

Environment Optimization for Crowd Evacuation

G. Berseth

University of British Columbia
gberseth@gmail.com
www.fracturedplane.com

M. Usman

York University
usman@cse.yorku.ca

B. Haworth

York University
m.brandon.haworth@gmail.com
www.cse.yorku.ca/~brandon/

M. Kapadia

Rutgers University
mubbasir.kapadia@gmail.com
www.cs.rutgers.edu/~mubbasir/

P. Faloutsos

York University
pfal@cse.yorku.ca
www.cse.yorku.ca/~pfal/

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.

Keywords: Crowd simulation, Optimization and Analysis

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

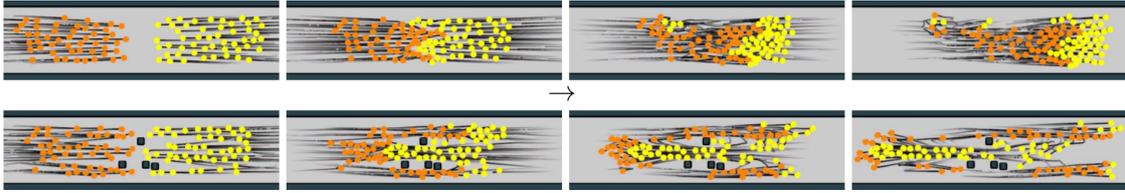


Figure 1: Snapshots in raster order of two similar scenarios where two groups of agents travel in opposite directions in a hallway. Placing four pillars at optimal locations (bottom) improves on average the flow for both groups compared to the case without pillars (top).

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 benchmarks. We use three different steering algorithms for this study, which include physically based models and optimal reciprocal methods. The benchmarks include variations on the number and placement of pillars, exit door sizes, as well as corridor and crowd flow configurations (uni-, bi- 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. In Section 4 we present our optimization formulation, we describe our benchmarks 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] 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 [3] 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 [4] use aggregate metrics for quantifying similarity. The work in [5, 6] measures the ability of a steering algorithm to emulate the behaviour of a real crowd dataset by measuring its divergence from ground truth. [7] 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 [8], or perceptual fidelity of relaxing collisions [9] in crowds.

Optimizing Crowd Simulation Parameters. Researchers [6, 10, 11, 12] have observed that the selection of a steering algorithm’s parameters can dramatically influence the performance and behavioural patterns of the aggregate crowd dynamics. The work in [13, 14] 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 [15, 16] use evolutionary approaches for procedural level creation and the placement of game level design elements. [17] optimizes platformer

games to maximize “fun”. [14] proposes a new method to improve the behaviour of crowd simulations using combinations of objectives. [18] uses optimization to find interesting variations in the game level, while [19] searched for optimal play space configurations for platformer games. [20] formulates a parameterization of the game level and evaluates each game level’s expected difficulty.

Architectural optimization. The works in [21, 22] 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 [23], 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 or infinite bounds defines a configuration space for the scenario that we refer to as scenario subspace, \mathcal{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 effects 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 the 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 \mathcal{P} .

3.1 Crowd Flow

There are a number of proposed measures to characterize the flow of a crowd [24, 25]. 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:

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

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:

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

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

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

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 because the method used to solve the problem, like many optimization methods, prefers constraints modelled 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, 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:

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

where

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

for distinct obstacles, o_1, o_2 , whose overlapping area, $ov(o_1, o_2)$, is non zero and 0 in all other cases.

4.2 The CMA-ES 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 [26]. The CMA-ES 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: (a) Global Navigation: selecting the next target location in space to steer towards, and (b) 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: (1) Jarring discontinuities in crowd flow due to the discretization of the environment into spatial grid cells. (2) Crowd congestion since 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 benchmarks, and the environmental features that we consider in our study. For all experiments, agents are represented by disks with a radius of 0.2286 meters. We use this radius as it gives us more realistic results and has no negative affect 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: (a) **ORCA**: an efficient and widely used technique that uses reciprocal velocity obstacles for collision avoidance [27], (b) **PPR**: a hybrid approach that uses rules to combine reactions, predictions, and planning [28] similar to [29], and (c) **SF**: a variant of the social forces method for crowd simulation [30]. 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 above steering algorithms on a variety of scenarios that exercise uni-,bi- and four-directional flows, using different configurations of corridors, pillars and exit doors.

5.2.1 Uni-directional Hallway

The configuration of this benchmark is shown in Figure 2. A hundred agents are randomly placed in a $12.5 \times 4 m^2$ region (blue). Up to 4 pillars are placed in the optimization region (grey). 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 $3.5 m$.

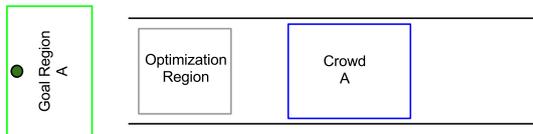


Figure 2: Uni-directional hallway scenario.

5.2.2 Bi-directional Hallway

This benchmark is an extension of the previous one with two groups of agents, A and B, travelling in opposite directions in the hallway, Figure 3. Each group contains 50 agents that are randomly placed in the corresponding blue region of size $6.25 \times 4 m^2$. Up to 4 pillars are placed in the $4 \times 4 m^2$ optimization region

(black). Each group must cross the optimization region to reach its corresponding target region (green).



Figure 3: Bi-directional-hallway scenario.

5.2.3 Two-way Egress

In this benchmark two groups of agents travelling 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 benchmarks. 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.

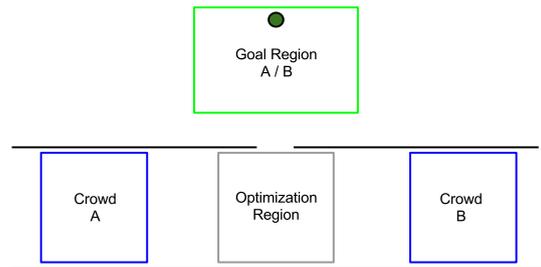


Figure 4: Two-way egress scenario.

5.2.4 Four-way Hallway

This is an extension of the previous bi-directional hallway benchmark in all cardinal directions, Figure 5. Four groups of 25 agents each travel from opposite directions in two hallways that share a $4 \times 4 m^2$ optimization region (grey) in the centre. The agents are randomly distributed in their corresponding region (blue) and must reach their target region (green) across the corresponding hallway.

These benchmarks 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 the method has this property. This can be attributed to the optimization algorithm’s robustness.

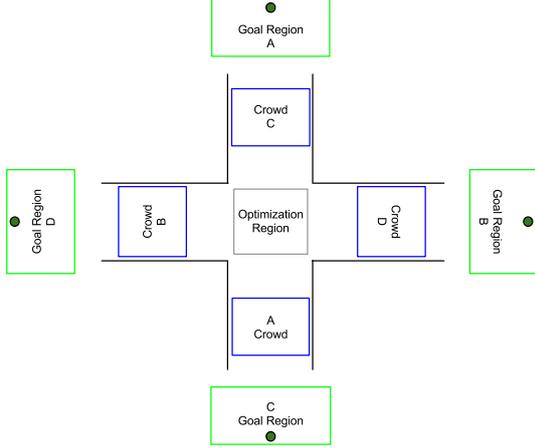


Figure 5: Four-way hallway scenario.

6 Experiments

We apply our methodology to variations of the benchmarks 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 uni-directional 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 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 behaviour with respect to pillar geometry. The **ORCA** algorithm had poor behaviour for circular pillars, treating the pillar as if it was an agent, signifi-

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

Table 1: The optimal crowd flow values, $f(\mathbf{p})$, for all experiments, where n -p means n pillars.

cantly 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, the axis-aligned boxes did not have this property. Last, the **PPR** algorithm was indifferent to the pillar geometry.

6.2 Uni-directional Hallway

Table 1 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 uni-directional hallway scenario, we can see that for all steering algorithms the crowd flow improves with the placement of pillars. Notably, three pillars produces the highest crowd flow for all three algorithms.

6.3 Bi-directional Hallway

Table 1 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 ap-

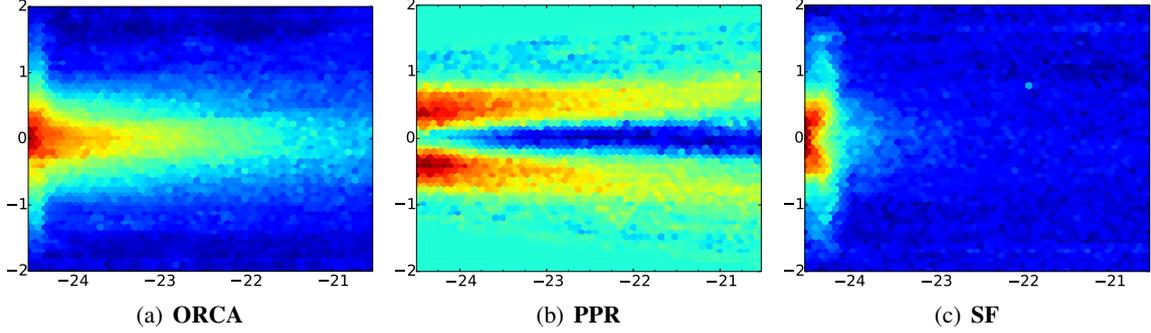


Figure 6: Uniform sampling of the optimization region for the uni-directional-hallway scenario, where blue indicates better flow. Clearly, the effect of the pillar differs per algorithm.

proximately 30% improvement with the optimal placement of two pillars.

6.4 Two-way Egress

Table 1 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 benchmark realistically without global planning.

6.5 Four-way Hallway

Table 1 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 behaviour the four groups seem to reach a deadlock in the middle of the hallways. On the other hand **ORCA** and **SF** both show 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 the rate of convergence depended not only on the steering algorithm but also the benchmark. Convergence was the fastest for the uni-directional hallway benchmark and the slowest for the two-way egress. This may be due to the change in *flow* direction the crowds need to make which is not present in the other

benchmarks. We show the optimization process for the **SF** algorithm on the four-way hallway benchmark 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 benchmarks.

The placement of additional pillars followed two patterns. In the case crowd flow was uni- or bi-directional, 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.

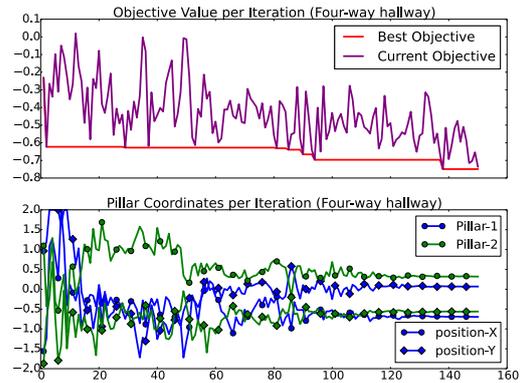


Figure 7: A visualization of the parameter selection process while optimizing the **SF** steering algorithm for the four-way hallway example with 2 pillars.

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 0 – 4 pillars for three different benchmarks. 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 uni-directional hallway scenario. Of particular interest is the case of 3 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** appear consistent for any number of pillars. The bottom row shows the histogram for **PPR** and the bi-directional 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 benchmarks that include doors and experiment with door openings that are $1.5\times$ and $2\times$ the original size.

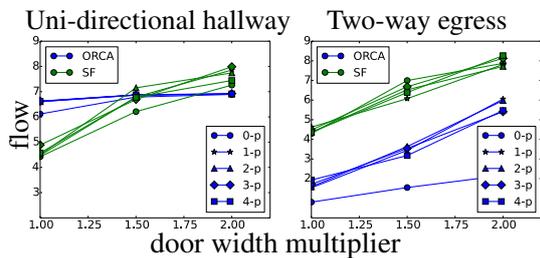


Figure 9: Optimal crowd flow for each algorithm for a few of the scenarios with larger size doors. Increasing door width generally increases crowd flow.

As expected, increasing the doorway size does improve the crowd flow for each of the steering algorithms. The improvement is significantly larger for the two-way egress scenario than for the uni-directional hallway scenario. **SF** surpasses **ORCA** in crowd flow for the largest doorway size in the uni-directional 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 this 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.

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 3. Door widths had a significant impact on crowd flow patterns, especially for bi-directional traffic, which highlights the importance of selecting the right door width depending on the expected crowd interactions.

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 behaviour of real humans. This is an important open question and the subject of future work.

Limitations. Our analysis is limited to homogeneous crowds and three specific algorithms. It would be interesting to extend the analysis to include other steering methods, and agents with diverse characteristics and behaviours.

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] Alon Lerner, Yiorgos Chrysanthou, Ariel Shamir, and Daniel Cohen-Or. Context-dependent crowd evaluation. *Comput. Graph. Forum*, 29(7):2197–2206, 2010.
- [2] S. Singh, M. Kapadia, P. Faloutsos, and G. Reinman. Steerbench : a bench-

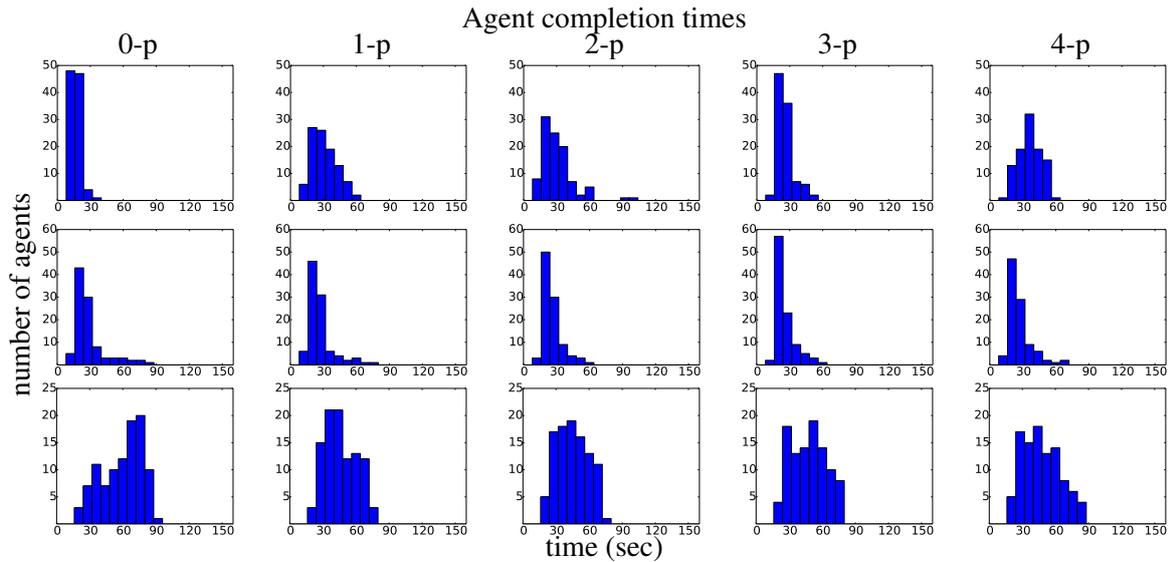


Figure 8: Agent completion histogram vs time. Top: **ORCA** for the uni-directional hallway scenario. Middle: **SF** for the four-way hallway scenario. Bottom: **PPR** for the bi-directional hallway scenario.

mark suite for evaluating steering behaviors. *Computer Animation And Virtual Worlds*, 20(February):533–548, 2009.

- [3] Mubbasir Kapadia, Matt Wang, Shawn Singh, Glenn Reinman, and Petros Faloutsos. Scenario space: characterizing coverage, quality, and failure of steering algorithms. In *Proceedings of ACM SIGGRAPH/EG SCA*, pages 53–62, 2011.
- [4] Armin Seyfried, Maik Boltes, Jens Khler, Wolfram Klingsch, Andrea Portz, Tobias Rupprecht, Andreas Schadschneider, Bernhard Steffen, and Andreas Winkens. Enhanced empirical data for the fundamental diagram and the flow through bottlenecks. In *Pedestrian and Evacuation Dynamics 2008*, pages 145–156. Springer Berlin Heidelberg, 2010.
- [5] Stephen J. Guy, Jur van den Berg, Wenxi Liu, Rynson Lau, Ming C. Lin, and Dinesh Manocha. A statistical similarity measure for aggregate crowd dynamics. *ACM TOG*, 31(6):11, 2012.
- [6] Julien Pettré, Jan Ondřej, Anne-Hélène Olivier, Armel Cretual, and Stéphane Donikian. Experiment-based modeling, simulation and validation of interactions

between virtual walkers. In *ACM SIGGRAPH/EG SCA*, pages 189–198, 2009.

- [7] Soraia R. Musse, Vinicius J. Cassol, and Cludio R. Jung. Towards a quantitative approach for comparing crowds. *Computer Animation and Virtual Worlds*, 23(1):49–57, 2012.
- [8] Rachel McDonnell, Michéal Larkin, Simon Dobbyn, Steven Collins, and Carol O’Sullivan. Clone attack! perception of crowd variety. *ACM Trans. Graph.*, 27(3):26:1–26:8, 2008.
- [9] Richard Kulpa, Anne-Hélène Olivier, Jan Ondřej, and Julien Pettré. Imperceptible relaxation of collision avoidance constraints in virtual crowds. In *ACM SIGGRAPH ASIA*, pages 138:1–138:10, 2011.
- [10] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You’ll never walk alone: Modeling social behavior for multi-target tracking. In *Proceedings of IEEE ICCV*, pages 261–268, 2009.
- [11] M. Davidich and G. Koester. Towards automatic and robust adjustment of human behavioral parameters in a pedestrian stream model to measured data. In *Pedes-*

- trian and Evacuation Dynamics*, pages 537–546. Springer US, 2011.
- [12] D. Wolinski, S. Guy, A.-H. Olivier, M. Lin, D. Manocha, and J. Pettr . Parameter estimation and comparative evaluation of crowd simulations. In *Eurographics*, 2014.
- [13] David Wolinski, Stephen J Guy, Anne-H el ne Olivier, Ming C Lin, Dinesh Manocha, and Julien Pettr . Optimization-based pedestrian model calibration for evaluation. *Transportation Research Procedia*, 2:228–236, 2014.
- [14] G. Berseth, M. Kapadia, B. Haworth, and P. Faloutsos. SteerFit: Automated Parameter Fitting for Steering Algorithms. In Vladlen Koltun and Eftychios Sifakis, editors, *Proceedings of ACM SIGGRAPH/EG SCA*, pages 113–122, 2014.
- [15] L. Cardamone, G. N. Yannakakis, J. Togelius, and P. L. Lanzi. Evolving interesting maps for a first person shooter. In *Applications of Evolutionary Computation*, pages 63–72. Springer, 2011.
- [16] N. Sorenson and P. Pasquier. Towards a generic framework for automated video game level creation. In *Proceedings of ICCX - Volume Part I*, pages 131–140. Springer-Verlag, 2010.
- [17] Nathan Sorenson and Philippe Pasquier. The evolution of fun: Automatic level design through challenge modeling. In *Proceedings of ACM ICCX*, pages 258–267, 2010.
- [18] Yinxuan Shi and Roger Crawfis. Optimal cover placement against static enemy positions. In *FDG*, pages 109–116, 2013.
- [19] A. W. Bauer, S. Cooper, and Z. Popovic. Automated redesign of local playspace properties. In *FDG*, pages 190–197, 2013.
- [20] G. Berseth, B. M. Haworth, M. Kapadia, and P. Faloutsos. Characterizing and optimizing game level difficulty. In *Proceedings of MIG*. ACM, 2014.
- [21] Samuel Rodriguez, Yinghua Zhang, Nicholas Gans, and Nancy M Amato. Optimizing aspects of pedestrian traffic in building designs. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1327–1334. IEEE, 2013.
- [22] Li Jiang, Jingyu Li, Chao Shen, Sicong Yang, and Zhangang Han. Obstacle optimization for panic flow-reducing the tangential momentum increases the escape speed. *PloS one*, 9(12):e115463, 2014.
- [23] Glen Berseth, Mubbasir Kapadia, and Petros Faloutsos. Steerplex: Estimating scenario complexity for simulated crowds. In *Proceedings of Motion on Games, MIG ’13*, pages 45:67–45:76, New York, NY, USA, 2013. ACM.
- [24] A. Johansson, D. Helbing, H. Z A-Abideen, and S. Al-Bosta. From crowd dynamics to crowd safety: A video-based analysis. *Advances in Complex Systems*, 11(4), 2008. arXiv:0810.4590.
- [25] D. Helbing, A. Johansson, and H. Zein Al-Abideen. Dynamics of crowd disasters: An empirical study. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 75(4):046109, 2007.
- [26] Nikolaus Hansen. A CMA-ES for Mixed-Integer Nonlinear Optimization. Technical Report RR-7751, INRIA, October 2011.
- [27] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *Robotics Research*, volume 70, pages 3–19. Springer, 2011.
- [28] S. Singh, M. Kapadia, B. Hewlett, G. Reinman, and P. Faloutsos. A modular framework for adaptive agent-based steering. In *Proceedings of I3D*, pages 141–150. ACM, 2011.
- [29] Mubbasir Kapadia, Shawn Singh, William Hewlett, Glenn Reinman, and Petros Faloutsos. Parallelized egocentric fields for autonomous navigation. *The Visual Computer*, 28(12):1209–1227, 2012.

- [30] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487–490, 2000.