DOI: 10.1111/cgf.14630
ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2022
D. L. Michels and S. Pirk
(Guest Editors)
Volume 41 (2022), Number 8

A Fusion Crowd Simulation Method: Integrating Data with Dynamics, Personality with Common

Tianlu Mao[1] , Ji Wang[1], Ruoyu Meng[2], Qinyuan Yan[1], Shaohua Liu[2] and Zhaoqi Wang[1]

[1]Institute of Computing Technology Chinese Academy of Science, China
ltm@ict.ac.cn, wangji18s@ict.ac.cn, yanqinyuan20s@ict.ac.cn, zqwang@ict.ac.cn
[2]Beijing University of Posts and Telecommunications, China
nengruoyu@bupt.edu.cn, liushaohua@bupt.edu.cn

Abstract

This paper proposes a novel crowd simulation method which integrates not only modelling ideas but also advantages from both data-driven methods and crowd dynamics methods. To seamlessly integrate these two different modelling ideas, first, a fusion crowd motion model is developed. In this model the motion of crowd are driven dynamically by different forces. Part of the forces are modeled under a universal interaction mechanism, which describe the common parts of crowd dynamics. Others are modeled by examples from real data, which describe the personality parts of the agent motion. Second, a construction method for example dataset is proposed to support the fusion model. In the dataset, crowd trajectories captured in the real world are decomposed and re-described under the structure of the fusion model. Thus, personality parts hidden in the real data could be locked and extracted, making the data understandable and migratable for our fusion model. A comprehensive crowd motion generation workflow using the fusion model and example dataset is also proposed. Quantitative and qualitative experiments and user studies are conducted. Results show that the proposed fusion crowd simulation method can generate crowd motion with the great motion fidelity, which not only match the macro characteristics of real data, but also has lots of micro personality showing the diversity of crowd motion.

CCS Concepts

• ***Computing methodologies*** → ***Multi-agent planning;*** • ***Applied computing*** → ***Law, social and behavioral sciences;***

1. Introduction

Vivid crowd simulation could increase the visual fidelity of virtual scenes, and becomes more and more important for animations, games and virtual reality. It is also vital for applications of public security and intelligent traffic, since realistic crowd simulation could provide a credible experimental environment for studies of crowd. In real crowd, each person, as a highly autonomous individual, can move flexibly avoiding surrounding ones and obstacles while reaching its destination efficiently and naturally. They also move differently from each other presenting diverse personality. Furthermore, as a collection of individuals, crowd movement follows some general laws of crowd dynamics. Therefore, it is a great challenge to simulate the motion of crowd, matching the macro and micro laws of real crowd and showing diverse and natural motion with collision-free interaction.

Extensive crowd simulation methods have been proposed from different technical point of views to handle the challenge. Roughly, they can be categorized into rule-based approaches and data-driven approaches. Rule-based approaches set rules for virtual crowds such as potential field, social force, cellular automata etc [XJJD14] [YLG*20] [vTP21] [MCT21]. They usually generate collision-free crowd motion but fail to produce diverse motion and natural interactions. Data-driven approaches extract features from the real data to imitate real crowds' motion. They can generate diverse and natural motion but fail to deal with collision problem. Furthermore, many data-driven approaches have high dependency between data and scenario. That means the data they used is only applicable to generate crowd scenario similar to data scenes.

This paper proposes a fusion crowd simulation method to combine both data-driven and rule-based modelling idea. Firstly, a fusion crowd motion model is developed which drives agents dynamically by different forces. Part of the forces are modeled under a universal interaction mechanism from [KSG14], which describe the common parts of crowd dynamics. Others are modeled by examples from real data, which describe the personality parts of the agent motion. Secondly, a construction method for example dataset is proposed, in which personality parts in real crowd trajectories are extracted by reverse solving the proposed fusion model and stored in dataset as a set of examples. Thirdly, a comprehensive crowd motion generation workflow is proposed which selects the appropriate example from the database for each agent based on its state and

drives them step by step. We conduct quantitative and qualitative experiments to evaluate the performance of the proposed method. Results proves that the method can generate realistic crowd motion which match the macro and micro laws of real crowd and show diverse and natural motion with collision-free interaction. Furthermore, a user study is conducted to prove our method offer a better visual experience. The main contribution of this work can be summarized as follow:

1. A novel fusion crowd motion model is proposed. It defines crowd motion in two parts, common dynamic parts and personality parts, and integrates them into a force driven framework. So that, it has the ability to present not only common dynamic law, but also diverse personality in crowd motion.
2. A construction method for example dataset is proposed which extracts personality parts from real data. In our method, personality parts mean the forces that push agents moving a little different from the general model. Such difference are tiny but vital to diverse and natural crowd motion. They are difficult to model with rules. So we extract them from real data by reverse solving the proposed fusion model and translate them into forces. Thus, personality parts in real data could be integrated seamlessly into a force drive crowd framework. Furthermore, personality parts have relatively low scenario dependency than trajectory or velocity, and could be transplant to different crowd scenes by our fusion model.

2. Related work

2.1. Rule-based Approaches

Rule-based approaches do not depend on real crowd motion data and define crowd‘s motion by a set of rules. These approaches aim to generate reliable collision-free agent motions. Following modeling the crowd as an entirety or as individuals, these approaches can be divided into macroscopic rule-based approaches and microscopic rule-based approaches.

Macroscopic rule-based approaches set moving rules for all agents in their entirety. [Hug00] [Hug02] [Hug03] considers crowd as flowing continuums and uses fluid mechanics to model them. [TCP06] builds a global dynamic potential field based on agent's intentional velocity, goal location, and obstacles, then use it to guide the agent motion. [BSK16] computes the harmonic field of the environment and uses the Reeb Graph to present the topology of the environment. With the help of harmonic field and Reeb Graph, crowd's motion can be acquired via a lightweight algorithm. Macroscopic rule-based approaches ignore the interactions among individuals, so they have high calculating efficiency and better global effects for large-scale crowd simulation. However, this neglect also results in homogeneity and lacks local details in crowd.

Microscopic rule-based approaches concentrate on the single agent's motion. They calculate the change of motion state at the next time step based on self-motion state and environment situation for every agent separately. [GM85] proposes the method which predicts agent locations using cellular methods. Further, [BEA97] [BA99] and [MIN99] introduce Cellular Automata into agent movement simulation. Force-based methods consider that agent motion is influenced by various virtual forces. [OM93] simulates crowd's behavior by magnetic method. [HFV00] proposes the social force model, which simulates escape crowd panic behavior by a set of socio-psychological and physical forces conducted by the goal, obstacles, and other agents. [KSG14] further studies the relationship between impact strength and estimated impact time from real data and propose Power Law in agent motion. In addition, [VdBLM08] [BGLM11] propose reciprocal velocity obstacles(RVO) and optimal reciprocal collision avoidance(ORCA) to search for efficient navigation for intelligent multi-agents scenes based on velocities. Other methods take visual information, auditory information, and psychological factors into account. [vTGG*20] proposes a novel framework that optimizes a cost function in a velocity space. By combing a particular cost function, different algorithm can be translated to this framework. Microscopic rule-based approaches have the advantage of high scenario generalization and strong stability. However, they regard agents responding to their surroundings in the same way, which loses the different characteristics of agents.

2.2. Data-driven Approaches

Data-driven approaches extract features from real data to guide crowd motion. These approaches generally focus on how to imitate real trajectories' patterns and reproduce natural behavior. These approaches can also be divided into macroscopic data-driven approaches and microscopic data-driven approaches.

Macroscopic data-driven approaches extract features including global velocity field, flow field, or spatial-temporal distribution to guide virtual agents' motion. [KBB*16] [CDG18] attribute virtual agents into various clusters or streams learned from crowd videos to imitate various crowd motion. [ZCLZ16] statistically analyzes the crowd flow and establish a velocity field at every exit to guide virtual agents' motion in the same scenes. [HXZW20] decomposes real-world crowd trajectories into a set of modes to regenerate crowd motion. Macroscopic data-driven approaches have strong scene dependence. When the virtual scene corresponds with the real scene, they can fast generate realistic simulation results. However, they fail to work when the virtual scene is altered.

Microscopic data-driven approaches learn the motion regularity of a single agent's motion from data instead of overall distribution in the scene. Trajectory reusing methods rearrange trajectory pieces in real data to generate virtual agnets' trajectories. [LCHL07] divides agent motion features into high-level and low-level to predict crowds motion in different scopes. [LCL07] [LFCCO09] separate short trajectory segments from the real trajectories and construct an example dataset for virtual agents to imitate. [CC14] further improves the searching efficiency of the trajectory example dataset by encoding virtual agent's state into Temporal Perception Pattern. [ZCT18] considers the influence of the time to collision to generate natural trajectories. [BKM15] proposes a trajectory fusion method to merge crowd's behavior from multiple videos. [WLZX18] [YZLL20] use real data to train a neural network that takes the agent's current state as input and outputs its motion at the next time step. [RXX*19] proposes Heter-Sim, which selects imitated objects in the dataset based on velocity. These methods can generate realistic results owing to utilizing the real data,

however they suffer from collision problems and may fail when the generation scene are different with the data scene.

3. Fusion Agent Motion Model

This section introduces our fusion crowd motion model which adopts both rule-based and data-driven modelling ideas and integrates data with dynamics, personality with common in crowd simulation. Specifically, we model the motion of each agents as a result of influence from the target point, other agents, obstacles, and the personalized motion features. And define these influence into four forces which drive the agent step by step to its target. Definitions of the first three forces, which describe the common parts of crowd dynamics, are mainly inherited from Power Law model [KSG14], which studies interaction dynamics of crowd from real data by a statistical-mechanical approach. Thus, PowerLaw can statistically reflect the basic collision-avoid, destination-driven and other basic intention in crowd motion. As a result, our fusion model matches the statistical laws of real crowd motion very well and could generates collision-free interaction of agents. In order to generate diverse and natural motion, we design the forth force to describe personality parts in the crowd motion. Specifically, in our model, the forth force pushes agents moving a little deviation from the motion defined by Power Law model. Such deviation do not affect the agent's overall movement but make the agents different from each other and show diversity of crowd. In our fusion model, the forth force is not defined as a formulaic rule. It is acquired from real data for each agent dynamically and respectively.

3.1. Agent Motion Model

Agent movements in our model are regarded in two-dimension and agents are regarded as discs of the radius r with uniform mass distribution. In our proposed model, the agent's position at the next frame $\vec{x_{t+1}}$ is calculated by its position $\vec{x_t}$ and velocity $\vec{v_t}$ at the current frame .

$$\vec{x}_{t+1} = \vec{x}_t + \vec{v}_t \cdot \Delta t \tag{1}$$

We use dynamics to model agent velocity change, which considers current velocity $\vec{v_t}$ is decided by its previous velocity $\vec{v_{t-1}}$ and acceleration $\vec{a_{t-1}}$. Furthermore, the acceleration $\vec{a_{t-1}}$ is determined by force applied to the agent $\vec{f_{t-1}}$.

$$\vec{v}_t = \vec{v}_{t-1} + \vec{a}_{t-1} \cdot \Delta t \tag{2}$$

$$\vec{a}_{t-1} = \frac{\vec{f}_{t-1}}{m} \tag{3}$$

where m is the mass of the target agent.

As mentioned above, we model force in agent movement in four aspects: goal driving force $\vec{f_d}$, agent interaction force $\vec{f_n}$, obstacle interaction force $\vec{f_o}$, and the personalized motion feature force $\vec{f_p}$.

$$\vec{f} = \vec{f}_d + \vec{f}_n + \vec{f}_o + \vec{f}_p \tag{4}$$

Through these forces, we can explicitly describe how the internal and external factors influence agent movement. (Figure 1)

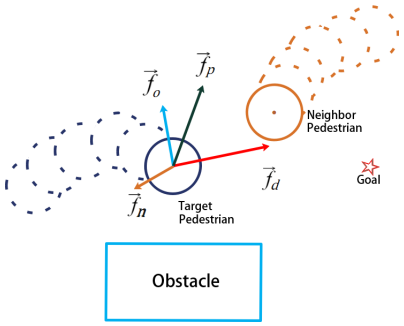

Figure 1: *Force in agent movement*

3.2. Force Calculation

Goal driving force Agents will choose a *preferred velocity* $\vec{v_p}$ pointing to their goals if there is no other interference in the scene. The goal driving force $\vec{f_d}$ is exerted when the agent's current velocity $\vec{v}$ is not equal to the agent's preferred velocity and tries to adjust the current velocity toward the preferred velocity.

$$\vec{f}_d = (\vec{v_p} - \vec{v})/\zeta \tag{5}$$

where ζ is the parameter which indicates the strength of goal driving force. Unlike [KSG14], the agent's preferred velocity is obtained from the real data, which will be further discussed later.

Agent interaction force We model the influence of other agents in the certain perceived distance into agent interaction force to avoid possible collisions. We apply the spatial gradient on the interaction energy between the agents referring to Power Law model [KSG14](which is called anticipatory force in Power Law), which founds the interaction energy between agents statistically follows a power law based on agents' projected time to a potential future collision. In this circumstance, agent interaction forces only exist when a potential future collision is predicted. To predict a potential future collision, we first calculate the relative velocity $\vec{v_{ij}}$ between agent i and agent j :

$$\vec{v_{ij}} = \vec{v_i} - \vec{v_j} \tag{6}$$

Then, we move the disc of agent i along the direction of relative velocity. If the disc does not collide with the disc of the agent j, there is no potential future collision between i and j. Otherwise, there exists a potential future collision, and the projected time τ is calculated as:

$$\tau = \frac{\vec{d_c}}{\vec{v_{ij}}} \tag{7}$$

where $\vec{d_c}$ is the distance between the current position of agent i and the position when the potential collision happens. The agents who have the potential future collision with agent i together apply the agent interaction force on agent i.

$$\overrightarrow{f_n} = \sum_{j(\neq i)} \overrightarrow{f_{ij}} \tag{8}$$

$$\overrightarrow{f_{ij}} = -\nabla E_{ij} \tag{9}$$

$$E_{ij} = \frac{k}{\tau^2} e^{-\frac{\tau}{\tau_0}} \tag{10}$$

where $\vec{f_n}$ is the resultant agent interaction force, τ_0 is the threshold of the projected time of the potential future collision, k is the parameter that indicates the strength of agents' interaction, $j(\neq i)$ represents all agents that exist potential future collision with agent i in the perceived distance.

Obstacle interaction force The calculation of obstacle interaction force is familiar with the agent interaction force. The model considers every line segment in the scene as an independent obstacle. Similarly, to predict the potential future collision, we move the disc of agent i along the direction of its velocity v_i. If a potential future collision with the obstacle is predicted, the projected time τ is calculated as:

$$\tau = \frac{\vec{d_c}}{\vec{v_i}} \tag{11}$$

where $\vec{d_c}$ is the distance between the current position of agent i and the position when the collision happens. The obstacles which have the potential future collision with agent i apply the obstacle interaction force on agent i.

$$\vec{f_o} = \sum_w \vec{f_{iw}} \tag{12}$$

$$\vec{f_{iw}} = -\nabla E_{iw} \tag{13}$$

$$E_{iw} = \frac{k}{\tau^2} e^{-\frac{\tau}{\tau_0}} \tag{14}$$

where $\vec{f_o}$ is the resultant agent interaction force, w is all obstacles that exist potential collision in the perceived distance of agent i.

Personalized motion feature force When we generate agent movements via the three forces mentioned above, the simulation results will differ from the real data. The main reason is that these rule-based methods only model the common motion regularities. However, in real agents' movement, personalized factors also play an essential role. To model these implicit factors, we use data-driven methods to construct a real agent motion example dataset e. The model would find a matching example e in the dataset during the simulation process based on the target agent's current state q. The personalized motion feature force is a part of the example e. The following section will discuss the structure of the example, the representation of the agent's state, and the matching function.

4. Construction of the Example Dataset

This section introduces how to extract features from real data and construct the agent motion example dataset. The real data used in our work include SNU [LCHL07], RCS [ZKSS12], ETH [PESVG09], and UCY [LFCCO09].

4.1. Example Construction

In dataset, every example corresponds to a trajectory piece of a real pedestrian and includes its personalized motion features and state features during the time period the trajectory piece occurred. The personalized motion features α describe the motion characteristic of the given agent, including the goal point $\vec{x_D}$, the magnitude of the preferred velocity $|\vec{v_P}|$, and the sequence of characteristics differences F_P. The state features S describe the states of agents and are used to match the suitable example for generating movements of crowd, including the sequence of positions X, the magnitude of initial velocity $|\vec{v_S}|$, and the sequence of density map P. Each example is a (α, S) pair which means in the given state as S, the real agent has exhibited personalized motion feature as α.

4.2. The Calculation of Personalized Motion Features

Goal Point We set the initial position of the trajectory piece as origin, the direction of initial velocity as the positive direction of the axis. In the following section, we use this local coordinate as default. The goal point of $\vec{x_D}$ is set as the position a period after the corresponding segment of agents trajectory.

$$\vec{x_D} = \overrightarrow{x_{t_{T'}}} \tag{15}$$

where T' is a certain time after the end of the example.

The magnitude of preferred velocity We calculate the average speed of the agents as the magnitude of preferred velocity $|\vec{v_P}|$:

$$|\vec{v_P}| = \frac{\Sigma_T |\vec{v_t}|}{T} \tag{16}$$

$$|\vec{v_t}| = \frac{|\vec{x_{t+1}} - \vec{x_t}|}{\Delta t} \tag{17}$$

where $\vec{v_t}$ is the velocity of the agent at frame t, T is the total frames of the example, $\vec{x_{t+1}}$ and $\vec{x_t}$ are adjacent position coordinates in the trajectory piece.

The sequence of characteristics The sequence of characteristics differences F_P consists of the personalized motion feature force $\vec{f_{pt}}$ at frame t in the example. In our model, the movements of the crowd are generated under four forces. The rule-based forces $\vec{f_d}$, $\vec{f_n}$ and $\vec{f_o}$ can directly calculated as Section 3.2. Then, the personalized motion feature force $\vec{f_p}$ can be reversely calculated by the resultant force $\vec{f_R}$.

$$\vec{f_{pt}} = \vec{f_{Rt}} - (\vec{f_{dt}} + \vec{f_{nt}} + \vec{f_{ot}}) \tag{18}$$

Based on the kinematics model, we can mathematically get $\vec{f_R}$ from real data:

$$\frac{\vec{x_{t+1}} - \vec{x_t}}{\Delta t} = \vec{v_t} + \frac{\vec{f_{Rt}}}{m} \cdot \Delta t \tag{19}$$

The sequence of characteristics differences F_P is composed of $\vec{f_{pt}}$ calculated at each frame. This sequence represents the deviation between real crowd movements and movements generated by rule-based forces. Therefore, when generating crowd motion, adding these characteristic differences will improve the fidelity of movements. Furthermore, the sequence structure will preserve the sequential and temporal characteristics of the real data, which is helpful to reproduce the temporal regularities in crowd movements.

4.3. The Calculation of State Features

Agents have different moving behaviors under different states, including their historical movements and their surroundings. Before simulating movements for crowd, we need to match proper personalized motion features for agents based on their states. We use the sequence of position and the magnitude of initial velocity to represent the agent's self-motion states, then use the density map sequence to represent the agent's surrounding states.

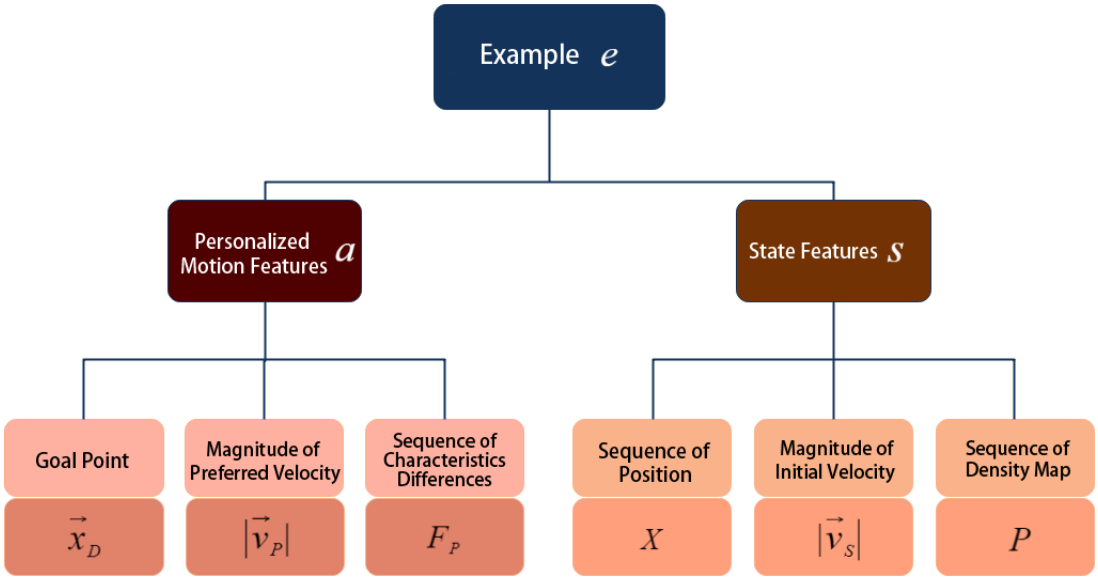

Figure 2: *Structure of the example*

The sequence of position The sequence of position X is a set of position at each frame $\vec{x}_t$ in the example.

$$X = \{\vec{x}_t | t \in [0,T), t \in N\} \tag{20}$$

The magnitude of initial velocity The magnitude of initial velocity $|\vec{v}_S|$ is defined as the speed of the frame before the example begins. Using this feature ensures speed continuity at the beginning of the example.

$$|\vec{v}_S| = \frac{|\vec{x}_0 - \vec{x}_{-1}|}{\Delta t} \tag{21}$$

The sequence of density map We abstract agent's surrounding states into the density map ρ_t, and the sequence of density map P represents the surrounding states of the agent at every frame in the example. Every density map consists of $L * L$ grids; the center of the density map is agent's position at the first frame, and the length of the grid is d. Here d and L are predefined constants. The grid in the density map has a density that represents distributions of other agents and obstacles. The mathematical presentation of the sequence of the density map is as follow:

$$P = \{\rho_t | t \in [0,T), t \in N\} \tag{22}$$

$$\rho_t = \{d_{mn} | m,n \in [1,L],\ m,n \in N\} \tag{23}$$

$$d_{mn} = d_{a_{mn}} + \varepsilon d_{o_{mn}} \tag{24}$$

where d_{mn} is the density of the grid, $d_{a_{mn}}$ is the density of agents, $d_{o_{mn}}$ is the density of the obstacles, ε is the weight to adjust two densities.

For the density of the agents $d_{a_{mn}}$ in the grid, we first find all neighbor agents in the density map at the current frame. Then we calculate the relative distances between neighbor agents and the center of the grid. The density of agents in the grid is calculated through the gaussian distribution of the relative distances.

$$d_{a_{mn}} = \sum_{j=1}^{j=N} \frac{1}{\sqrt{2\pi\sigma}} exp(-\frac{l_j^2}{2\sigma^2}) \tag{25}$$

where l_j is the distance between agent j and the center of the grid (m,n). (Figure 3)

The calculation of the density of the obstacles $d_{o_{mn}}$ is similar; however, we need to map the obstacles line segments into the grids. We consider that the obstacles have the same influence on the grid once they are within the grid. The density of obstacles in the grid is calculated as:

$$d_{o_{mn}} = \sum_{k=1}^{k=N} \frac{1}{\sqrt{2\pi\sigma}} exp(-\frac{l_k^2}{2\sigma^2}) \tag{26}$$

where l_k is the distance between the center of the grid which have obstacle in it and the center of the grid (m,n). (Figure 4)

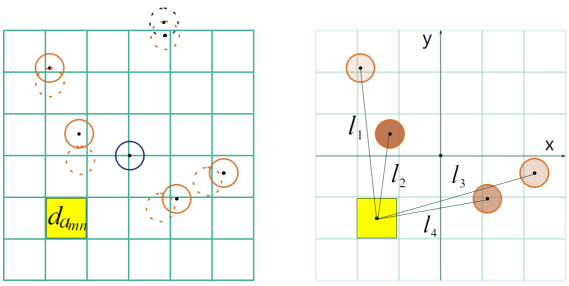

Figure 3: *Calculation of agent density of the grid*

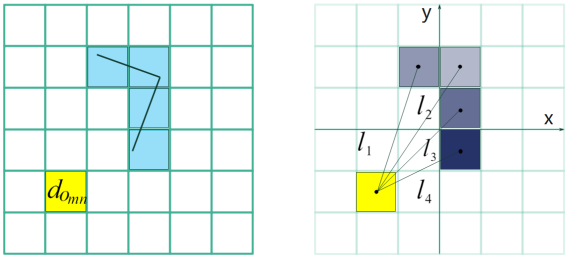

Figure 4: *Calculation of obstacle density of the grid*

5. Agent Motion Generation

5.1. Position Prediction

We match suitable examples for agents based on the state features to get personalized motion features. To calculate the sequence of the density map, we need to know the position sequences of all agents in the scene. However, they are unknown at the begin of a simulation or when the matching example needs to be renewed or ended. We predict the future positions for such agents using the linear velocity.

After predicting the future positions for every agent, we can calculate the rest state features of the agents as Section 4.3, including the magnitude of initial velocity and the sequence of the density map.

5.2. Example Match

We need to match an example for the virtual agent in three situations: no matching example for the agent, the matching example has reached its ending, the deviation between the generating position and the position in the example is too large. When matching an example in the example dataset, we calculate the similarity score between the state features of the virtual agent and every example in the example dataset. The similarity score Sim is consisted of two parts: the score of the sequence of density maps Sim_P and the score of the magnitude of initial velocity Sim_V:

$$Sim = Sim_P + \mu \cdot Sim_V \tag{27}$$

where μ is the parameter of weight.

The score of the sequence of density maps Sim_P is the weighted sum of all density deviation between all grids of the current sequence and the example sequence:

$$Sim_P = \sum_{N,T} |d_{smnt} - d_{dmnt}| \cdot \lambda_{mnt} \tag{28}$$

where d_{smnt} and d_{dmnt} are the densities of grid (m,n) at frame t in the current sequence and the example sequence, N is the number of the grid, T is the length of the sequence. The grids have different influences on the score, which we use a weight parameter λ_{mnt} to represent. On the space scale, the grid closer to the center has a higher weight. In time scale, the grid in the earlier frame has a higher weight. The mathematical expression is:

$$\lambda_{mnt} = (\frac{T-t}{T}) \cdot (\frac{K-l}{K}) \tag{29}$$

where t is the number of the current frame, T is the length of the sequence, K is the distance between the furthest grid and the center grid, l is the distance between the target grid and the center grid.

The score of the magnitude of initial velocity Sim_V is calculated as:

$$Sim_V = ||\overrightarrow{v_{Ss}}| - |\overrightarrow{v_{Se}}|| \tag{30}$$

The matching example is chosen randomly among several examples with the top similarity score. Worth noting that we apply an additional detection mechanism on the example to avoid that the deviation grow larger as time goes by. We set a deviation threshold d_l, and calculate the deviation between the virtual agent sequence and the example sequence. When the deviation is larger than d_l, we need to rematch a new example for the agent.

The implementation details of our approach is shown in the supplementary material. The design of the example dataset and the motion generation workflow is to insure the added personalized motion features could generate diverse agent motion matching the macro and micro laws of real crowd, while not causing new collisions.

6. Experiment Results

We conduct quantitative and qualitative experiments to evaluate the proposed method. The datasets used in the experiment include SNU [LCHL07], RCS [ZKSS12], ETH [PESVG09], and UCY [LFCCO09]. We compare our method with the advanced rule-based method Power Law [KSG14] and the advanced data-driven method Heter-Sim [RXX*19]. Furthermore, user studies are conducted shown in supplementary material to prove the superiority of our approach.

6.1. Quantitative Experiment

We randomly select 50% of each dataset to establish the example dataset, which is used for agent motion generation. The rest 50% of data are used as ground truth to calculate the quantitative metrics and provide initial states for the virtual agent motion generation. We introduce five metrics referring to [ZCT18] to quantify crowd's motion in various aspects, including collision rate, average speed, average vertical deviation distance, average speed change, and average angle change. The values of all the parameters are provided in the supplementary material for reproducibility.

Average speed is the average magnitude of the velocity of all agents at every timestep:

$$V = \frac{1}{I} \cdot \sum_{I} \frac{\sum_{T_i}(|\vec{x_t} - \vec{x_{t-1}}|)/\Delta t}{T_i} \tag{31}$$

where T_i is the lasting frame of agent i, I is the total number of agents in the scene.

Average speed change is the average of the change of the magnitude of the velocity of all agents at every timestep:

$$S = \frac{1}{I} \cdot \sum_{I} \frac{\sum_{T_i}(|\vec{v_t}| - |\overrightarrow{v_{t-1}}|)/\Delta t}{T_i} \tag{32}$$

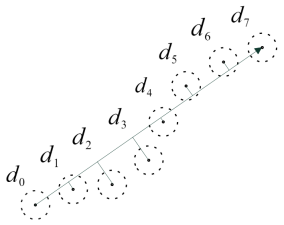

Figure 5: *Distance of vertical deviation*

where $\vec{v_t}$ and $\overrightarrow{v_{t-1}}$ are agent's velocities at time t and $t-1$.

Average angle change is the average of the change of the direction of all agents at every timestep:

$$A = \frac{1}{I} \cdot \sum_{I} \frac{\sum_{T_i} |\vartheta_t - \vartheta_{t-1}| / \Delta t}{T_i} \tag{33}$$

where ϑ_t and ϑ_{t-1} are agent's directions at time t and $t-1$.

Distance of vertical deviation is the distance between the agent and the line from the agent's initial position to its final position, as Figure 5. This metric has a strong relationship with the shape of trajectory. Average distance of vertical deviation is the average of the distance of vertical deviation of all agents at every timestep:

$$L = \frac{1}{I} \cdot \sum_{I} \frac{\sum_{T_i} d_t}{T_i} \tag{34}$$

We use these four quantitative metrics to evaluate whether the simulation results match the macro and micro laws of real crowd. The collision rate indicates collision avoidance ability and is calculated using the number of agents who have collisions divided by the total number of agents. When the distance of two agents' centers is less than the sum of their radius, they are considered collision agents. Our method and Power Law can generate collision-free motion, while the data-driven approach Heter-Sim has a higher collision rate (Figure 6). The result proves that our method combining the dynamics method can effectively solve the collision problem, which is common in most data-driven methods.

As shown in Figure 7 and Table 1, Power Law has an approximately equal average speed in all scenes and cannot fit the real data because its parameters are artificially predefined. Our approach performs similarly compared with Heter-Sim in average speed. The result of Heter-Sim is a little closer to the real data because it directly obtains velocity from the real data. However, our approach does not explicitly use the speed of real data and achieve a similar average speed compared with real data, which proves the effectiveness of our agent motion model.

	Our approach	Heter-Sim	Power Law
RCS	0.12	**0.04**	0.36
SNU(one way)	0.11	**0.01**	0.39
SNU(stroll)	0.23	**0.09**	0.91
SNU(stagger)	0.27	**0.12**	0.70
UCY(zara1)	0.13	**0.06**	0.16
UCY(zara2)	0.11	**0.02**	0.16
UCY(univ)	0.45	**0.28**	0.51
ETH(eth)	0.17	**0.03**	0.88
ETH(hotel)	0.09	**0.08**	0.09

Table 1: *Error table of average speed (m/s)*

Considering average speed change in Figure 8 and Table 2, the rule-based Power Law has a small average speed change regardless of the scene. Through Heter-Sim use the velocity in real data, our approach still outperforms Heter-Sim in most scenes because Heter-Sim fails to model the continuity of speed in the time dimension. The use of examples with a time span can smoothen the speed of agents, which helps generate realistic motion.

	Our approach	Heter-Sim	Power Law
RCS	0.18	**0.12**	0.31
SNU(one way)	**0.07**	0.23	0.45
SNU(stroll)	**0.07**	0.09	0.17
SNU(stagger)	**0.06**	0.09	0.21
UCY(zara1)	0.06	**0.04**	0.15
UCY(zara2)	**0.06**	0.15	0.13
UCY(univ)	0.04	**0.03**	0.17
ETH(eth)	**0.28**	0.46	0.62
ETH(hotel)	0.15	**0.12**	0.42

Table 2: *Error table of average speed change (m/s^2)*

In Figure 9 and Table 3, our approach still performs better than Heter-Sim and Power Law. If there is no expected collision, the Power Law will not proactively change the direction of the agent, but this does not accord with the psychological and social factors in agents. Noticing that in the scene, which exists many small turns like SNU(stroll) and SNU(stagger), the advantage of our approach becomes more apparent, which proves that our feature extraction method can effectively model the local characteristics in agent motion.

	Our approach	Heter-Sim	Power Law
RCS	**0.09**	0.17	0.36
SNU(one way)	**0.06**	0.09	0.18
SNU(stroll)	**0.25**	0.66	0.83
SNU(stagger)	**0.28**	0.47	0.57
UCY(zara1)	0.08	**0.04**	0.07
UCY(zara2)	**0.10**	0.14	0.15
UCY(univ)	**0.09**	0.30	0.18
ETH(eth)	**0.03**	0.23	0.26
ETH(hotel)	**0.22**	0.33	0.46

Table 3: *Error table of average angle change(rad/s)*

Figure 10 and Table 4 show that our approach achieves the best performance when Heter-Sim is better than Power Law. The result indicates that our approach can regenerate the shapes of realistic trajectories because we combine the characteristics of each agent and surrounding states at each timestep.

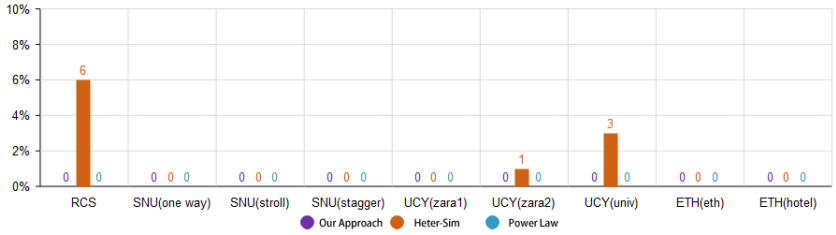

Figure 6: *Collision rate results*

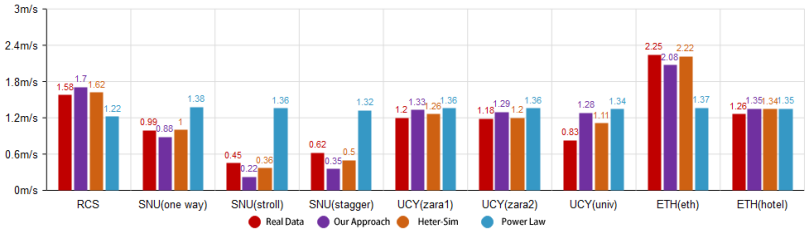

Figure 7: *Average speed results*

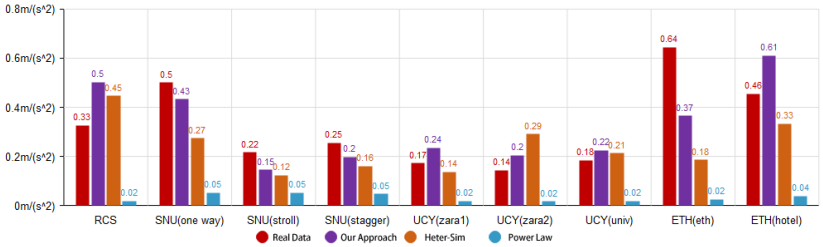

Figure 8: *Average speed change results*

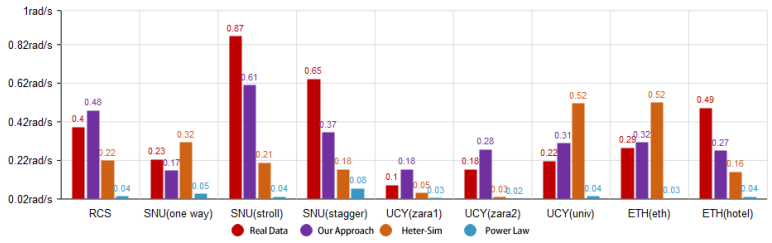

Figure 9: *Average angle change results*

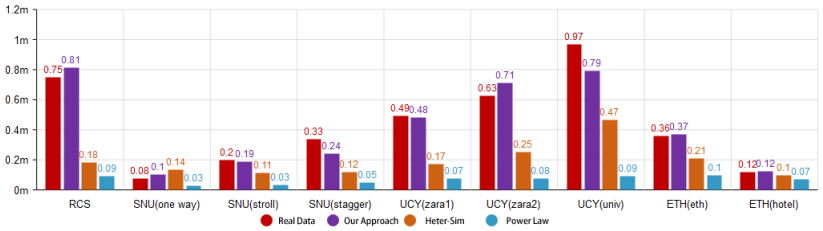

Figure 10: *Average vertical deviation distance results*

	Our approach	Heter-Sim	Power Law
RCS	**0.06**	0.57	0.66
SNU(one way)	**0.02**	0.06	0.05
SNU(stroll)	**0.01**	0.09	0.16
SNU(stagger)	**0.09**	0.22	0.29
UCY(zara1)	**0.01**	0.32	0.42
UCY(zara2)	**0.08**	0.38	0.55
UCY(univ)	**0.17**	0.50	0.87
ETH(eth)	**0.01**	0.15	0.26
ETH(hotel)	**0.00**	0.02	0.05

Table 4: *Error table of vertical deviation distance(m)*

6.2. Qualitative Experiments

The performance of visualization is vital for agent motion modeling. We conduct several qualitative experiments to evaluate different approaches further.

First, we conduct experiments on generating trajectories. The red circle represents the target agent, and grey and blue circles represent the neighbor agents at the first frame in the scene. As is shown in the Figure 11 12 13 14, the trajectory generated by our approach is closest to the real data. Power Law tends to generate straight trajectories if there is no predicted collision. Our approach establishes the connection between agents' personalized motion features and state features, thus generating better trajectories based on their surrounding situations. Moreover, in Figure 14, the trajectory of Heter-Sim appears turbulence in the opposite scene, while our approach uses the dynamics method to model agent motion to generate smooth trajectories in complex situations. The reason that the data-driven Heter-Sim is less real than our approach is partly because the limit of dataset completeness and selecting rules.

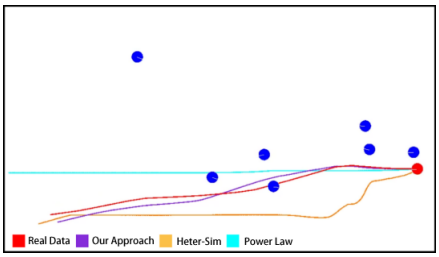

Figure 11: *Comparison of individual agent trajectories*

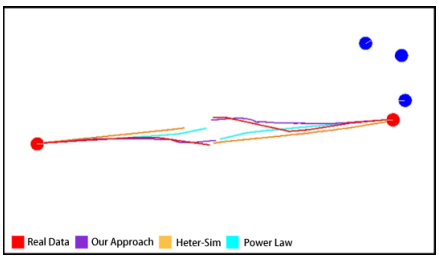

Figure 12: *Comparison of opposite agent trajectories*

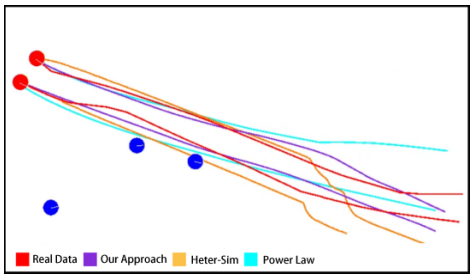

Figure 13: *Comparison of parallel agent trajectorie*

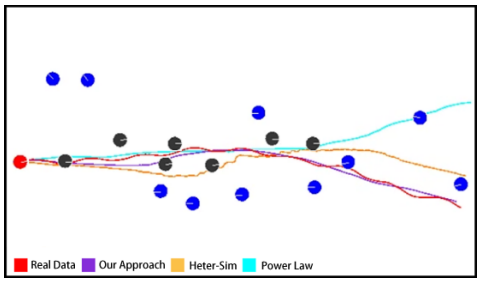

Figure 14: *Comparison of individual agent trajectories in opposite scene*

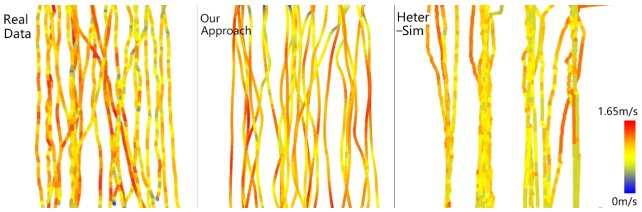

Figure 15: *Results of using one way dataset*

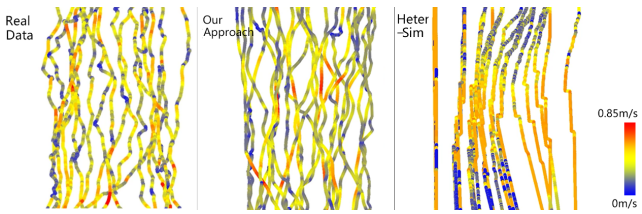

Figure 16: *Results of using stroll dataset*

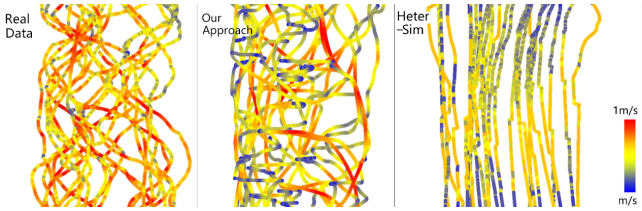

Figure 17: *Results of using stagger dataset*

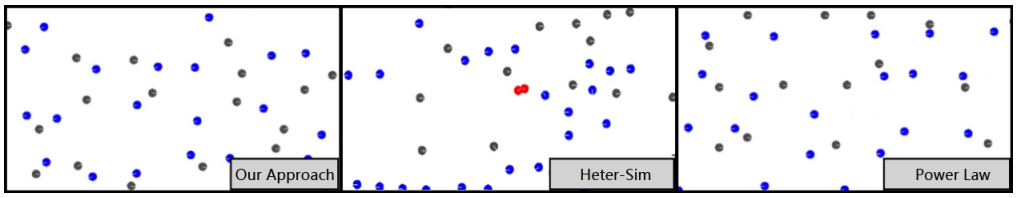

Figure 18: *Results of scaling up agent. The grey dots and blue dots represents two groups of agents moving in the opposite direction, the red dots indicate the agent is collide with another agent. The meaning of the colored dots is same in Figure 19 and 20.*

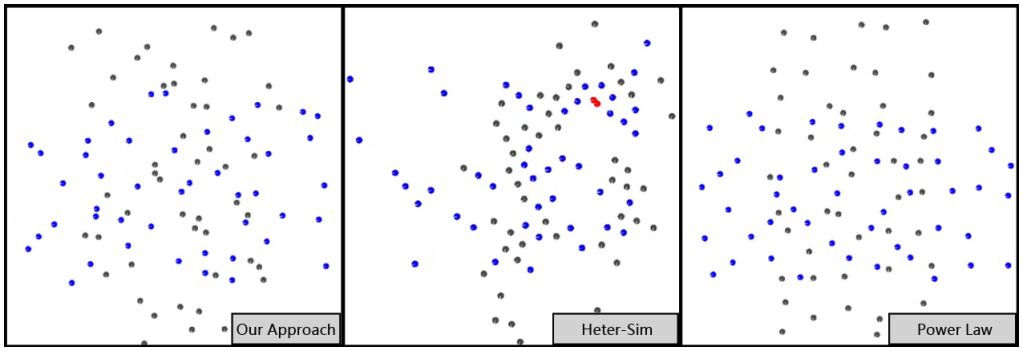

Figure 19: *Results of changing the direction of motion*

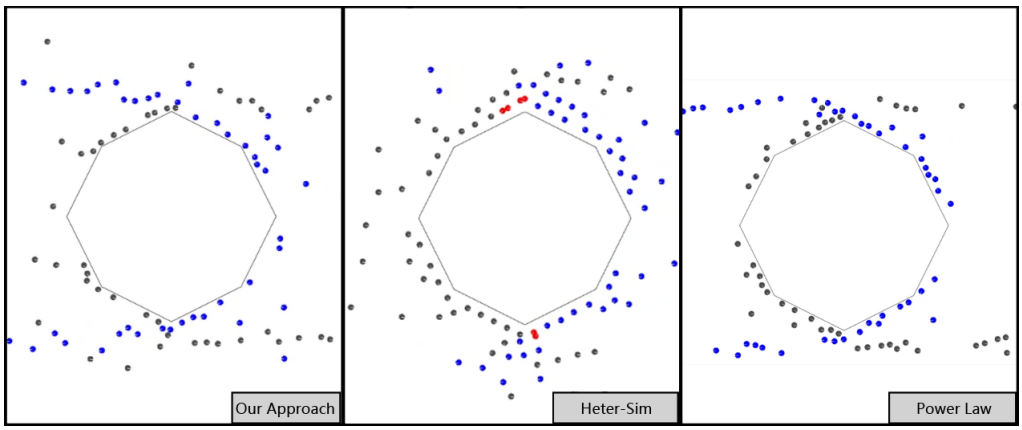

Figure 20: *Results of adding obstacles*

Secondly, we conduct experiments on restoring the data characteristics. This experiment is designed to visualize the approach's ability to extract personalized features from different datasets. We separately select 50% data in SNU(one way), SNU(stroll), and SNU(stagger) as real data for our approach and Heter-Sim. These datasets in SNU have different complicated characteristics. A good approach should regenerate similar motion patterns with these characteristics after selecting a particular example set. We use each training set as an example dataset to generate agent motion based on the initial state of the real data, then compare the generated movement with the real data. As shown in Figure 15 16 17, our approach outperforms Heter-Sim in the shape of trajectories and the velocity distribution during the agent's motion. To some extent, Heter-Sim can restore the distribution of velocity, but it fails to restore trajectories' shape. Compared with Heter-Sim, our approach appears to tend to restore the dataset's characteristics in all three scenes.

	Scaling Up	Direction Changing	Adding Obstacles
Our Approach	0	0	0
Heter-Sim	0.75	0.61	0.53
Power Law	0	0	0

Table 5: *Collision rate in different scenes*

Last, to demonstrate that the proposed personalized features have relatively low scenario dependency and could be transplant to dif-

ferent crowd scenes by our fusion model, we conduct the experiments of scene variation. We change the scene in the RCS dataset differently and use different approaches to generate agents' motion. We increase the number of agents that walk in opposite directions in every row from approximately 2/3 to 5/6. Then based on the increasing scale situation, we change the opposite direction into the vertical direction. Finally, we add obstacles in the center of the scene. As is shown in the Figure 18 19 20 and Table 5, our method can generate natural and collision-free agent motion. Power Law can generate collision-free motion, but the motion is inflexible. Heter-Sim results in many collisions, especially when adding obstacles in the scene, and causes unusual disturbances in agents' motion. Overall, the result proves that our approach has the good ability of collision avoidance and can run well in new scenes which has different layout, crowd density and motion trend.

7. Conclusion

This paper proposes a fusion crowd motion modeling method which combines data-driven method and crowd dynamics method into a force driven framework. We model the main influence factor of agent motion by four forces, including goal driving force, agent interaction force, obstacle force, and the personalized motion force. The first three forces are calculated by rules mainly inherited from Power Law model [KSG14] to ensure our method matches the macro laws of real crowd laws and could generates collision-free motion. The personalized motion force is from real data. An example dataset construction method is proposed to extract personalized motion force from real data. And a crowd motion generation workflow is proposed to sophisticatedly and dynamically select examples for each agents. So that our method could generate diverse and natural agent motion matching the micro laws of real crowd perfectly.

In the future, some works can be done to improve the performance of crowd motion modeling. First, we model the crowd motion in microscopic scale, which may ignore the overall distribution of the crowds. Introducing global features into the method can improve the macroscopic performance of crowd motion. Second, the structure of example dataset can be modified as [CC14] and optimize the searching algorithm to improve the efficiency of matching examples. Finally, the example dataset is pre-constructed before generating crowd's motion. A self-adaptive approach that chooses proper motion features from different datasets during generating crowd's motion can make the modeling method more usable. Moreover, deep-learning based approaches is also a plausible way to improve the framework by automatically extracting proper features.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1710400, in part by the Major Program of National Natural Science Foundation of China under Grant 91938301, in part by the Youth Program of National Natural Science Foundation of China under Grant 62002345, and in part by the Innovation Program of Institute of Computing Technology Chinese Academy of Sciences under Grant E261070.

References

[BA99] BLUE V., ADLER J.: Bi-directional emergent fundamental pedestrian flows from cellular automata microsimulation. *Transportation and traffic theory 14* (1999), 235–254. 2

[BEA97] BLUE V., EMBRECHTS M., ADLER J.: Cellular automata modeling of pedestrian movements. In *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation* (1997), vol. 3, Ieee, pp. 2320–2323. 2

[BGLM11] BERG J. V. D., GUY S. J., LIN M., MANOCHA D.: Reciprocal n-body collision avoidance. In *Robotics research*. Springer, 2011, pp. 3–19. 2

[BKM15] BERA A., KIM S., MANOCHA D.: Efficient trajectory extraction and parameter learning for data-driven crowd simulation. In *Proceedings of the 41st Graphics Interface Conference* (2015), pp. 65–72. 2

[BSK16] BARNETT A., SHUM H. P., KOMURA T.: Coordinated crowd simulation with topological scene analysis. In *Computer Graphics Forum* (2016), vol. 35, The Eurographs Association & John Wiley & Sons, Ltd. Chichester, UK, pp. 120–132. 2

[CC14] CHARALAMBOUS P., CHRYSANTHOU Y.: The pag crowd: A graph based approach for efficient data-driven crowd simulation. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 95–108. 2, 11

[CDG18] CHENG Q., DUAN Z., GU X.: Data-driven and collision-free hybrid crowd simulation model for real scenario. In *International Conference on Neural Information Processing* (2018), Springer, pp. 62–73. 2

[GM85] GIPPS P. G., MARKSJÖ B.: A micro-simulation model for pedestrian flows. *Mathematics and computers in simulation 27*, 2-3 (1985), 95–105. 2

[HFV00] HELBING D., FARKAS I., VICSEK T.: Simulating dynamical features of escape panic. *Nature 407*, 6803 (2000), 487–490. 2

[Hug00] HUGHES R.: The flow of large crowds of pedestrians. *Mathematics and Computers in Simulation 53*, 4 (2000), 367–370. 2

[Hug02] HUGHES R. L.: A continuum theory for the flow of pedestrians. *Transportation Research Part B: Methodological 36*, 6 (2002), 507–535. 2

[Hug03] HUGHES R. L.: The flow of human crowds. *Annual review of fluid mechanics 35*, 1 (2003), 169–182. 2

[HXZW20] HE F., XIANG Y., ZHAO X., WANG H.: Informative scene decomposition for crowd analysis, comparison and simulation guidance. *ACM Transactions on Graphics (TOG) 39*, 4 (2020), 50–1. 2

[KBB*16] KIM S., BERA A., BEST A., CHABRA R., MANOCHA D.: Interactive and adaptive data-driven crowd simulation. In *2016 IEEE Virtual Reality (VR)* (2016), IEEE, pp. 29–38. 2

[KSG14] KARAMOUZAS I., SKINNER B., GUY S. J.: Universal power law governing pedestrian interactions. *Physical review letters 113*, 23 (2014), 238701. 1, 2, 3, 6, 11

[LCHL07] LEE K. H., CHOI M. G., HONG Q., LEE J.: Group behavior from video: a data-driven approach to crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), pp. 109–118. 2, 4, 6

[LCL07] LERNER A., CHRYSANTHOU Y., LISCHINSKI D.: Crowds by example. In *Computer graphics forum* (2007), vol. 26, Wiley Online Library, pp. 655–664. 2

[LFCCO09] LERNER A., FITUSI E., CHRYSANTHOU Y., COHEN-OR D.: Fitting behaviors to pedestrian simulations. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), pp. 199–208. 2, 4, 6

[MCT21] MUSSE S. R., CASSOL V. J., THALMANN D.: A history of crowd simulation: the past, evolution, and new perspectives. *The Visual Computer* (2021), 1–16. 1

[MIN99] MURAMATSU M., IRIE T., NAGATANI T.: Jamming transition in pedestrian counter flow. *Physica A: Statistical Mechanics and its Applications 267*, 3-4 (1999), 487–498. 2

[OM93] OKAZAKI S., MATSUSHITA S.: A study of simulation model for pedestrian movement with evacuation and queuing. In *International Conference on Engineering for Crowd Safety* (1993), vol. 271. 2

[PESVG09] PELLEGRINI S., ESS A., SCHINDLER K., VAN GOOL L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision* (2009), IEEE, pp. 261–268. 4, 6

[RXX*19] REN J., XIANG W., XIAO Y., YANG R., MANOCHA D., JIN X.: Heter-sim: Heterogeneous multi-agent systems simulation by interactive data-driven optimization. *IEEE transactions on visualization and computer graphics 27*, 3 (2019), 1953–1966. 2, 6

[TCP06] TREUILLE A., COOPER S., POPOVIĆ Z.: Continuum crowds. *ACM TOG 25*, 3 (2006), 1160–1168. 2

[VdBLM08] VAN DEN BERG J., LIN M., MANOCHA D.: Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE international conference on robotics and automation* (2008), Ieee, pp. 1928–1935. 2

[vTGG*20] VAN TOLL W., GRZESKOWIAK F., GANDÍA A. L., AMIRIAN J., BERTON F., BRUNEAU J., DANIEL B. C., JOVANE A., PETTRÉ J.: Generalized microscropic crowd simulation using costs in velocity space. In *Symposium on Interactive 3D Graphics and Games* (2020), pp. 1–9. 2

[vTP21] VAN TOLL W., PETTRÉ J.: Algorithms for microscopic crowd simulation: Advancements in the 2010s. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 731–754. 1

[WLZX18] WEI X., LU W., ZHU L., XING W.: Learning motion rules from real data: Neural network for crowd simulation. *Neurocomputing 310* (2018), 125–134. 2

[XJJD14] XU M.-L., JIANG H., JIN X.-G., DENG Z.: Crowd simulation and its applications: Recent advances. *Journal of Computer Science and Technology 29*, 5 (2014), 799–811. 1

[YLG*20] YANG S., LI T., GONG X., PENG B., HU J.: A review on crowd simulation and modeling. *Graphical Models 111* (2020), 101081. 1

[YZLL20] YAO Z., ZHANG G., LU D., LIU H.: Learning crowd behavior from real data: A residual network method for crowd simulation. *Neurocomputing 404* (2020), 173–185. 2

[ZCLZ16] ZHONG J., CAI W., LUO L., ZHAO M.: Learning behavior patterns from video for agent-based crowd modeling and simulation. *Autonomous Agents and Multi-Agent Systems 30*, 5 (2016), 990–1019. 2

[ZCT18] ZHAO M., CAI W., TURNER S. J.: Clust: simulating realistic crowd behaviour by mining pattern from crowd videos. In *Computer graphics forum* (2018), vol. 37, Wiley Online Library, pp. 184–201. 2, 6

[ZKSS12] ZHANG J., KLINGSCH W., SCHADSCHNEIDER A., SEYFRIED A.: Ordering in bidirectional pedestrian flows and its influence on the fundamental diagram. *Journal of Statistical Mechanics: Theory and Experiment 2012*, 02 (2012), P02002. 4, 6