Crowd Simulation with Detailed Body Motion and Interaction

Xinran Yao[1], Shuning Wang[1], Wenxin Sun[1], He Wang[2], Yangjun Wang[3], and Xiaogang Jin[1*]

[1] State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, P. R. China, jin@cad.zju.edu.cn
[2] University of Leeds, UK
[3] CROS of Tencent Games, P. R. China

Abstract. Crowd simulation methods generally focus on high fidelity 2D trajectories but ignore detailed 3D body animation which is normally added in a post-processing step. We argue that this is an intrinsic flaw as detailed body motions affect the 2D trajectories, especially when interactions are present between characters, and characters and the environment. In practice, this requires labor-intensive post-processing, fitting individual character animations onto simulated trajectories where anybody interactions need to be manually specified. In this paper, we propose a new framework to integrate the modeling of crowd motions with character motions, to enable their mutual influence, so that crowd simulation also incorporates agent-agent and agent-environment interactions. The whole framework is based on a three-level hierarchical control structure to effectively control the scene at different scales efficiently and consistently. To facilitate control, each character is modeled as an agent governed by four modules: visual system, blackboard system, decision system, and animation system. The animation system of the agent model consists of two modes: a traditional Finite State Machine (FSM) animation mode, and a motion matching mode. So an agent not only retains the flexibility of FSMs, but also has the advantage of motion matching which adapts detailed body movements for interactions with other agents and the environment. Our method is universal and applicable to most interaction scenarios in various environments in crowd animation, which cannot be achieved by prior work. We validate the fluency and realism of the proposed method by extensive experiments and user studies.

Keywords: crowd simulation, agent-based, terrain-adaptive

1 Introduction

High-fidelity crowd animation has been a central topic in various graphics applications and can be used in many applications such as computer games, industry films, and virtual reality. From a macroscopic perspective, the fidelity of crowd motion is determined by the group behavior, such as the authenticity of trajectories and the rationality of crowd movement [9]. From a microscopic point of

view, detailed individual motions also greatly affect the realism of crowd animation. While much effort has been spent on the former, the effort spent on the latter has mainly focused on single character animation. There has been little effort in systematic in-depth integration of both.

To address the aforementioned problems, we co-model crowd motions and individual motions, and propose a crowd behavior and animation control framework using a three-level hierarchy. The top level is the global control of crowd motions. The middle level targets the behavioral motions of different groups within the crowds. The low level governs the individual motions and agent interactions with other agents and the environment. We exhaustively evaluate our systems at different levels with both quantitative and qualitative metrics. The results show that our system can efficiently generate physically plausible and visually pleasing crowd motions with detailed individual motions and interactions. Formally, we propose a new three-level control framework for realistic crowd simulation with detailed individual motions. Also, we propose a new agent model for intelligent agent behaviors with awareness of the surroundings.

2 Related work

Crowd simulation raises numerous challenges e.g. modeling, authoring, rendering, animation, navigation, behavior, and perception [28] [33] [29].

2.1 Crowd Simulation

The classic method of crowd simulation is based on a force model [9]. In order to simulate a more real trajectory in crowds, a new mechanical model [2] is proposed to simulate the following movement. On the other side, the force-based model produces problems such as oscillation and bottleneck congestion [10]. So the agent-based crowd model comes into being. The crystal model [19] takes the influences of multiple factors into account and integrates the theories of sociology and anthropology into crowd simulation. Shao et al. [25] propose a more flexible model named automatic pedestrian model to simulate crowd. In terms of performance, the hierarchy structure for scene management can greatly improve the search, and it is widely used in games [21] and swarm simulation [22]. Similarly, Low et al. [18] also propose an agent-based crowd simulation framework dividing the perception model into high level and low level.

To further improve the decision-making ability of individual agents, Markov decision process is introduced into crowd simulation [24]. Besides, many new methods are proposed to improve the decision-making ability of agents with the rise of deep learning and reinforcement learning. Dünmez et al. [3] use the reinforcement learning method to improve the ability of individual obstacle avoidance. To achieve crowd simulation of intelligence, only four agents can be trained as leaders [27], followed by the Boids model [23]. Individual obstacle avoidance can also be improved with reinforcement learning [3].

Our research is categorically different from the above methods in that we focus on the integration of crowd simulation with detailed individual motions. Rather than employing character animation techniques as a separate step, we deeply root individual motions into crowd simulation.

Besides simulation, different metrics have been proposed to validate simulation fidelity, where comparing simulations with real data becomes popular [7, 8, 30–32]. However, these methods are designed to compare two sets of 2D trajectories. Given our aim is to generate detailed individual motions among crowds, they cannot be employed to evaluate our framework. We, therefore, propose our own quantitative and qualitative metrics in terms of physical plausibility and visual realism.

2.2 Motion Matching

Hoyet et al. [13] confirm that enhancing the animation adaptiveness to the environment can improve the authenticity of the simulation results by adding shoulder movement in crowd simulation. Considering the authenticity of leg movements, Narang et al. [20] propose a motion simulation method based on gait constraints. However, the above methods only simulate the scenes on flat ground. Recently, deep learning has been used in various games [16]. The PFNN model [12] can enable individuals to complete a series of complex actions, which can well adapt to terrain changes.

In addition to terrain adaptiveness, interaction with other objects is also within the scope of motion matching. Agrawal et al. [1] propose an action template that could realize a variety of footstep animations for specific tasks, such as sidestep. However, the matching of the above interactive actions depends on a large amount of data and can only be used on specific tasks. A collaborative animation/simulation model [5] is developed by embedding multiple character animation methods in the form of components within the same framework.

Our research is also orthogonal to motion matching. It depends on motion matching on the low-level motion generation, but focuses on how agents are influenced by high level information such as crowd or local group behaviors.

3 Overview

The whole simulation scene is under the control of a hierarchical structure called "Global-Group-Individual". At the individual level, an agent model is built to achieve auto-perception, auto-decision, and cooperation with other group members. The animation system, as a co-simulation system, is constructed by a motion matching mode implemented by a phase-functioned neural network (PFNN) model [12] and a traditional animation mode implemented by finite state machine (FSM) [6]. An overview of our method is shown in Fig. 1.

4 Hierarchical Control

In our system, the structure has three levels: global, group, and individual.

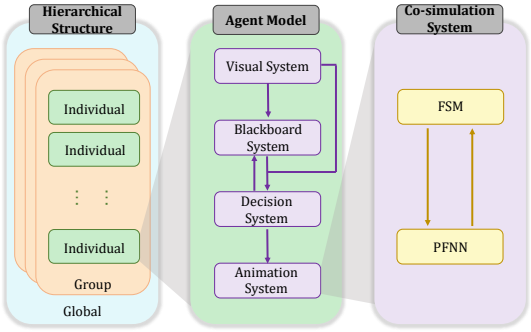

Fig. 1. The overview of our approach. The whole framework consists of three parts: hierarchy structure, agent model, and co-simulation system.

Global This level acts as a global control to manage the information that needs to be broadcast. For environmental information management, the global level abstracts the scene into a waypoint map [4] and a grid map as shown in Fig. 2. Waypoint maps describe the feasible areas and the location of obstacles in the scene. The grid map G is defined as $G = (C_x, C_y, w, h, d)$, in which C_x and C_y is the position of the lower left corner of the map on the horizontal axis and the vertical axis, w is the width of the map, h is the height of the map, and d is the size of grids. Every grid $g(i, j)$ stores the pointer to the head of a doubly linked list. The position of each individual is stored in a linked-list node and linked to an appropriate list head. The relation between individual position (p_x, p_y) and grid $g(i, j)$ can be expressed as $i = \lfloor \frac{p_x - C_x}{d} \rfloor, j = \lfloor \frac{p_y - C_y}{d} \rfloor$.

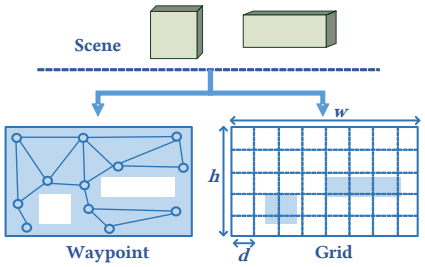

Fig. 2. The schematic diagram of the abstract structure of global level. Waypoint map and grid map are used to represent the entire scene. In the grid map, w and h are the width and the height of the map, respectively.

Group This level acts as the role of a leader, responsible for controlling the individuals in a group, updating their information and coordinate motions, such as moving to a specific target.

Individual This layer has two motion components: individual and group members. The former deals with the plausibility of individual motions, and the latter ensures that the individual motions are also consistent with group behaviors.

5 Agent Model

Fig. 4 illustrates the agent model which consists of four modules: visual system, blackboard system, decision system, and animation system. The four models correspondingly simulate four functions of human beings: perception, memory, decision-making, and behavior.

5.1 Visual System

The scope of human observation is assumed to be a fan-shaped area centered on the agent location. Based on this assumption, the visual system is divided into three layers: short-distance perception, mid-distance perception, and long-distance perception. Fig. 3 depicts the structure of the visual system.

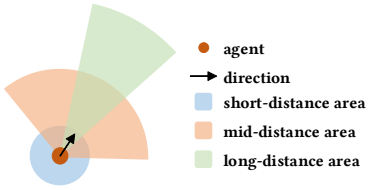

Fig. 3. The structure of visual system. The visual system is divided into three layers. Different layers correspond to different viewing distance and viewing angle.

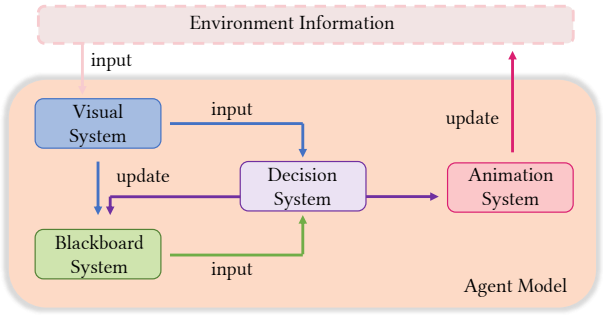

Fig. 4. The framework of the agent model. Based on the external environment information, the visual system passes the information to the blackboard system and the decision system. According to the information from pre-order systems, the decision system sends the decision results to the animation system. Then the animation system controls the agent to move or interact with other agents.

The grid map is used to get the information of the short-distance and mid-distance perception layers. We inspect the nodes in the covered grids after selecting a rough rectangular range based on the double length of the visual distance.

For the mid-distance perception layer or other layers whose visual angle less is than 180°, the coverage area can be narrowed by a straight line passing through the center and perpendicular to the forward direction of the agent. The ray method is used in the long-distance layer. The position of the agent and the emission angles are sampled within the visual angle range.

5.2 Blackboard System

The blackboard system is organized as a dictionary data structure with the item name as keywords and item information as contents. A blackboard is separated into two parts to store individual attributes and memorize knowledge respectively. The individual attributes, defining the characteristics of agents. Memory knowledge has three blocks of content: public, friend, and private. Only the owner of the blackboard has the authority to modify the contents while other individuals have the authority to notify the modification.

5.3 Decision System

The decision system is composed of motion decision and behavioral decision. Motion decision, which determines the movement speed of the agent in the next frame, is influenced by the target position and the surrounding environment. It is divided into two parts: pathfinding implemented by waypoint maps and collision avoidance based on a social force method. The pathfinding part determines the general direction of individual movement while the collision avoidance part adjusts the local movement speed taking the surrounding environment into consideration. The attractive force is calculated according to the moving speed determined by pathfinding. Based on the visual system's perception of the surrounding individuals, the repulsive forces and frictions are calculated according to the distance to the surrounding individuals perceived by the visual system. Behavior Tree is generally used for fast-action games to create interactive characters with a similar social intelligence like soldiers in a battlefield [14, 17]. Therefore, our method uses BT to determine the behavior state of the next frame.

5.4 Animation System

The animation system is composed of the traditional animation state machine [6] and the PFNN model [12]. Animation state machine is responsible for controlling actions that interact with other objects. The PFNN model replaces the animation related to the interaction of the terrain in the traditional animation state machine, so as to ensure the adaptability of the agent to the terrain in the process of movement.

6 Co-simulation System

6.1 Mode Switch

The switching process is shown in Fig. 6. We propose a method based on the interpolation of intermediate transition animations to realize the fluent switch

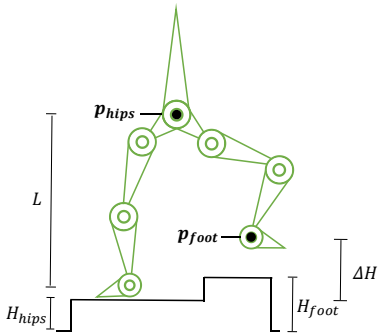

Fig. 5. The diagram of foot position in IK. This figure shows how the foot movement adapts to height on uneven ground. Here p_{hips} represents the position of the agent's center and p_{foot} represents the position of the agent's foot.

between the PFNN model and the animation state machine, in which a transition clip is inserted between the traditional animation state machine and the PFNN model.

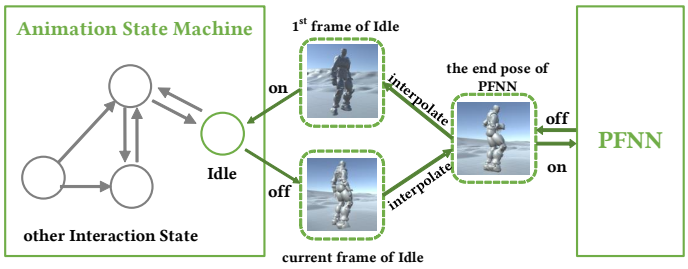

Fig. 6. The process of mode switches between the animation state machine and the PFNN model. We interpolate a transition animation between the traditional animation state machine and the PFNN model.

6.2 Inverse Kinematic

In the PFNN mode, IK is realized by the geometrical analysis method, which is offered by the IK components of Unity [15]. With the IK result, we update the position of each joint. In the animation state machine mode, we only need to set an appropriate foot position as the endpoint for IK, and the height of position can be sampled from height maps. However, the feet will slide on the ground if the foot height of each frame is directly set as the height of the ground. The relationship between foot positions and terrain height is shown in Fig. 5 and the calculation process is described as follows:

1. Get the current center position p_{hips} and foot position p_{foot} for every frame;
2. According to the projection position of p_{hips} and p_{foot} on the ground, the height under them, h_{hips} and h_{foot}, are sampled directly;

3. Calculate the target height difference between the foot and the ground $\Delta H = L - (L + h_{hips} - p_{foot}.y) = p_{foot}.y - h_{hips}$, where L is the height difference between the hips and the ground;
4. Calculate the actual height difference $p_{foot}.y = H_{foot} + \Delta H * \max(\frac{L+h_{hips}-h_{foot}}{L}, 0)$, where $\max(\frac{L+h_{hips}-h_{foot}}{L}, 0)$ is the regulator of height difference. When the sampled height of the foot is larger than that of the hips, the affection of ΔH decreases, and the regulator is smaller therefore. The more similar the sampled heights are, the closer the actual height of the foot to ΔH.

7 Results

All simulation results in this paper are achieved through the game engine Unity, 2017.1.5F1 version, and the program runs in the environment with Intel Core i7-8700 CPU and NVIDIA GeForce RTX 2070 GPU.

Fig. 7. Crowd behaviors simulated by our method. The whole team moves toward the same goal (left) and each individual tends to avoid collision with others (right).

Fig. 8. Results of terrain adaptiveness for training data. Movements can well adapt to different kinds of terrains. Even on an unusual terrain (the left figure), the joint angle can be adjusted to adapt to the terrain.

7.1 Behaviour Simulation

For all the presented results in the scenario, the red and blue teams are against each other, and the goal of a group always finds the enemy members. Fig. 7(left)

shows that both teams have a tendency to move towards the nearest enemy member no matter whether the group members are gathered or dispersed at present, and the movement is influenced by the group's overall goal. In Fig. 7(right), the individual movement trajectory clearly shows the collision avoidance with obstacles and other individuals.

7.2 Animation Simulation

This section focuses on the animation adaptiveness to terrain and shows the results from multiple perspectives. Fig. 8 shows the results of terrain adaptiveness for training data, and Fig. 9 shows the animation result of the character moving on a very steep cliff. As can be seen from the figure, the new environment also gives rise to new motions, such as climbing a cliff, jumping, and maintaining body balance while sliding down a cliff with the help of arms. In addition, in order to verify the necessity of adding IK, in the process of experiment, we also compared the situation of with and without IK. We define a slipping error to represent the degree of foot slipping. For each foot landing, the position where the toe joint first contacts the ground is taken as the landing point, and the horizontal distance between the toe joint and the landing point in subsequent frames is calculated.

Fig. 9. Results of terrain adaptiveness for testing data. Our method can also produce acceptable results on special terrains such as a cliff.

Table 1. Average Penetration Error about Movement

Terrain Type	Our Method with IK(cm)	Our Method without IK (cm)	FSM (cm)
flat	**0.000470**	0.001627	1.232807
uneven	**1.136922**	3.756697	2.595681

Table 2 shows the average slipping errors on the flat and uneven ground caused by different methods. The results of the table show that the foot slipping produced by our method is much more slighter than that of the animation state machine, and the phenomenon is further weakened with the assistance of IK.

Another experiment mainly studies the clipping in the crowd animation. Table 1 shows the average penetration errors calculated from each method on

different terrains when characters are moving. The results show that our method with IK produces the least penetration error and can effectively weaken the penetration phenomenon.

Table 2. Average Slipping Error

Terrain Type	Our Method with IK(cm)	Our Method without IK(cm)	FSM(cm)
flat	10^{-6}	0.115825	22.974229
uneven	**2.603551**	3.05557	15.985041

7.3 User Study

To evaluate the smoothness of mode switching in the animation system, we conduct a user study with two clips of the comparison videos, one is a single confrontation while another is a group confrontation. Each video contains two segments, one is simulated by our method and the other only by the animation state machine [6]. Participants were asked to rate the videos on two dimensions, fluency and reality. The score is on a scale of 1 to 9, with a lower score indicating better performance in the first segment on that dimension, and vice versa. Results show that the average score of the single confrontation is 3.10 on fluency and 3.71 on reality, respectively. It is clear that our method outperforms the traditional method in terms of both fluency and reality. In terms of group confrontation, the fluency score and the reality score are 5.14 and 5.10 respectively, which is still better than the traditional animation state machine method [6].

8 Conclusion and Future Work

We have presented a novel hierarchical crowd behavior and animation control framework with detailed body locomotion and interaction, which can control at various levels consistently and support the interaction of multiple agents/groups in complex terrain scenes. Under control of the hierarchy structure, the information is propagated from level to level, which simplifies the complexity of scene management and improves the efficiency of information transfer. The four modules in the agent model are complementary to each other, which together make the agent motion realistic, intelligent, and flexible. At the same time, in order to perform terrain adaptiveness, the PFNN model is employed into the animation state machine. This co-simulation can simulate a variety of interactive tasks on different kinds of terrains.

Our method has some limitations. The motion matching method in our implementation can only control characters with a fixed skeleton structure. In addition, our simulation results of motion matching are highly dependent on the quality of the dataset. In the future, we would like to improve the ability of decision-making by the state-of-art technology, such as reinforcement learning to further strengthen the intelligence of agents. Generating crowd animation

through the learned motion matching model [11] or the neural state machine [26] is also an interesting direction to explore.

Acknowledgement Xiaogang Jin was supported by the National Natural Science Foundation of China (Grant No. 62036010) and the Key Research and Development Program of Zhejiang Province (Grant No. 2020C03096).

References

1. Agrawal, S., van de Panne, M.: Task-based locomotion. ACM Transactions on Graphics (TOG) **35**(4), 82:1–82:11 (2016)
2. Chraibi, M., Tordeux, A., Schadschneider, A.: A force-based model to reproduce stop-and-go waves in pedestrian dynamics. In: Traffic and Granular Flow'15, pp. 169–175 (2016)
3. Dönmez, H.A.: Collision avoidance for virtual crowds using reinforcement learning. Master's thesis (2017)
4. Felder, A., Van Buskirk, D., Bobda, C.: Automatic generation of waypoint graphs from distributed ceiling-mounted smart cameras for decentralized multi-robot indoor navigation. In: Proceedings of the 13th International Conference on Distributed Smart Cameras, pp. 1–7 (2019)
5. Gaisbauer, F., Lehwald, J., Agethen, P., Sues, J., Rukzio, E.: Proposing a co-simulation model for coupling heterogeneous character animation systems. In: Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP, pp. 65–76 (2019)
6. Gillies, M.: Learning finite-state machine controllers from motion capture data. IEEE Transactions on Computational Intelligence and AI in Games **1**(1), 63–72 (2009)
7. Guy, S.J., van den Berg, J., Liu, W., Lau, R., Lin, M.C., Manocha, D.: A statistical similarity measure for aggregate crowd dynamics. ACM Transaction on Graphics (TOG) **31**(6) (2012)
8. He, F., Xiang, Y., Zhao, X., Wang, H.: Informative scene decomposition for crowd analysis, comparison and simulation guidance. ACM Transaction on Graphics (TOG) **4**(39), 50 (2020)
9. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature **407**(6803), 487–490 (2000)
10. Helbing, D., Johansson, A.: Pedestrian, crowd, and evacuation dynamics. In: Encyclopedia of complexity and System Science, pp. 6476–6495 (2009)
11. Holden, D., Kanoun, O., Perepichka, M., Popa, T.: Learned motion matching. ACM Transactions on Graphics (TOG) **39**(4), 53 (2020)
12. Holden, D., Komura, T., Saito, J.: Phase-functioned neural networks for character control. ACM Transactions on Graphics (TOG) **36**(4), 42:1–42:13 (2017)
13. Hoyet, L., Olivier, A.H., Kulpa, R., Pettré, J.: Perceptual effect of shoulder motions on crowd animations. ACM Transactions on Graphics (TOG) **35**(4), 53:1–53:10 (2016)
14. Johansson, A., Dell'Acqua, P.: Emotional behavior trees. In: 2012 IEEE Conference on Computational Intelligence and Games (CIG), pp. 355–362 (2012)
15. Juliani, A., Berges, V.P., Teng, E., Cohen, A., Harper, J., Elion, C., Goy, C., Gao, Y., Henry, H., Mattar, M., et al.: Unity: A general platform for intelligent agents. arXiv preprint arXiv:1809.02627 (2018)

16. Justesen, N., Bontrager, P., Togelius, J., Risi, S.: Deep learning for video game playing. IEEE Transactions on Games **12**(1), 1–20 (2020)
17. Llobera, J., Boulic, R.: A tool to design interactive characters based on embodied cognition. IEEE Transactions on Games **11**(4), 311–319 (2019)
18. Low, M., Cai, W., Zhou, S.: A federated agent-based crowd simulation architecture. In: The 2007 European Conference on Modelling and Simulation, Prague, Czech Republic, pp. 188–194. Citeseer (2007)
19. Manenti, L., Manzoni, S.: Crystals of crowd: Modelling pedestrian groups using mas-based approach. In: Proceedings of the 12th Workshop on Objects and Agents, Rende (CS), Italy, Jul 4-6, 2011, *CEUR Workshop Proceedings*, vol. 741, pp. 51–57 (2011)
20. Narang, S., Best, A., Manocha, D.: Simulating movement interactions between avatars & agents in virtual worlds using human motion constraints. In: 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 9–16 (2018)
21. Nystrom, R.: Game programming patterns (2014)
22. Ren, J., Sun, W., Manocha, D., Li, A., Jin, X.: Stable information transfer network facilitates the emergence of collective behavior of bird flocks. Physical Review E **98**(5), 052,309 (2018)
23. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. In: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pp. 25–34 (1987)
24. Ruiz, S., Hernandez, B.: Real time markov decision processes for crowd simulation. GPU Zen pp. 323–341 (2017)
25. Shao, W., Terzopoulos, D.: Autonomous pedestrians. Graphical models **69**(5-6), 246–274 (2007)
26. Starke, S., Zhang, H., Komura, T., Saito, J.: Neural state machine for character-scene interactions. ACM Transactions on Graphics (TOG) **38**(6), 209:1–209:14 (2019)
27. Sun, L., Zhai, J., Qin, W.: Crowd navigation in an unknown and dynamic environment based on deep reinforcement learning. IEEE Access **7**, 109,544–109,554 (2019)
28. Thalmann, D.: Crowd simulation. Wiley encyclopedia of computer science and engineering (2007)
29. van Toll, W., Pettré, J.: Algorithms for microscopic crowd simulation: advancements in the 2010s. Computer Graphics Forum **40**(2) (2021)
30. Wang, H., Ondřej, J., O'Sullivan, C.: Path patterns: Analyzing and comparing real and simulated crowds. In: ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2016, pp. 49–57 (2016)
31. Wang, H., Ondrej, J., O'Sullivan, C.: Trending paths: A new semantic-level metric for comparing simulated and real crowd data. IEEE Transactions on Visualization and Computer Graphics **23**(5), 1454–1464 (2017)
32. Wang, H., O'Sullivan, C.: Globally continuous and non-markovian crowd activity analysis from videos. In: B. Leibe, J. Matas, N. Sebe, M. Welling (eds.) Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V, vol. 9909, pp. 527–544 (2016)
33. Zhou, S., Chen, D., Cai, W., Luo, L., Low, M.Y.H., Tian, F., Tay, V.S.H., Ong, D.W.S., Hamilton, B.D.: Crowd modeling and simulation technologies. ACM Transactions on Modeling and Computer Simulation (TOMACS) **20**(4), 1–35 (2010)