Towards a Graphical User Interface for Exploring and Fine-Tuning Crowd Simulations

Andrea Bönsch[1,*] Marcel Jonda[1] Jonathan Ehret[1,*] Torsten W. Kuhlen[1,*]

[1]Visual Computing Institute, RWTH Aachen University, Germany

Abstract

Simulating a realistic navigation of virtual pedestrians through virtual environments is a recurring subject of investigations. The various mathematical approaches used to compute the pedestrians' paths result, i.a., in different computation-times and varying path characteristics. Customizable parameters, e.g., maximal walking speed or minimal interpersonal distance, add another level of complexity. Thus, choosing the best-fitting approach for a given environment and use-case is non-trivial, especially for novice users.

To facilitate the informed choice of a specific algorithm with a certain parameter set, crowd simulation frameworks such as Menge [3] provide an extendable collection of approaches with a unified interface for usage. However, they often miss an elaborated visualization with high informative value accompanied by visual analysis methods to explore the complete simulation data in more detail – which is yet required for an informed choice. Benchmarking suites such as *SteerBench* [14] are a helpful approach as they objectively analyze crowd simulations, however they are too tailored to specific behavior details. To this end, we propose a preliminary design of an advanced graphical user interface providing a 2D and 3D visualization of the crowd simulation data as well as features for time navigation and an overall data exploration.

Index Terms: H.5.2 [User Interfaces]: User Interfaces—Graphical user interfaces (GUI);

1 Introduction

A crowd simulation describes the autonomous navigation of a plethora of virtual entities through a virtual scene. While animals, i.e., flocks or schools, are only considered rarely, the entities referred to are mainly virtual pedestrians. Summarizing Pelechano et al. [11], crowd simulations thereby typically comprise the following two components: (1) In the *global path planning* component the scene's complete walkable space is considered to extract a high-level path per pedestrian, guiding the entity to its intended, final goal. (2) In the *local path planning* component the global path is refined by numerous, connected interim paths. These are chosen based on the current local surrounding of a respective pedestrian as well as the anticipated dynamic environmental chances due to, e.g., other pedestrian paths. While unnecessary movements of a pedestrian are minimized or avoided (accounting for a path's smoothness and efficiency), the paths are collision-free (w.r.t. to other entities as well as the scene itself) and guide the pedestrians constantly towards their final goals (goal-directed).

Algorithmically, both components can be modeled in various ways. Established methods for the global path planning require the scene's walkable space described as graph structures, i.e., as navigation meshes or roadmaps [7]. Search algorithms, i.e., A* or vector flow fields [2] are then used to compute the high-level path to the final goal. For the local path planning common approaches are social force models [5, 6], rule-based models [12] and predictive models [4].

*e-mail: {boensch|ehret|kuhlen}@vr.rwth-aachen.de

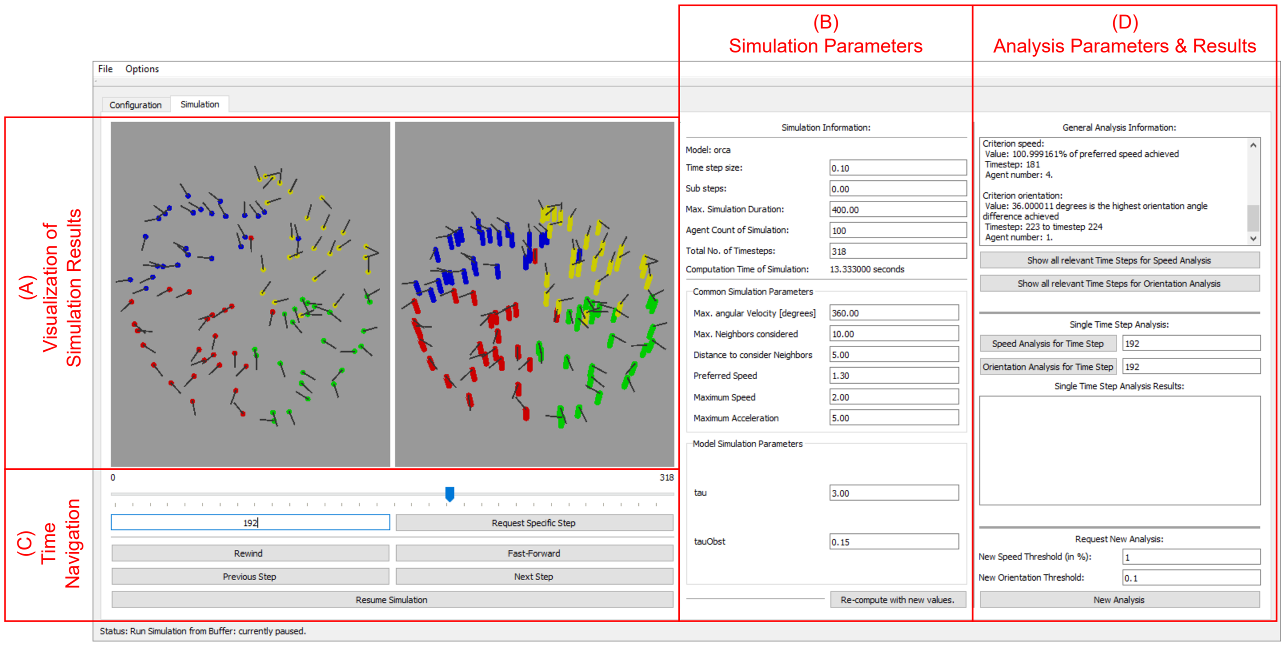
Although numerous solutions for the complex sub-problems of path finding in crowd simulations exist, the quality of the computed paths differ strongly in computation-related and path-related aspects. Computation-related factors are, e.g., the overall computation time per algorithm and memory usage. For the latter category, aspects like the smoothness of the computed paths or the duration till an entity reaches its final goal vary, influencing the naturalness of the resulting crowd behavior. In addition, various algorithms are parametrized, allowing knowledgeable users to adjust and thus fine-tune the computed paths to the virtual environment and pedestrian use-case. In case of a stressful evacuation scenario, the minimal accepted interpersonal distance between neighboring pedestrians might be reduced compared to low-stress simulations, while the maximal walking speed might be increased(e.g., [10]). Thus, choosing the best-fitting approach for a given environment and use-case is non-trivial.

A first valid step is to visually evaluate the global look-and-feel of a crowd simulation [9]. This step is facilitated by crowd simulation frameworks such as *Menge* [3] which provide collections of widely used approaches through a unified interface. Thus, users are able to quickly test different mathematical models in a given environment. Furthermore, the modularity of these frameworks allows a simple extension through the addition of new approaches.

In a second step, a more sophisticated evaluation based on objective grading approaches and metrics [9, 14] is preferable to further ease the selection process of appropriate approaches. One sample program for such an evaluation is the benchmark suite *SteerBench* [14] as part of the framework *SteerSuite* [13], which provides a set of universal metrics to score crowd algorithms. *SteerSuite* additionally provides an elaborated crowd visualization with a high informative value.

SteerBench focuses on a set of specified test cases, which cover "common, frequently appearing scenarios, but with challenging, worst-case parameters" [14]. Although additional flexibility is given by means of customizable scenarios, the suite's key focus is analyzing certain details of the locomotion behavior. This is an inevitable approach when developing own crowd algorithms or improving existing ones. However, the application of our tool is selecting an existing crowd algorithm for a use-case specific environment and agent behavior, while adapting the algorithm's parameters to fit the use-case.

To the best of our knowledge no such general algorithm fine-tuning suites are available. Thus, we contribute a preliminary design of an advanced graphical user interface (GUI) providing a first set of basic visual and analytical evaluation options for given crowd algorithms building upon *Menge*. A global look-and-feel evaluation is provided by a linked 2D and 3D view visualizing the virtual entities based on the positions computed by *Menge's* simulation core. Basic features such as manually altering parameters of the respective simulation allow an easy crowd adaption. For a more sophisticated evaluation, a proper time navigation is available as well as a data exploration at different time steps. These time steps are either chosen manually by the user or are proposed by an intelligent analyzer searching the simulation data for time steps of interest. The latter are points in time in which, e.g., a sudden in- or decrease of walking speed is found or time steps in which paths change their orientation drastically. These more advanced analytical criteria are thereby related to metrics used, e.g., in [14].


(B) Simulation Parameters
(D) Analysis Parameters & Results
File Options
Configuration Simulation
(A) Visualization of Simulation Results
(C) Time Navigation
0
318
Request Specific Step
Rewind
Fast-Forward
Previous Step
Next Step
Resume Simulation
Re-compute with new values.
Show all relevant Time Steps for Speed Analysis
Show all relevant Time Steps for Orientation Analysis
Speed Analysis for Time Step
Orientation Analysis for Time Step
New Analysis

192

Simulation Information:

Model: orca

Time step size:	0.10
Sub steps:	0.00
Max. Simulation Duration:	400.00
Agent Count of Simulation:	100
Total No. of Timesteps:	318
Computation Time of Simulation:	13.333000 seconds

Common Simulation Parameters

Max. angular Velocity [degrees]	360.00
Max. Neighbors considered	10.00
Distance to consider Neighbors	5.00
Preferred Speed	1.30
Maximum Speed	2.00
Maximum Acceleration	5.00

Model Simulation Parameters

tau	3.00
tauObst	0.15

General Analysis Information:

```
Criterion speed:
 Value: 100.999161% of preferred speed achieved
 Timestep: 181
 Agent number: 4.

Criterion orientation:
 Value: 36.000011 degrees is the highest orientation angle
difference achieved
 Timestep: 223 to timestep 224
 Agent number: 1.
```

Single Time Step Analysis:

Speed Analysis for Time Step	192
Orientation Analysis for Time Step	192

Single Time Step Analysis Results:

Request New Analysis:

New Speed Threshold (in %):	1
New Orientation Threshold:	0.1

Status: Run Simulation from Buffer: currently paused.

Figure 1: Simulation tab of the GUI while analyzing a crowd algorithm based on ORCA [15] for a circular-pedestrian-layout.

2 GUI TO EXPLORE CROWD SIMULATIONS

In this section, we will provide an overview of our features for the data exploration and fine-tuning of crowd simulations, divided into basic features for the visualization and the global look-and-feel evaluation (Sec. 2.1) and advanced features for a more sophisticated analysis (Sec. 2.2). While focusing on the GUI's main functions, convenience elements, i.e, status reports will be neglected. Furthermore, we assume that a connection between our GUI and the crowd simulation computation framework *Menge* was already established (cp. Sec. 3).

2.1 Basic Features

Our GUI consists of a configuration tab and a tab for the simulation and analysis of the simulation results. In the *configuration tab* users first select the crowd algorithm to be explored from a set of available approaches either provided directly by *Menge* or added as own implementation to the framework. Second, they specify the environmental settings of the scene and the pedestrians behavior. Both specifications have to be pre-scripted in a layout identical to *Menge's* XML specifications of the scene components (consisting of, e.g., start positions of the pedestrians, obstacle positions and dimensions) and the pedestrians behavior over time (as state machines). Thus, users can either rely on examples provided by *Menge* or they can easily specify their own use-case dependent scene and pedestrian settings.

After the configuration, users switch to the *simulation tab* to start the data exploration. The tab consists of four logical areas, presented in Figure 1. Details for areas (A) to (C) will be given in the following, while area (D) will be explained in Section 2.2. Accompanying the descriptions, user feedback of a small, preliminary qualitative evaluation with 7 computer scientists (*M*=27.14, *SD*=4.53) is directly provided.

Area (A) consists of a 2D orthographic and a 3D perspective view of the crowd simulation results. The individual pedestrians are represented by commonly used circles (2D) and cylinders (3D). Additionally, each pedestrian's walking direction for the next simulation step is indicated by means of a short vector. Both views are linked with respect to the simulation time step, ensuring to represent the exact same crowd status. In contrast, the presented section of the results per view can be changed freely and independently of each other by means of scrolling and zooming techniques via a basic input device. Thus, users are able to focus on specific regions of interest in the scene during their data exploration. In addition to this free navigation, users can select an individual pedestrian to receive more information. The respective entity is automatically highlighted in both views and its past and future trajectories, for up to 100 time steps each, are visualized as shown in Figure 2. Our subjects rated both views and the provided interactions positively and characterized area (A) as supportive and meaningful. Their suggestions for improvement were options to select several pedestrians in parallel and a more flexible time range for the past and future trajectories.

Area (B) provides an overview of the simulation parameters, divided into three categories. First, general configuration parameters are listed, i.e., the number of agents or the number of simulation time steps. Second, common crowd parameters used in the majority of crowd simulations are listed, e.g., the pedestrians preferred walking speed or minimal interpersonal distance between neighboring pedestrians. When not being used in a user-selected algorithm, the respective entry is disabled. Third and finally, model-specific parameters are listed, which are dynamically extracted from the chosen crowd algorithm. During their data exploration, users are able to alter these parameters and start a re-computation of the simulation to explore the parameter's impact on the results. Our subjects rated the parameter categorization positively. However, they asked for options to change the parameter sets not only for the complete population, but also for certain pedestrian subgroups.

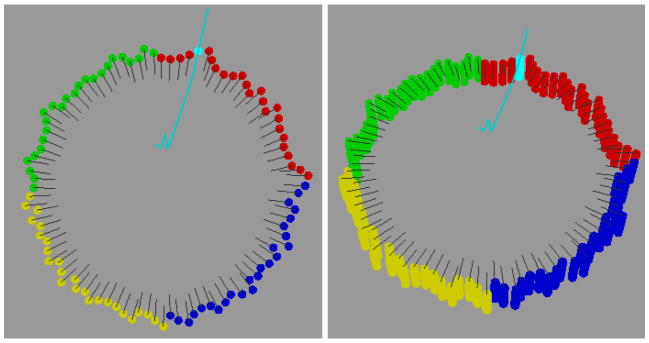


Figure 2: Past and future trajectories of a selected pedestrian for up to 100 time steps each.

Area (C) provides time navigation features to support users in the exploration of the time-variant crowd simulation results. In the area's top, a time slider is provided. A frame counter indicates which time step of the simulation is currently shown in area (A). Users can move the frame counter as known from various video and audio tools to scroll through the simulation or to quickly jump to certain time regions of interest. However, due to scaling issues this time navigation technique often does not allow a precise adaption by means of single time step manipulations. Thus, additional features are provided to overcome this shortcoming of the time slider. First, users can jump to a specific time step of the simulation, by entering the respective time step number in a designated time field beneath the time slider. Second, designated buttons to advance to the next simulation step or to go back to the previous time step allow an exact, time-step-based navigation in time. In addition, rewind and fast-forward buttons are provided quickly passing through the next 100 time steps in either direction. This time delta is coupled with the length of the illustrated trajectories of a selected pedestrian to quickly explore the next or previous path segments. Furthermore, the simulation can be paused and resumed by another designated button. Finally, additional information, e.g., the number of skipped time frames on rewind or fast-forward, appear on hover. Our subjects rated the functionality provided by area (C) positively and characterized the individual elements as easy and intuitive to use. However, they asked for a reordering of the buttons to improve the area's logical coherence. Additionally, they stated that they would prefer symbols instead of text for the buttons. By this, the GUI design will become leaner, while symbols for common functionality from various video and audio tools can be reused.

2.2 Features to Algorithmically Analyze Simulation Data

Besides the previously described basic functions, we developed a set of more advanced features to algorithmically analyze the simulation data. In contrast to *SteerBench*, we thereby do not compute scores or benchmarks, but present time steps in which potential miss-behaviors of pedestrians occur. For the miss-behavior detection, we made an initial selection of three criteria based on various general evaluation metrics described in the literature (e.g., [9, 13, 14, 16]): stops, speed and orientation. The first criterion is analyzed automatically on selecting an individual pedestrian in area (A), while area (D) provides control elements for the remaining two criteria. If one of the three criteria is used during the data exploration, its respective result is visualized on the time slider in area (B), replacing any previous analysis result (cp. Fig. 3).

Stops One goal of crowd algorithms is to avoid prolonged standstills or times of minimal movement of pedestrians, defined as stops, resulting from movement impediments due to scene geometry or neighboring pedestrian movements. Neglectable exceptions are stops due to inevitable waiting times while, e.g., a huge crowd passes through a narrow hallway or due to pedestrians reaching their final goal. Thus, finding time steps of stops in the simulation data is valuable in order to evaluate the suitability of the chosen crowd algorithm in the configured environment. It is worthy to note, that this criteria is not directly described in the literature. Nevertheless, it is based on merging the ideas of different individual metrics evaluating turns, distance and speed of the pedestrians. To evaluate the stops, the complete trajectory of a selected pedestrian is analyzed to find signs of blocking. If the distance traveled between two succeeding time steps t_i and t_{i+1} is close to zero, t_i is marked as relevant time step. To classify the relevance, the pedestrian's position in t_i is further analyzed. If it is the pedestrians final goal position, t_i is marked as *final*, otherwise as *stop*. The marked time steps are then highlighted on the time slider as shown in Figure 3(a): for stops red line markers are added, finals are represented by blue line markers. Thus, the fewer red markers occur, the better does the respective crowd parameters suite the scene.

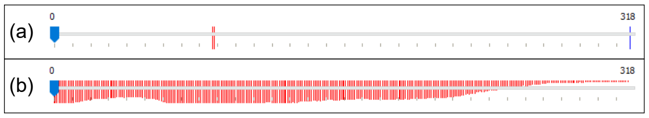


Figure 3: Two visualizations of relevant time steps (TSs): (a) TSs of minimal movement (red) and first TS of reaching the goal (blue) for one selected pedestrian. (b) Speed or orientation criterion evaluated for each TS, the length of the markers represent the number of pedestrians matching the criterion.

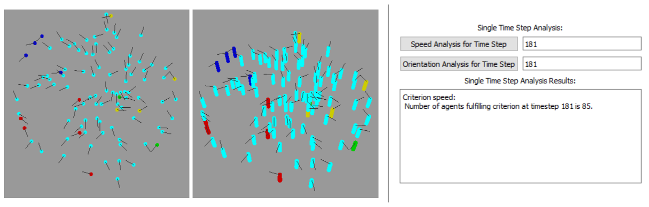

Single Time Step Analysis:
Speed Analysis for Time Step
Orientation Analysis for Time Step
Single Time Step Analysis Results:
181
181
Criterion speed:
Number of agents fulfilling criterion at timestep 181 is 85.

Figure 4: All pedestrians matching the speed criterion in time step 118, defined in area (D), are highlighted in both views of area (A).

Speed Crowd algorithms vary a pedestrian's speed with respect to its current as well as future surrounding. However, they aim to reach and maintain the preferred walking speed, one of the common crowd parameters. Consequently, it is relevant to find those time steps in which pedestrians reach a speed within an acceptable range around the actually preferred walking speed. Thus, the speed criterion is directly based on distance and speed metrics such as average speed described in [14]. As a few individuals failing in single time steps may not harm the overall application, all pedestrians need to be considered. Thus, the number of pedestrians meeting the speed criterion is relevant to improve the validity of the speed criterion. To this end, we count per time step, how many pedestrians travel in a speed within the acceptable speed range. The size of this speed range can thereby be specified by the user in area (D), while the preferred walking speed can be altered in area (B). To visualize the criterion results, markers are added to the time slider, as shown in Figure 3(b). The length of the markers are thereby scaled by the count of matching pedestrians per time step. In conclusion, the more lengthy adhering markers occur, the better is the crowd simulation suited for the evaluated scene.

Orientation The angular velocity of pedestrians influences the smoothness of the entity's trajectory and is thus of interest while exploring crowd algorithms. This is also described in literature when considering different turning metrics, e.g., in [14]. In contrast to the speed criterion, most crowd algorithms only define a maximal angular velocity, however no preferred one. Thus, our proposed analysis evaluates whether the difference between the angular speeds of a pedestrian in two succeeding time step t_i and t_{i+1} is within a certain threshold. If so, time step t_i is marked as relevant and it's entity count is increased by one. The respective threshold can be defined by the user in area (D). The technique to visualize the results of the orientation criterion is identical to the one used for the speed criterion. Thus, the more lengthy markers, the better is the crowd simulation suited for the evaluated scene.

Besides defining the thresholds in area (D), more elements for the speed and orientation criterion are provided here. Designated buttons are, e.g., used to recompute the criterion evaluation or to display the results of the last computation, while text fields summarize the visualized information. Furthermore, user's can retrieve more details per time step: after entering a time step number in area (D) for one of both criteria, the pedestrians matching the criterion are, e.g., highlighted in both views of area (A), as shown in Figure 4.

The idea of the analytical support during the data exploration met with a positive response among our subjects. However, they characterized the current techniques as rather unintuitive and stated that they were facing a steep learning curve handling them. Thus, the advanced analysis features need to be improved, as discussed in Section 4.

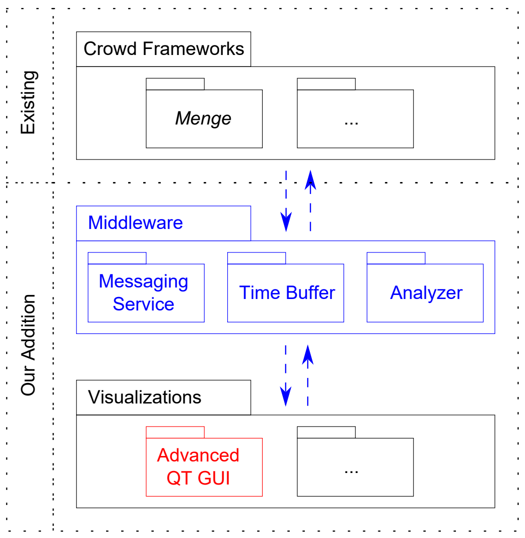


Figure 5: Architecture Overview

3 Software Setup

Figure 5 provides an overview of the architecture used to realize the data exploration of crowd simulations. Although the design was tested specifically with *Menge* and our proposed GUI, the setup enables exchanging both components if desired. This is due to our added *Middleware* which consists of three logical units. The *Messaging Service* unit is used to establish a bi-directional data transfer between the involved components: the visualization (VIS), the middleware, and the framework. Once all connections are established, the VIS can request simulation data for specific, user-configured crowd simulations. The request is retrieved by the middleware and forwarded to the framework. The results of the framework's computation core are then retrieved by the *Time Buffer*, where the data is stored and forwarded to the VIS. All requests made for the stored data, i.e., displaying another time step or analyzing the data, are then answered directly by the middleware: either new data is directly forwarded by the buffer unit, or the *Analyzer* unit computes the requested analysis using the data of the buffer and sends the results to the VIS. If another set of crowd parameters is tested or if the requested crowd simulation changes, the VIS informs the middleware, which requests the new data from the framework and the previously described data transfers are redone.

4 Discussion

While designing the GUI, we focused on evaluating which features may support users in gaining insight into time-variant crowd simulation data. We first considered basic features, i.e., orthographic and perspective visualizations of the pedestrians and their trajectories as well as time navigation elements, providing an option for a global look-and-feel evaluation of a crowd simulation. Furthermore, these features are widely used in different image, video or audio tools, allowing a fast familiarization and usage. This was confirmed by our seven subjects. However, these features require a manual data exploration. To ensure an exact and worthwhile evaluation of an algorithm's suitability in a specific setting, users require knowledge about the algorithm, the pedestrians' behavior, and the environment's characteristics. For novice users without this prior knowledge, a manual evaluation is thus tedious and time-consuming as all time steps have to be inspected closely.

To overcome this shortcoming, we added features which automatically recommend relevant time steps of the simulation for inspection, resulting in a guided data exploration. Thereby, we initially focused on three general criteria, related to state-of-the-art metrics used, e.g., in benchmark suites, which can be easily determined algorithmically: the number of standstills a pedestrian has, its speed as well as orientation changes. All three criteria highly impact the overall impression of a crowd simulation result. If pedestrians, e.g., change their orientation between time steps drastically, the resulting trajectories are not smooth and thus, unmotivated movements occur [11]. This will result in an unnatural behavior, impairing the overall impression. To this end, getting to know which time steps may contain unnatural behavior supports the evaluation process. Feedback of our subjects confirmed our thoughts, however it turned out that our design needs to be improved. The GUI does not provide any information on the standstill criterion, so using it needs further knowledge about the GUI itself. Subjects also had troubles understanding the threshold entries for the speed and the orientation criteria. In addition, only one criterion can be analyzed at a time, ignoring potential interaction effects which might be of relevance. Another aspect to be revised – which was interestingly not recognized by the subjects – is the contradiction in the evaluation of the three criteria: while time steps recommended for inspection are highlighted for the stops criterion, time-steps without a detected miss-behavior are highlighted for speed and orientation criteria. A consistent approach would be more beneficial, preferring the highlighting of time steps for inspection. Thus, the GUI needs to be revised and extended. Thereby, aiming for a lean design with intuitive elements, while embedding basic and advanced features in logical coherence is important (cp., e.g., Gestalt theory [1]).

Although we consciously started with a small set of analysis criteria for a crowd algorithm, an extension of this set would be reasonable. While time steps of standstills are already determined, deadlocks of pedestrians are not considered yet. These are time intervals in which pedestrians are sent to and fro between two positions, potentially forever. As this behavior contributes to an impaired impression of an algorithm's naturalness, these situations should be explicitly presented to the user. Additionally, adding criteria which focus on the model-specific parameters would further enhance the guided data exploration. Overall, various examples are provided in the literature which should be considered in the future work.

In our current design, we chose a time-slider-based result visualization for the evaluation criteria. This technique only provides time-dependent information, neglecting the space-dependent aspects. However, these spatial aspects might be of interest as well. Congestion, as mentioned in the last paragraph, e.g., only occur in a specific area of the tested environment. Thus, supporting users in finding this location quickly is as least as important as highlighting the time span of the congestion event. To this end, the design needs to be reconsidered, e.g., using a space-time-cube-based visualization (cp., e.g., [8]).

All features considered above allow analyzing one crowd algorithm with a specific parameter set and a defined environment at a time. Comparing two or more algorithms thus requires consecutive data explorations. This results in switching forth and back between the algorithms when comparing details. To avoid this tedious procedure, an option for a side-by-side comparison would be beneficial. Thereby, the free navigation by means of zooming and scrolling should be coupled between the respective views of the algorithms, while 2D and 3D can stay decoupled. When evaluating different parameter settings for the same crowd algorithm, a merged visualization may be of value. By showing, e.g., the current trajectories in contrast to the ones resulting from a different set of parameter values enables users to see all variations at a glance. Thus, users can quickly optimize the crowd algorithm for their defined environment.

In summary, the features combined in our preliminary GUI design turned out to be a valuable starting-point for developing a graphical tool supporting evaluation and fine-tuning of crowd simulations for complete environments. While confirming the need to provide simply usable elements for manual as well as guided data exploration, new concepts for enhanced functionality emerged as beneficial GUI extensions.

5 Conclusion

Choosing the best-fitting crowd algorithm for a specific environment is non-trivial, especially for novice users. Elaborated visualizations with high informative value accompanied by visual analysis methods would support an informed choice. While benchmarking suites focus on analyzing certain details of the locomotion behavior to improve the algorithms itself, our goal is to provide a tool for selecting and fine-tuning existing algorithms. Thus, we present a preliminary design of a graphical tool to explore time-variant crowd simulation data. Our proposed GUI consists of visualization and time navigation elements as well as features for a guided data analysis. The latter focus on relevant simulation aspects such as standstills of pedestrians, their walking speed, and orientation changes. A small preliminary evaluation indicated, that the proposed ideas are suitable to evaluate a crowd algorithm in a specific environment. However, future work should be focused on improving the interfaces for the available analysis criteria as well as on extending the guided data exploration.

References

[1] D. Chang, L. Dooley, and J. E. Tuovinen. Gestalt Theory in Visual Screen Design: A New Look at an Old Subject. *Proceedings of the Seventh World Conference on Computers in Education*, 8:5–12, 2002. doi: 10.5555/820060.820062

[2] X. Cui and H. Shi. A * -based Pathfinding in Modern Computer Games. *International Journal of Computer Science and Network Security*, 11(1):125–130, 2011.

[3] S. Curtis, A. Best, and D. Manocha. Menge: A Modular Framework for Simulating Crowd Movement. *Collective Dynamics*, 1(0):1–40, 2016. doi: 10.17815/CD.2016.1

[4] P. Fiorini and Z. Shiller. Motion Planning in Dynamic Environments Using Velocity Obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998.

[5] D. Helbing, I. Farkas, and T. Vicsek. Simulating Dynamical Features of Escape Panic. *Nature*, 407(6803):487–490, 2000. doi: 10.1038/35035023

[6] D. Helbing and P. Molnar. Social Force Model for Pedestrian Dynamics. *Physical Review E*, 51(5):4282, 1995. doi: 10.1103/PhysRevE.51.4282

[7] M. Kallmann and M. Mataric. Motion Planning Using Dynamic Roadmaps. *Proceedings of the IEEE International Conference on Robotics & Automation*, 5:4399–4404, 2004.

[8] M.-J. Kraak. The Space-Time Cube Revisited From a Geovisualization Perspective. In *Proceedings of the 21st International Cartographic Conference*, pp. 1988–1996. Citeseer, 2003.

[9] A. Lerner, Y. Chrysanthou, A. Shamir, and D. Cohen-Or. Data Driven Evaluation of Crowds. *International Workshop on Motion in Games*, pp. 75–83, 2009. doi: 10.1007/978-3-642-10347-6_7

[10] M. Moussaïd, M. Kapadia, T. Thrash, R. W. Sumner, M. Gross, D. Helbing, and C. Hölscher. Crowd Behaviour During High-Stress Evacuations in an Immersive Virtual Environment. *Journal of the Royal Society Interface*, 13(122), 2016. doi: 10.1098/rsif.2016.0414

[11] N. Pelechano, J. M. Allbeck, M. Kapadia, and N. I. Badler. *Simulating Heterogeneous Crowds with Interactive Behavior*. Taylor&Francis Inc., 2016.

[12] C. W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*, 21(4):25–34, 1987.

[13] S. Singh, M. Kapadia, P. Faloutsos, and G. Reinman. An Open Framework for Developing, Evaluating, and Sharing Steering Algorithms. *International Workshop on Motion in Games*, pp. 158–169, 2009. doi: 10.1007/978-3-642-10347-6_15

[14] S. Singh, M. Kapadia, P. Faloutsos, and G. Reinman. SteerBench: A Benchmark Suite for Evaluating Steering Behaviors. *Computer Animation and Virtual Worlds*, 20:533–548, 2009. doi: 10.1002/cav

[15] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha. Reciprocal n-Body Collision Avoidance. In *Robotics Research*, pp. 3–19. 2011. doi: 10.1007/978-3-642-19457-3_1

[16] D. Wolinski, S. J. Guy, A.-H. Olivier, M. C. Lin, D. Manocha, and J. Pettré. Parameter Estimation and Comparative Evaluation of Crowd Simulations. *Eurographics*, 33(2), 2014. doi: 10.1111/cgf.12328