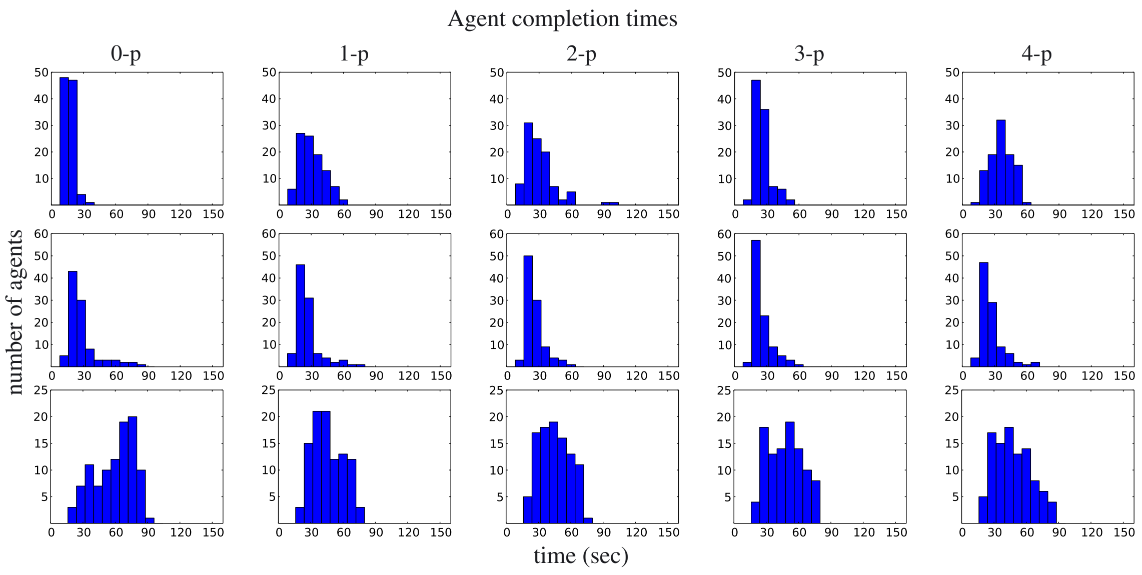


Figure 8. Agent completion histogram versus time. Top: ORCA for the unidirectional hallway scenario. Middle: SF for the four-way hallway scenario. Bottom: PPR for the bidirectional hallway scenario.

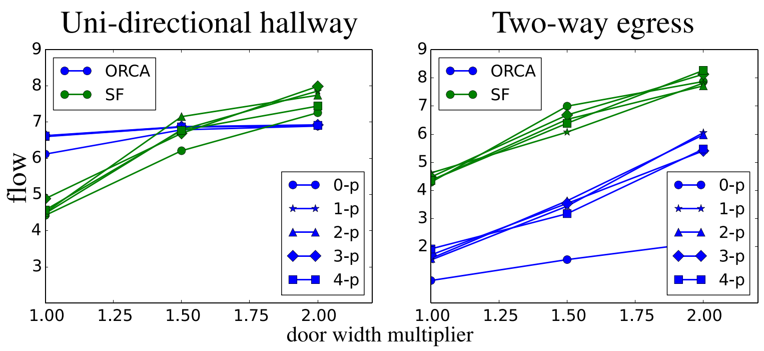


Figure 9. Optimal crowd flow for each algorithm for a few of the scenarios with larger size doors. Increasing door width generally increases crowd flow.

7. CONCLUSION

We have presented a methodology to systematically study how the configuration of an environment impacts crowd flow. Our results reveal several interesting insights, highlighting the sensitivity of optimal environment configurations on the choice of steering algorithm, as well as the shape and number of environment elements such as pillars. We observe that in a majority of scenarios, the optimal number of pillars was found to be three. Door widths had a significant impact on crowd flow patterns, especially for bidirectional traffic, which highlights the importance of selecting the right door width depending on the expected crowd interactions.

Limitations and future work. Our analysis is limited to homogeneous crowds and three specific algorithms. It would be interesting to extend the analysis to include other steering methods, more complex agent models [28,29], and agents with diverse characteristics and behaviors [30].

It is of particular consequence that the optimal placement of obstacles varies with the steering algorithm. For applications involving virtual humans, the environment can be optimized for the steering algorithm used. For applications involving the design of real buildings, however, the community needs to first establish an algorithm that models the steering behavior of real humans. This is an important open question and the subject of future work. We believe that this initial study motivates the need for further, larger scale research in the domain of environment optimization for crowd simulation.

REFERENCES

1. Lerner A, Chrysanthou Y, Shamir A, Cohen-Or D. Context-dependent crowd evaluation. *Computer Graphics Forum* 2010; **29**(7): 2197–2206.
2. Singh S, Naik M, Kapadia M, Faloutsos P, Reinman G. Watch out! A framework for evaluating steering

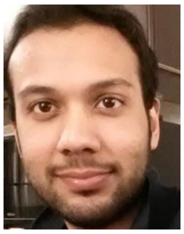
behaviors. In *Motion in Games, First International Workshop*, Utrecht, The Netherlands, 2008; 200–209.

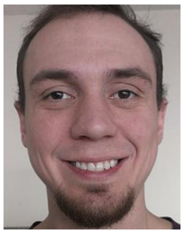
3. Singh S, Kapadia M, Faloutsos P, Reinman G. Steerbench: a benchmark suite for evaluating steering behaviors. *Computer Animation And Virtual Worlds* 2009; **20**(February): 533–548.
4. Kapadia M, Wang M, Singh S, Reinman G, Faloutsos P. Scenario space: characterizing coverage, quality, and failure of steering algorithms. In *Proceedings of ACM SIGGRAPH/EG SCA*, Vancouver, British Columbia, Canada, 2011; 53–62.
5. Seyfried A, Boltes M, Kähler J, *et al.* Enhanced empirical data for the fundamental diagram and the flow through bottlenecks. In *Pedestrian and Evacuation Dynamics 2008*. Springer: Berlin Heidelberg, 2010; 145–156.
6. Guy SJ, van den Berg J, Liu W, Lau R, Lin MC, Manocha D. A statistical similarity measure for aggregate crowd dynamics. *ACM Transactions on Graphics* 2012; **31**(6): pp. 11.
7. Pettré J, Ondřej J, Olivier A, Cretual A, Donikian S. Experiment-based modeling, simulation and validation of interactions between virtual walkers. In *ACM SIGGRAPH/EG SCA*, New Orleans, Louisiana, 2009; 189–198.
8. Musse SR, Cassol VJ, Jung CR. Towards a quantitative approach for comparing crowds. *Computer Animation and Virtual Worlds* 2012; **23**(1): 49–57.
9. McDonnell R, Larkin M, Dobbyn S, Collins S, O'Sullivan C. Clone attack! Perception of crowd variety. *ACM Transactions on Graphics* 2008; **27**(3): 26:1–26:8.
10. Kulpa R, Olivierxs A-H, Ondřej J, Pettré J. Imperceptible relaxation of collision avoidance constraints in virtual crowds. In *ACM SIGGRAPH ASIA*, 2011; 138:1–138:10.
11. Davidich M, Koester G. Towards automatic and robust adjustment of human behavioral parameters in a pedestrian stream model to measured data. In *Pedestrian and Evacuation Dynamics*. Springer: US, 2011; 537–546.
12. Wolinski D, Guy S, Olivier A-H, Lin M, Manocha D, Pettré J. Parameter estimation and comparative evaluation of crowd simulations. In *Eurographics*, Strasbourg ,France, 2014; 303–312.
13. Berseth G, Kapadia M, Haworth B, Faloutsos P. SteerFit: automated parameter fitting for steering algorithms, 2014. In *Proceedings of ACM SIGGRAPH/EG SCA*, Koltun V, Sifakis E (eds): The Eurographics Association, Geneva, Switzerland; 113–122.
14. Cardamone L, Yannakakis GN, Togelius J, Lanzi PL. Evolving interesting maps for a first person shooter. In *Applications of Evolutionary Computation*. Springer: Berlin Heidelberg, 2011; 63–72.
15. Sorenson N, Pasquier P. Towards a generic framework for automated video game level creation. In *Proceedings of ICCCX - Volume Part I*. Springer-Verlag, 2010; 131–140.
16. Sorenson N, Pasquier P. The evolution of fun: automatic level design through challenge modeling. In *Proceedings of ACM ICCCX*, Lisbon, Portugal, 2010; 258–267.
17. Shi Y, Crawfis R. Optimal cover placement against static enemy positions. In *FDG*, Chania, Crete, Greece, 2013; 109–116.
18. Bauer AW, Cooper S, Popovic Z. Automated redesign of local playspace properties. In *FDG*, Chania, Crete, Greece, 2013; 190–197.
19. Berseth G, Haworth BM, Kapadia M, Faloutsos P. Characterizing and optimizing game level difficulty. In *Proceedings of Motion on Games,* ACM, Playa Vista, California, 2014; 153–160.
20. Rodriguez S, Zhang Y, Gans N, Amato NM. Optimizing aspects of pedestrian traffic in building designs. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* IEEE, Tokyo, Japan, 2013; 1327–1334.
21. Berseth G, Kapadia M, Faloutsos P. Steerplex: estimating scenario complexity for simulated crowds. In *Proceedings of Motion on Games,* MIG '13, ACM, New York, NY, USA, 2013; 45:67–45:76.
22. Johansson A, Helbing D, A-Abideen HZ, Al-Bosta S. From crowd dynamics to crowd safety: a video-based analysis. *Advances in Complex Systems* 2008; **11**(4). arXiv:0810.4590.
23. Helbing D, Johansson A, Al-Abideen HZ. Dynamics of crowd disasters: an empirical study. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* 2007; **75**(4): 046109.
24. Hansen N. A CMA-ES for mixed-integer nonlinear optimization. *Technical Reports RR-7751*, INRIA, Le Chesnay Cedex France, 2011.
25. van den Berg J, Guy SJ, Lin M, Manocha D. Reciprocal n-body collision avoidance. In *Robotics Research*, Vol. 70. Springer: Berlin Heidelberg, 2011; 3–19.
26. Singh S, Kapadia M, Hewlett B, Reinman G, Faloutsos P. A modular framework for adaptive agent-based steering. In *ACM SIGGRAPH 3D*, San Francisco, California, 2011; 141–150.
27. Helbing D, Farkas I, Vicsek T. Simulating dynamical features of escape panic. *Nature* 2000; **407**(6803): 487–490.

28. Singh S, Kapadia M, Reinman G, Faloutsos P. Footstep navigation for dynamic crowds. *Computer Animation and Virtual Worlds* 2011; **22**(2–3): 151–158.
29. Kapadia M, Beacco A, Garcia F, Reddy V, Pelechano N, Badler NI. Multi-domain real-time planning in dynamic environments. In *ACM SIGGRAPH/Eurographics SCA*, Anaheim, California, 2013; 115–124.
30. Schuerman M, Singh S, Kapadia M, Faloutsos P. Situation agents: agent-based externalized steering logic. *Computer Animation and Virtual Worlds* 2010; **21**: 267–276.

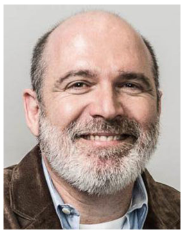
AUTHORS' BIOGRAPHIES

Glen Berseth is a PhD student in the Department of Computer Science at the University of British Columbia. He received the BSc degree in Computer Science from York University in 2012 and his MSc from York University under the supervision of Petros Faloutsos in 2014. His research interests include character animation, crowd simulation, machine learning, and cognitive agents.


Muhammad Usman is a Masters student in the Electrical Engineering and Computer Science Department at York University. Previously, he was working with TkXel as a Software Engineer. He has also worked with Mentor Graphics Corporation as an Internee. Usman's research interest areas are crowd simulation, crowd steering behavior, and design architecture optimizations.


Brandon Haworth is a PhD student in the Department of Electrical Engineering and Computer Science at York University. He received the BSc degree in Computer Science from York University in 2013. His research interests include crowd steering behaviors, crowd simulation, architectural optimization, assistive technologies, rehabilitative technologies, digital game design, and serious games.

Mubbasir Kapadia is an Assistant Professor in the Computer Science Department at Rutgers University. Previously, he was an Associate Research Scientist at Disney Research Zurich. Kapadia's research aims to develop integrated solutions for full-body character animation, planning-based control, behavior authoring, and statistical analysis of autonomous virtual human simulations. http://www.cs.rutgers.edu/~mubbasir/


Petros Faloutsos is an associate professor at the Department of Computer Science and Engineering at York University. Before joining York, he was a faculty member at the Computer Science Department at the University of California at Los Angeles, where in 2002, he founded the first computer graphics lab at UCLA, called M.A.Gix. Faloutsos received his PhD degree (2002) and his MSc degree in Computer Science from the University of Toronto, Canada, and his BEng degree in Electrical Engineering from the National Technical University of Athens, Greece.