IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 30, 2018, accepted September 21, 2018, date of publication October 1, 2018, date of current version October 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2872731

Toward Modeling Emotional Crowds

LEI LYU[1,2] AND JINLING ZHANG[3]
[1]Department of Information Science and Engineering, Shandong Normal University, Jinan 250014, China
[2]Shandong Provincial Key Laboratory for Novel Distributed Computer Software Technology, Jinan 250014, China
[3]School of information, Renmin University of China, Beijing 100872, China

Corresponding author: Lei Lyu (lvbu007@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61502505, Grant 61472232, and Grant 61572299, in part by the Shandong Key Research and Development Program under Grant 2017GSF20105, in part by the Major Program of Shandong Province Natural Science Foundation under Grant ZR2018ZB0419, in part by the Natural Science Foundation of Shandong Province under Grant ZR2016FB13, and in part by the Shandong Province Higher Educational Science and Technology Program under Grant J16LN09.

ABSTRACT This paper presents an Emotion Network structure for modeling crowd emotions to simulate socially emotional crowds. The emotion model focuses on entire crowds, builds individual emotion spaces as network nodes, embodies psychological linkages with relationship arcs, and clusters emotionally homogeneous crowds into EmotionTrees. According to the maximum value of the emotion component and the psychological distance, crowds are initialized as different trees. With relational distributions of crowd changes, agents move between different subgroups. Then, we construct emotion-oriented commands with subgroup constraints according to intergroup emotion theory. For execution, motion variables are calculated from emotion commands (i.e., emotion source-oriented orientation, and velocity and emotion strength-based motion clip selection). Experiments on bystanders, football fans, violent protestors, and guided marching behaviors demonstrate that the Emotion Network can improve crowd behavior modeling with particular attention to the impacts of group constraints and emotional characteristics, which can reflect environment stimuli and changes. The model can be integrated into physical crowd simulators as a high-level emotion manager, providing intelligent emotional support in improving the realism of human-like crowds.

INDEX TERMS Agent-based simulation, crowd behavior, emotional crowd, emotion network.

I. INTRODUCTION

As a requirement for many applications (e.g., advanced animations for movies, cartoons, and advertising; immersive industrial and safety training simulations; and interactive military war games and mission rehearsals) through which virtual characters behave emotionally, the problem of modeling agents with rich emotional features is a central issue of crowd simulation. Previous work based on geometric or physical crowd models has contributed to the area of crowd dynamics. Recent work on behavior modeling have focused on issues of high level cognitive functionality, but it has tended to ignore the effects of emotions in simulating crowd behavior. Several advanced emotion modeling methods for generating emotional crowds have been developed.

The state of art in emotion modeling is mainly derived from theoretical methods. A large collection of peer-reviewed studies on human emotions and performance as a function of personality differences, moods, and collective psychologies has been published. Unfortunately, the existing literature rarely addresses ways to encode such findings into methods suitable for controlling crowd behavior. Therefore, the first key focus of our method is to make use of theoretical results for the management of emotional crowds.

Almost all recently conducted simulations involving an emotion model have aimed at the presentation of individual emotional behaviors. However, in regard to crowd emotions, numerous factors complicate such simulations, such as the anonymity of individual psychologies in a crowd, relations between individuals and small groups, and conformity to intergroup emotion. Thus, our second key goal is to develop an emotion model focused on entire crowds to improve the realism of socially emotional crowds by considering interaction mechanisms functioning among individuals on an emotion level.

Based on theories of social network and intergroup emotions, we present an Emotion Network framework for simulating socially emotional crowds. It creates individual emotion spaces as network nodes, embodies psychological links between individuals as relationship arcs, and divides crowds into subgroups as EmotionTrees. EmotionTrees encode these paradigms into commands oriented by emotions and executed by tree nodes. With changes in the relational distribution of crowds and in the contagion of emotions, agents also move between subgroups.

Experiments on typical emotional crowd members, such as wandering bystanders, passionate fans, and raging protestors, demonstrate that our method can be used to simulate autonomous crowd behaviors with various emotional features. Likewise, our method can be integrated into a physical crowd simulator as a high-level emotion manager providing intelligent support to a low-level driver.

II. RELATED WORK

In this section, we give a brief overview of prior work related to crowd simulation, path planning and emotional crowd. We refer interested readers to excellent reviews [1]–[4].

A. CROWD SIMULATION

There is a great deal of research in the topic of modeling behavior of a crowd. These solutions can be divided into two diverse categories: global and local approaches.

In the global models, a crowd is regarded as a continuous group and can be formulated as fluid flows. Hughes [5] proposes a continuum theory for the flow, Treuille *et al.* [6] extend the work using a dynamic potential field to integrate global navigation with moving obstacles and people, [7] focus specifically on the problem of simulating the inter-agent dynamics of large, dense crowds in real time. In addition, Chenney [8] employs flow tiles to design velocity fields ensures the smooth flow of agents, Jin *et al.* [9] proposes a simple method to generate fields for the interactive control of crowd navigation, Patil *et al.* [10] offer a goal-directed navigation field for multiple groups. These approaches are particularly suitable for high-density simulations. However, such approaches model a global optimum, whereas humans generally behave less optimally and can even get stuck in very dense situations.

Local models for agent behavior can be traced back to the seminal work of Reynolds [11], [12] by means of a set of rules and the particle system [13], [14] which resort to forces. Many future crowd simulation methods derive from the two kinds of work. They have accounted for sociological factors [15], psychological effects [16], directional preferences [17], and other models of pedestrian behavior, including [18]–[20], [54]. Local approaches are hard to properly model the behavior of agents that aim to move towards specified goals due to easily getting stuck behind an obstacle. Thus, local models are often combined with global planning techniques for practical crowd simulation which will be elaborated on the next section.

Additionally, the example-based methods to crowd simulation has recently emerged as a new way of reproducing complex crowd behaviors mimicking real crowd. The key idea is to reedit pre-recorded trajectories [21], [22] or learn the motion model [23]. These methods can maintain the realism of local trajectories, but do not generalize as well as the microscopic approaches which try to mathematically formulate the main principles by which humans move in crowds.

B. PATH PLANNING

Extensive literature exists on path planning in terms of graph, roadmap, or static potential field. Pettré *et al.* [16] uses navigation graphs of static scenes to find pedestrian paths in real-time but does not account for collisions or congestion. Early roadmap-based planning methods directly sampled roadmap nodes from space-time volume [24], [25]. More recent methods construct the roadmap by a static spatial probabilistic roadmap [26], [27]. Later, some algorithms are extended to dynamic environments, multiple agents and deformable models [28], [29]. However, they may not adapt to environments with a wealth of independent agents.

Many other methods such as the potential field have been introduced for path-planning. These methods have the ability to execute smooth paths, but requires appreciable computational resources, thus is hard to be performed in real time. Later, Treuille *et al.* [6] present a dynamic potential field simultaneously integrates global navigation with moving obstacles such as other people.

The methods mentioned above are devised to generate collision-free paths [55], but they do not take into account the physical characteristics of the agents. For this reason, the preliminary approach is proposed by Choi *et al.* [30]. In this approach, a deformable motion model is presented to allow animated characters to navigate through cluttered environments by real-time interactive navigation and path planning. But the method didn't ensure the energy efficiency of agents. But some methods highlight the computation of expended energy. Levine *et al.* [31] combine path planning in space and time with parameterized controllers by integrating various cost functions. However, the energy minimization has not yet been examined and implemented. Guy *et al.* [32] use an optimization method to compute a biomechanically energy-efficient, collision-free trajectory that minimizes the amount of effort for each heterogeneous agent in a large crowd. In [33], Anagnostopoulos use a mathematical equation that returns an approximation of the VO2 expenditure for different actions as a parameter to describe the expended energy of a character performing a given task.

C. EMOTIONAL CROWD SIMULATION

Traditional crowd simulators ignore the psychological differences between individuals. Many methods integrate emotion and psychological states into the simulation of autonomous agents [34], [35], [40]. Shao and Terzopoulos [18] incorporate perceptual, behavioral and cognitive control components into the autonomous pedestrians model. Following the study, Yu and Terzopoulos [36] build a behavioral model using decision networks upon the autonomous pedestrians model. Similar to the approach, [37] uses formal models of personality and emotions, traits are represented as nodes of decision networks.

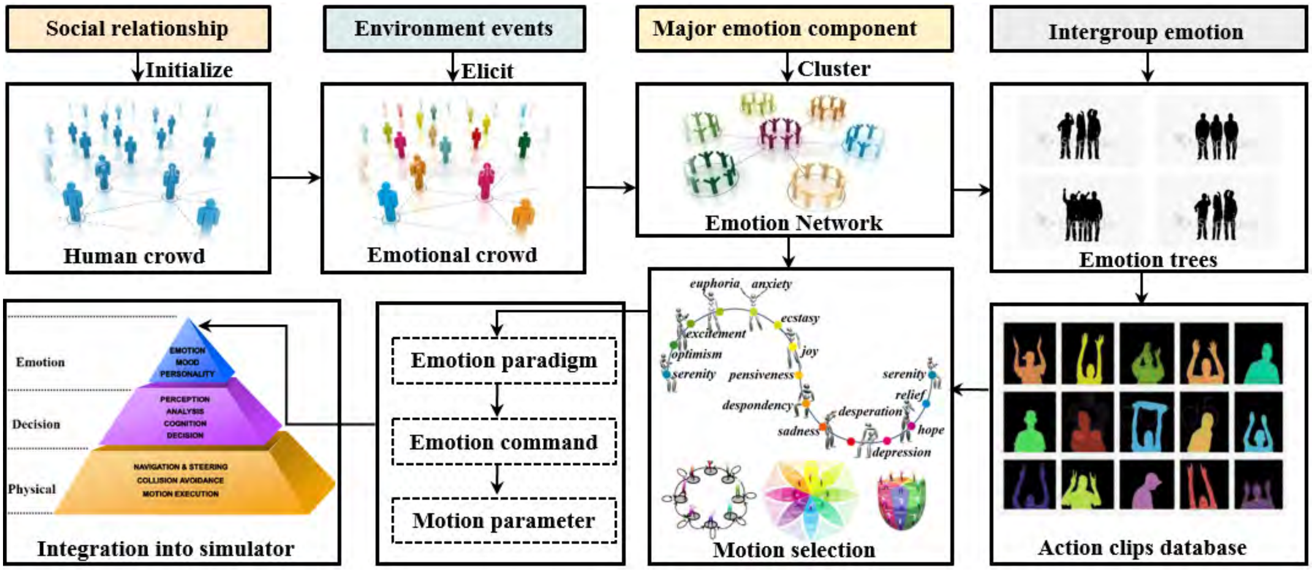

FIGURE 1. Methodological overview.

Representing a middle ground between high-level cognition and low-level physiology, emotion is central to crowd behavior research. A handful of work on emotional concepts can be categorized into the following clusters: compiling scripts for emotional behavior, mapping emotional factors based on motion parameters, and developing regulations on crowd behavior.

Representative works that compile script include the PMFserv system and the ViCrowd model [38], [39]. PMFserv abstracts normal forms from the literature and builds performance moderator functions for human behavior modeling. ViCrowd divides crowds into clusters based on degrees of autonomy, and each cluster is given its own behavior database. These methods always involve an onerous preprocess for constructing script databases and lack support from an interaction mechanism between individuals on an emotion level.

Another intuitive and directive method involves connecting emotional factors to motion parameters. The social force model [14], [41] defines an urgency/panic factor that affects motion parameters and that simulates an emergency escape. Panic element methods embedded in Cellular Automata denote the panic threshold and the tendency for imitation to represent the possibility for an agent to copy another's motion. The personality marked HiDAC model [42] maps personality traits onto crowd motion parameters and diverse motion extents for crowds. However, these methods tend focus on the effects of crowd dynamics and reveal little in terms of behaviors.

Psychological and behavioral rules are easily integrated into physical rule-based models. One simulation of the Pennsylvania Station combined cognitive rules for local steering and task scheduling for decision-making. Pan et al. allowed for the authority of guidance and aggressive and gentle propensities for several agents to simulate evacuation behaviors of leading, queuing and herding [43]. Yeh *et al.* divide these authorities into circles that could change in size and shape depending on conditions in performing different functions. However, rules fixed in scenarios are particular to special situations and are non-portable. Furthermore, mature psychology behavior models for individuals [44] have been proposed for the purpose of producing rich expressions and natural gestures. In addition, an emotion contagion method [45]–[47] for crowd evacuation has been proposed in recent years. However, due to a lack of complex interaction mechanisms among intelligent individuals, they are difficult to apply to crowds.

III. OVERVIEW

In this paper, we present an Emotion Network structure for crowd emotion modeling and emotion-oriented behavior generation. The architecture of the Emotion Network is inspired by social network theory [48] and intergroup emotion theory [49]. Problems related to emotion propagation are addressed as the contagion of energy inspired by Freud's "psychic energy" [50] and by related recently developed physiological theories [51]. This section provides an overview of our method (as is shown in Fig.1) and then presents an algorithm that integrates our novel structure into a traditional simulator.

Original crowds initialized by social relations settled randomly are input as a network structure. Emotions elicited by environmental events classify crowds into subgroups based on the maximum value of the emotional component and the psychological distance. Each subgroup is characterized by a specific emotion that increases cohesion within a group and that loosens linkages with other groups. These emotion-oriented groups are denoted as EmotionTrees that organize agent emotional spaces as nodes within. An EmotionTree reflects outside stimuli, updates its structure, and gives emotion commands to its nodes. A motion selection system is used to match emotional energy with corresponding action clips from a special clip database of a given EmotionTree. Finally, commands are executed by the agent, affecting decision-making and physical simulator implementation.

A traditional simulation system can be functionally parsed into two layers: a *physical layer* (PL) for navigation, collision prevention and motion execution, and a *decision layer* (DL) for cognition, analysis and decision-making.

The PL is denoted as:

$$\mathrm{PL} = \{\Phi_{pcs}, \Psi_{ens}\}, \tag{1}$$

in which Φ_{pcs} represents the collection of physical states of crowds simulated and Ψ_{ens} represents the environmental states of geometric entities and paths.

Likewise, the DL is built on a PL can be denoted as:

$$\mathrm{DL} = \{\Omega_{tl}, \Phi_{ics}, \Psi_{ene}|\mathrm{PL}\}, \tag{2}$$

in which Ω_{tl} represents the target list of crowds; Φ_{ics} represents the collection of intentional states of crowds, and Ψ_{ene} denotes environmental events causing crowds to change movement decisions.

We build our *emotion layer* (EL) on the two layers as:

$$\mathrm{EL} = \{\Phi_{ecs}|\mathrm{PL}, \mathrm{DL}\}, \tag{3}$$

in which Φ_{ecs} represents crowd emotional states.

We assume that during each time step, the simulator performs the following steps:

- EL updates crowd emotion states using EL.UPDATE () as explained in detail in Section III B (2).

- DL perceives external information, cognizes environmental events, analyzes internal intentions and makes decisions for a crowd denoted as function DL.UPDATE ().

- PL receives decisions from DL, searches through the geometric states of entities and paths, and applies collision avoidance to address problems of motion execution denoted as function PL.UPDATE ().

The updated function of DL computes decisions for crowds as follows:

$$\vec{v}_d \leftarrow \mathrm{DL.\,UPDATE}(\vec{v}_e, \Omega_{tl}, \Phi_{ics}, \Phi_{pcs}, \Psi_{ens}, \Psi_{ene}, \Gamma_{fsm}, \Gamma_{navigation}) \tag{4}$$

in which $\vec{v}_e$ denotes the emotion-controlled velocity computed from the emotion layer; Γ_{fsm} represents an algorithm of a finite state machine; $\Gamma_{navigation}$ is an algorithm for navigation, and decisions are simplified by the collection of expected velocity vectors for navigation.

The update function of PL calculates the final movements of crowds, which is written as:

$$\vec{v}_p \leftarrow \mathrm{PL.\,UPDATE}(\vec{v}_d, \Phi_{pcs}, \Psi_{ens}, \Gamma_{ca}), \tag{5}$$

in which $\vec{v}_d$ is the determined velocity given by the decision layer and Γ_{ca} represents an algorithm for collision avoidance that generates a collection of velocity vectors that denote the movements of crowds.

Various decision-making and cognitive mechanisms can be used to update the decision layer, and methods used to update crowd physical states can vary between simulation systems. Regardless of the mechanisms of UPDATE functions used in physical and decision layers, our emotion model is functional.

IV. EMOTION NETWORK

The architecture of the Emotion Network is composed of submodules of an EmotionTree that includes individual nodes and relationship arcs as is shown in Fig. 2.

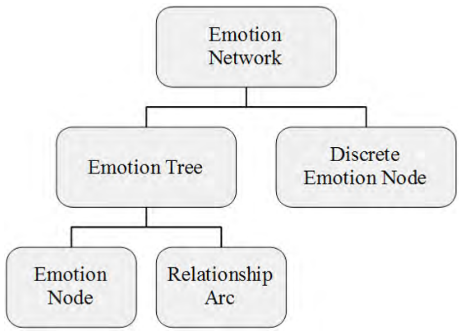

FIGURE 2. Architecture of the emotion network.

A. DEFINITION

Definitions of each sub-module are presented as follows:

EmotionTree: intergroup emotions. Birds of a feather flock together; people of the same sort are usually attracted to one another. The EmotionTree is denoted as:

$$\mathrm{EmotionTree} = \{\delta_i, \delta_e, r, \varphi, \Omega, C\}, \tag{6}$$

where

• δ_i represents a collection of internal emotion values of the group for identification, including the emotions of this tree, the threshold of emotion energy, and the period during which the emotion is experienced;

• δ_e represents a collection of external emotion values that impact the out group, including emotion concentration levels (calculated from the following equation), density, and the radius of the covered area.

$$\rho_{e_k} = \frac{1}{Area\,(convexhull)} \sum_{i \epsilon Tree(k)} e_{i_k} \tag{7}$$

• r is the root of the group, representing the uniform center, goal, and target or focused object of the group;

• φ is the collection of group members;

• Ω denotes the boundaries of groups across a geometric distribution;

• C represents the database used to store action clips clustered by linkages between specific emotions and specific types of behaviors.

Relationship Arc: psychological links between individuals. Humans are social creatures and rely on others to fulfill their social and emotional needs. The psychological and societal links between people allow for emotional perception and communication (e.g., trust in others, interpersonal perceptions, and group coherence). The Relationship Arc is denoted as

$$\text{Relationship Arc} = \{\alpha, \beta, \omega; \textit{Spring}, \textit{Propagation}\} \tag{8}$$

Where:

- α and β represent the source node and destined node, respectively. As psychological linkages are always directional, the direction of emotion contagion should be determined.
- ω is the weight of the relationship or the psychological distance. Members of a family related by kin with a uniform target or religion or belonging to the same group are more similar psychologically than strangers. Therefore, the function for computing the weight is given by

$$\begin{aligned} ArcWeight(i,j) = \sum\{&dis(i,j), uniform(i,j), dGoal(i,j),\\ &acq(i,j), pert(i,j), dMood(i,j)\}. \end{aligned} \tag{9}$$

Here *dis* (.,.) is the distance between two nodes; *uniform* (.,.) denotes whether two agents belong to the same community; *dGoal* (.,.) denotes differences in their goals; *acq* (.,.) denotes acquaintances; *pert* (.,.) denotes the pertinence of emotion states; *dMood* (.,.) denotes differences in mood states.

- *Spring* represents the flexibility of a given connection. Emotional relations psychologically deter or repulse individuals. Reflected in the geometric space is the change in distance: *stretch* (.,.) or *shrink* (.,.).
- *Propagation* denotes the process of emotion contagion. In quantifying psychological factors including subconscious and conscious processes, we develop a propagation process for the weight of relationship linkages computed from (10):

$$\begin{aligned} \Delta e_{ij_k} &= \left(e_{i_k} - e_{j_k}\right) - \frac{\mu_{e_k}}{\vartheta_{e_k}} \cdot ArcWeight\,(i,j) \\ \Delta E_{ij} &= (\Delta e_{ij_k})_{\|e_{i_k}\| > \|e_{j_k}\|}, \quad k = 0, 1, 2, 3, \end{aligned} \tag{10}$$

in which ϑ_{e_k} is the propagating speed of emotion e_k (the distance over which emotions are spread in the unit time); μ_{e_k} is the attenuating speed (the attenuated value per unit of time); Δe_{ij_k} is the remaining value after the original emotion difference propagates along the arc; and ΔE_{ij} is the propagating emotion vector from agent i to j. In addressing emotion propagation, the contributions of individuals with more emotional strength are considered, and thus emotional energy can only transfer from a higher node to a lower one.

Emotion Node: the individual emotion model is denoted as:

$$\text{EmotionNode} = \{\text{E}, \text{P}, \text{C}\}, \tag{11}$$

where:

· E is the emotion space of the agent, which is defined based on the emotion wheel theory developed by Plutchik [52]. The wheel of emotion (as shown in Fig. 8) includes eight basic emotions that are opposed in pairs and that are of multiple shades. As emotions range from negative emotions to positive ones on the diagonal, we divide the emotion space into four basic dimensions: e_0 (surprise-interest), e_1 (sadness- happiness), e_2 (fear-anger), e_3 (disgust-trust), ranged in $[-1, 1]$.

Definition of the emotion state: $\forall t \in (0, T_{now}), i \in (0, 3)$; the emotion state at time $t$is given by (12):

$$\begin{aligned} \text{E}_t &= \Psi\left(\text{E}_{t-1}\right) + \text{P}\left\{\Phi_t\left(\sigma\right) + \Gamma\left(\omega\right)\right\} \\ &= \Psi\left(e_{0,t-1}, e_{1,t-1}, e_{2,t-1}, e_{3,t-1}\right) \\ &\quad + \text{P}\left\{\Phi_t\left(\sigma_0, \sigma_1, \sigma_2, \sigma_3\right)\right\} + \Gamma\left(\omega_0, \omega_1, \omega_2, \omega_3\right). \end{aligned} \tag{12}$$

In this equation, $\Psi\left(\text{E}_{t-1}\right)$ is the remaining emotional strength after decay from the last update time and is given by (13), $P(x)$ is the function for personalization computed as (14), $\Phi_t\left(\sigma\right)$ is the part evoked by environmental events and is given by the emotion appraisal approach, and $\Gamma\left(\omega\right)$ is the influence of emotion contagion effects given by the crowd Emotion Network computed as (10).

Definition of the emotion decay: As the attenuation curve of emotion is approximate to the curve of the exponential function, emotion decay is calculated using (13):

$$\Psi\left(e_{i,t}\right) = e_{i,t-\Delta t}\exp\{-\mu_i \Delta t\}. \tag{13}$$

· P is the personality vector based on the OCEAN model [53]. The five main factors considered are openness, conscientiousness, extraversion, agreeableness and neuroticism, as denoted by vector:

$$\text{P} = (p_0, p_1, p_2, p_3, p_4)^T. \tag{14}$$

We use matrix A to personalize the emotional influence of others, and the emotional strength elicited from environmental events is written as:

$$\text{A} = (a_{st})_{5\times 4}, \tag{15}$$

in which a_{st} is the weighting coefficient of influence that personality s imposes on emotion t.

Definition of the emotion influence: Assume that an agent with personality P has received an original increment of emotion as ΔE. After personalization, the emotion influence is given by (16):

$$\begin{aligned} \Delta\text{E}' = \text{P}^{\text{T}} \cdot \text{A} \cdot \Delta\text{E} &= (p_0, p_1, p_2, p_3, p_4)^{\text{T}} \\ &\cdot \begin{pmatrix} a_{00} & \cdots & a_{03} \\ \vdots & \ddots & \vdots \\ a_{40} & \cdots & a_{43} \end{pmatrix} \cdot \begin{pmatrix} \Delta e_0 \\ \Delta e_1 \\ \Delta e_2 \\ \Delta e_3 \end{pmatrix}. \end{aligned} \tag{16}$$

Combined with the propagated emotion given by (10), the outside influence of emotion contagion effects on agent i can be represented as:

$$\sum E_i' = P_i \times A_{5\times 4} \times \left(\sum E_i\right), \quad \sum E_i = \sum\nolimits_{j=0}^{n} E_{ij} \tag{17}$$

Here n is the number of neighbors who impact agent i.

· C is the emotional command constructed on the theoretical foundations of emotion rules given by the EmotionTree that the node belongs to. We extract emotion rules from the psychological literature and observed data, combine these rules into paradigms, and construct emotion commands as:

$$\mathrm{C}_k = \{\mathrm{P}_{event}, \mathrm{P}_{object}, e_k, Action_k, \mathrm{T}_k\}, \tag{18}$$

where C_k is the command given by EmotionTree k, P_{event} represents the positioning of the event source, P_{object} is the positioning of the object that emotions affect, e_k is the corresponding emotional energy, $Action_k$ is the selected motion clip moving with e_k, and T_k is the period during which this emotion is active. The process of abstracting commands from emotion rules is given in Section IV C.

B. ALGORITHM

Based on definitions of the sub-modules given in the last section, algorithms of the Emotion Network are designed as follows. The process involves the initialization of the Emotion Network and updating the EmotionTrees.

1) INITIALIZE THE EMOTION NETWORK

When a set of agents are input with random social relations and initial emotions, we use an emotion clustering algorithm as follows:

i. Initialize eight basic emotions $\{\mathrm{ES}_0^+, \mathrm{ES}_0^-, \mathrm{ES}_1^+, \mathrm{ES}_1^-, \mathrm{ES}_2^+, \mathrm{ES}_2^-, \mathrm{ES}_3^+, \mathrm{ES}_3^-\}$ and cluster the agents into corresponding sets based on the maximum and minimum component emotions.

For agent i: $E_i = (e_0, e_1, e_2, e_3)$, $k = \{x | e_x = \max(|e_j|), j = 0, 1, 2, 3\}$, if $e_k >= 0$, $E_i \in \mathrm{ES}_k^+$, else $E_i \in \mathrm{ES}_k^-$.

ii. Find the clustering center: the original center of cluster ES_k is denoted as the emotion vector of an agent with the maximum emotion component $|e_k|$.

$$Center_k = \{E_c | e_c = \max(e_k), E_k \in \mathrm{ES_k}\}. \tag{19}$$

iii. Calculate the average intra-cluster distance of the emotion set for the main component of the emotion vector. Sort the elements in set ES_k based on the main emotion component e_k as $\{E_0, E_1, E_2, \cdots, E_m\}$ with $e_{k,0} < e_{k,1} < \cdots < e_{k,m}$, and calculate the psychological distance between the two adjacent emotion nodes:

$$d_i = ArcWeight(i+1, i), \quad (i = 0, 1, \cdots, m-1). \tag{20}$$

iv. Calculate the average inter-cluster distance between emotion sets. Assuming that the number of emotion sets is n, the inter-cluster average distance can be computed as:

$$d_c = \frac{2}{(n(n-1))} \cdot \sum\nolimits_{i,j=0}^{n} ArcWeight(Center_i, Center_j). \tag{21}$$

v. Build EmotionTrees. Combine ES_k^+ and ES_k^- into ES_k with e_k ranging at $(-1, 1)$, use the *heap_insert* algorithm to generate an EmotionTree from sorted EmotionNodes in ES_k (if $ArcWeight\ (i,\ j)\ <\ d^*,\ j\ \in$ Tree), and then apply *push_heap* (i). The remaining nodes are pushed into a free node spool.

vi. Cluster terminates when $\sigma \cdot d^* < d_c$ where σ is a weighting coefficient.

After the above processes, the resulting EmotionTree satisfies the following: the absolute value of the main emotion component is maximized and the average inter-cluster distance is minimized.

2) UPDATE THE EMOTIONTREE

After individual emotion updating by (12), the relational distribution of crowds' changes and agents move across different subgroups. Therefore, the EmotionTree should be able to absorb new nodes, expel old ones and regulate intra-configurations while maintaining its boundary.

Absorb: when a free node enters the influence field of an EmotionTree, its corresponding emotional component exceeds the low threshold of this tree while the psychological distance (relation weight) is close enough and thus the free node will be absorbed by the EmotionTree. A tree may also absorb a node when its corresponding emotion component is stronger than the threshold value regardless of its positioning or psychological distance to other members. The pseudocode for Algorithm 1 is shown in Table 1.

TABLE 1. Algorithm for absorb.

Algorithm 1: *Absorb*

Input: d^*, R_{down}, e_{thr}, $e_{concentration}$, e_{up}, p_{root}
Output: *EmotionTree*
/* d^* is the average psychological distance; R_{down} is the low threshold of physical distance; e_{thr} is the low threshold of emotion value; $e_{concentraion}$ is the emotion concentration of the tree; e_{up} is the *upper limit of emotion value*; p_{root} is the position of the root node.*/
If ArcWeight (*root, node*) $<d^* \wedge$ Distance (*root, node*)-*tree.radius*$<R_{down}$ **then**
  If $e_{node} > e_{thr}$ **then**
    EmotionTree$_k$.absorb (*node*);
  End
End
EmotionTree.absorb (*node*)
foreach *node of freenodelist* **do**
  $cond1 \leftarrow$ InArea (*node*) $\vee$ $\|p_{node} - p_{root}\| <$ *tree.radius*;
  $cond2 \leftarrow e_{node} > e_{concentration}$;
  $cond3 \leftarrow e_{node} > e_{up}$;
  If $cond1 \wedge cond2 \vee cond3$ **then**
    add node to the candidate node list;
  End
End

Expel: each member node in an EmotionTree should be checked to ensure that its main emotion component exceeds the lower threshold. Otherwise it will be expelled from the tree. A member far from the boundary or positioned at a large psychological distance to other members that exceeds the average d^* value should also be expelled. The pseudocode for Algorithm 2 is shown in Table 2.

Configuration regulation: the intra-configuration of the EmotionTree is organized as a max heap. Its root has the maximum major emotion strength level, and leaf nodes have minimum levels, rendering the expel operation and computation of emotion propagation more convenient. Therefore, methods used to regulate the intra-configuration of an EmotionTree are similar to those applied to a heap.

TABLE 2. Algorithm for expel.

Algorithm 2: *Expel*

Input: d^*, R_{up}, e_{thr}, T, δ
Output: *EmotionTree*
Foreach $node_i$ *in memberlist* **do**
  $cond1 \leftarrow period_i < 0$;
  $cond2 \leftarrow node_i.e_k < e_{down}$;
  If $cond1 \vee cond2$ **then**
    reset the tree info of $node_i$;
    clear *tree.command* of $node_i$;
    add $node_i$ to *freenodelist;*
    delete $node_i$ from *memberlist*;
  End
End
MakeHeap (*memberlist*);
Fix the arclist for members;

TABLE 3. Algorithm for Melkman.

Algorithm 3: *Melkman*

Input: d, v_i
Output: d
/*Initialize:*/
If Left (v_0, v_1, v_2) **then**
  $d \leftarrow (v_2, v_0, v_1, v_2)$;
Else
  $d \leftarrow (v_2, v_1, v_0, v_2)$;
End
$i \leftarrow 3$;
While Left (d_{t-1}, d_t, v_i) ∧Left (d_b, d_{b+1}, v_i) **do**
  $i \leftarrow i+1$;
End
/*Restore convexity:*/
Repeat
  Pop d_t;
  Push v_i;
Until Left (d_{t-1}, d_t, v_i));
Repeat
  Remove d_b;
  Insert v_i;
Until Left (v_i, d_b, d_{b+1});
$i \leftarrow i+1$
Go to while;

Boundary fix: the geometrical boundary is denoted as a plane polygon, and the method used to fix an EmotionTree's boundary is the same as that used to compute the convex hull of the set of tree members' positions in two dimensions. We use the Melkman algorithm to compute the convex hull. In comparison to the rest of the convex hull algorithms, it is actually an on-line algorithm. It computes the hull as points are entered (note that all other algorithms involve finding an extreme point first). This means that the ordering of vertices does not need to be known either. The pseudocode for Algorithm 3 is shown in Table 3.

C. CROWD EMOTION PARADIGMS

Emotions and their expression are regulated by social norms, values, and expectations. These norms and values influence what the appropriate objects of emotion are defined as (i.e., what events should make a person angry, happy, jealous, and so on), and they also influence how emotions are to be expressed. We introduce the concept of the Emotion Paradigm (EP) to render these norms and potential processes of emotion more recognizable and convenient to model.

An EP has five necessary elements: a source (event that evoked the corresponding emotion), an object (to whom or what the emotion directed), a strength level (quantified energy level of an emotion), a performance level (how to display or express an emotion), and a period (active period of emotional expression), denoted as:

$$EP = \{Source, Object, Strength, Performance, Period\}. \tag{22}$$

We quantify *EP* into a computable emotion command C_k as noted in Section IV A:

$$C_k = \prod (EP_k) = \{\mathrm{P}_{event}, \mathrm{P}_{object}, e_k, Action_k, \mathrm{T}_k\}. \tag{23}$$

Source and *Object* are determined based on their positions P_{event} and P_{object}. *Strength* is normalized to e_k, with a range of $(-1, 1)$ for measurement purposes. As we aim to simulate the behavior of a crowd, emotions are expressed as various actions and are stored in a clip database. The method used to select an action clip based on corresponding levels of emotional strength is described in detail in Section IV D. *Period* is denoted as the active time T_k of emotion k.

We divide the subgroup constraints into emotional paradigms. EmotionNodes are divided into three phases based on stages of emotional performance: free nodes, candidates and members. Free nodes are agents of eventless emotions without any major emotional component or that are positioned outside of the influence of an emotional group. Candidates are impacted by a strange emotional component and expect to join congeners, but such emotions can be expressed. When candidates enter a group, inspired by others their main emotions become more pronounced, resulting in the creation of member nodes. Transitions of the three emotion states are shown in Fig. 3. The tree provides different commands to candidates and members so that they can constrain their behavior.

FIGURE 3. Transitions of the three emotion states.

When simulating, agents execute emotion commands by calculating motion variables from these commands (e.g., emotion source-oriented orientation and direction, emotion strength-based motion clip selection, action-constrained velocity, etc.).

We construct emotion paradigms and commands for the basic emotions: interest and surprise, happiness and sadness, anger and fear, and trust and disgust.

1) INTEREST AND SURPRISE

Emotions of interest and surprise can evoke various behaviors (e.g., attentive watching, gazing, etc.). The focus is placed on the event or object that causes the emotion. When emotions spread to crowds, because crowds enjoy social comparison and anonymous responsibility, collective phenomena (e.g., crowding) occur with long-term effects. As people tend to imitate others, individuals always adopt the behaviors of congeners. Therefore, a common event that cannot attract considerable interest may attract curious onlookers due to a group effect. We use the Interest & Surprise EmotionTree to depict bystander behaviors and define its properties as follows:

- Based on novel information from the environment, set the agent with the most interest (surprise) as the root, and build an interest or surprise tree.

- Absorb nodes with interest (surprise) levels greater than the emotion threshold; add them to the candidate list; give a command for candidates to gather.

- Once the candidates gather and enter the tree area, remove them from the candidate list and add them to the member list.

- Command the members to engage in corresponding actions through the tree action database.

- Measure the time period and expel the node that is no longer active.

Emotion paradigms and commands of the interest and surprise tree and function Ω_0 are denoted as is shown in Table 4. And the performance of the interest and surprise tree is shown in Fig. 4.

TABLE 4. Emotion paradigm and command for interest and surprise.

Emotion paradigm and command: Interest and surprise

I&S Emotion Paradigm:

$EP_{i\&s}$= {*StrangeObject*, *StrangeObject*, *WonderDegree*, *WonderActions*, *WonderPeriod*}

I&S Commands for candidates and members:

$$C^{i\&s}_{candidates} = \{\vec{p}_o, \vec{p}_o, e_0, walk(e_0), t_{max}\}$$

$$C^{i\&s}_{members} = \{\vec{p}_o, \vec{p}_o, e_0, watch(e_0), t_{e_0}\}$$

Motion variables:

$$\{\vec{o}_{e_0}, e_0, clip(e_0), t_{e_0}\} = \Omega_0(C^{i\&s})$$

$$\vec{o}=\Omega_0^o(\vec{p}_o, \vec{p}_{me}) = \frac{\vec{p}_o-\vec{p}_{me}}{\|\vec{p}_o-\vec{p}_{me}\|}$$

$$e_0 = \Omega_0^e(e_{base}, rand(), \vec{p}_o, \vec{p}_{me}) = e_{base} + personalize(rand()) + \lambda_0 \exp\{-\lambda_1 \cdot (\vec{p}_o - \vec{p}_{me)}\}$$

$$clip_{candidate} = \Omega_0^c(\widetilde{walk}, e_0) = matching(\widetilde{walk}, e_0)$$

$$clip_{member} = \Omega_0^c(\widetilde{watch}, e_0) = matching(\widetilde{watch}, e_0)$$

$$\vec{v}_{candidate} = \Omega_0^v(\vec{o}, clip_{candidate}) = \vec{o} \cdot velocity(clip_{candidate})$$

$$\vec{v}_{member} = \Omega_0^v(\vec{o}, clip_{member}) = \vec{o} \cdot velocity(clip_{member})$$

$$t_{member} = \Omega_0^t(e_0) = tree_0(e_0) \cdot period$$

2) HAPPINESS AND SADNESS

Happy and sad emotions are identified based on whether individual expectations are consistent with the situation. When the two extreme states are found in a crowd synchronously,

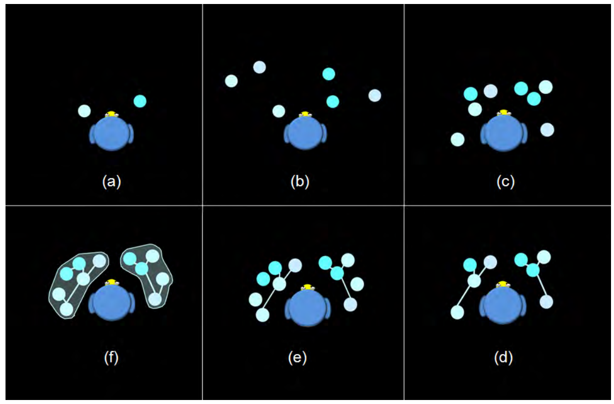

FIGURE 4. Performance of the Interest and Surprise Tree. The strange entity attracts pedestrians (presented as nodes) and piques their interest: (a) nodes of higher interest move closer; (b), (c) several nodes have been attracted; (d) two interest trees have formed; (e) the candidates turn into members; (f) an emotion field has formed with concentration.

they stimulate one another. This feature is particularly pronounced among fans. While watching a football game, football fans can exhibit extreme emotions ranging from sheer exhilaration to pure sadness. The bond forged between fans of a team provides them with emotional camaraderie while leading them to share a bitter hatred of the enemy. We build a happiness and sadness EmotionTree to imitate fan behaviors, and its properties are denoted as follows:

- Divide nodes into two groups based on initial settings; build the happiness tree and then the sadness tree.

- Compute the tree boundaries and emotion concentrations. When a node occupies its enemy tree's area, the intensity of its major emotion will increase.

- Choose the action through the clip database according to the emotional energy.

Commands of the happiness and sadness tree and of the corresponding function Ω_1 are denoted as shown in Table 5. And the performance of the happiness and sadness tree is shown in Fig. 5.

3) ANGER AND FEAR

Anger is a situated and embodied emotional experience that inhibits or improves ongoing respectful relationships. James Averill has identified the following rules for anger:

- A person has the right (duty) to become angry at any intentional wrongdoing or at unintentional misdeeds when such misdeeds are correctable.

- Anger should only be directed at persons and by extension at other entities that can be held responsible for certain actions.

- Anger should not be displaced on an innocent third party, and it should also not be directed at the target for reasons other than the instigation.

- The purpose of anger should be to correct a situation, to restore equity, and/or to prevent recurrence, and it should not be used to inflict injury or pain on the target or to achieve selfish ends through intimidation.

TABLE 5. Emotion paradigm and command for happiness and sadness.

Emotion paradigm and command: Happiness and Sadness

H&S Emotion Paradigm:

$EP_{i\&s}=$ {*Focus*, *sportscast*, *Null*, *PassionDegree*, *AudienceActions*, *GamePeriod*}

H&S Commands for candidates and members:

$$C^{h\&s}_{candidates} = \{\vec{p}_f, \vec{p}_f, e_1, standup(e_1), t_{max}\}$$

$$C^{h\&s}_{members} = \{\vec{p}_f, \vec{p}_f, e_1, clap(e_1), t_{e_1}\}$$

Motion variables:

$$\{\vec{o}_{e_1}, e_1, clip(e_1), t_{e_1}\} = \Omega_1(C^{h\&s})$$

$$\vec{o}=\Omega^o_1(\vec{p}_o, \vec{p}_{me}) = \frac{\vec{p}_o-\vec{p}_{me}}{\|\vec{p}_o-\vec{p}_{me}\|}$$

$$e_1 = \Omega^e_1(e_{base}, rand()) = e_{base} + personalize(rand())$$

$$clip_{candidate} = \Omega^c_1(\widetilde{sit2stand}, e_1) = matching(\widetilde{sit2stand}, e_1)$$

$$clip_{member} = \Omega^c_1(\widetilde{clapping}, e_1) = matching(\widetilde{clapping}, e_1)$$

$$\vec{v}_{candidate} = \Omega^v_1(\vec{o}, clip_{candidate}) = \vec{o}\cdot\vec{0})$$

$$\vec{v}_{member} = \Omega^v_1(\vec{o}, clip_{member}) = \vec{o}\cdot\vec{0}$$

$$t_{member} = \Omega^t_1(e_1) = tree_1(e_1)\cdot period$$

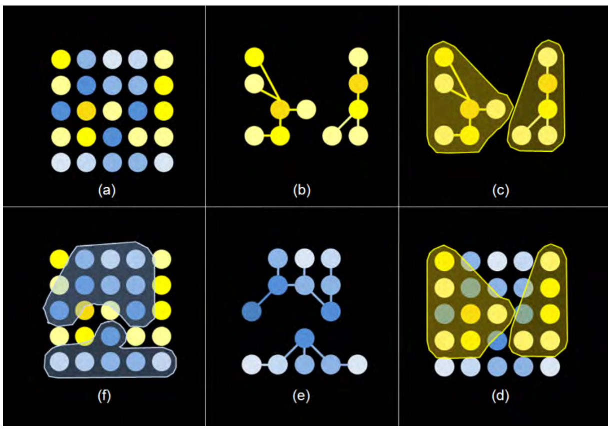

FIGURE 5. Performance of the Happiness and Sadness Tree. (a) Divide the crowd into two groups; (b), (e) build happiness and sadness trees, respectively; (c) compute tree boundaries and emotion concentrations; (d), (f) determine the influence of a tree's concentration on inner opposed nodes.

TABLE 6. Emotion paradigm and command for anger and fear.

Emotion paradigm and command: Anger and Fear

A&F Emotion Paradigm:

$EP_{a\&f}=$ {*Source*, *Fuse*, *Strength*, *FightAction*, *AngerPeriod*}

A&F Commands for candidates and members:

$$C^{a\&f}_{candidates} = \{\vec{p}_s, \vec{p}_f, e_2, rush(e_2), t_{max}\}$$

$$C^{a\&f}_{members} = \{\vec{p}_s, \vec{p}_f, e_2, fight(e_2), t_{e_2}\}$$

Motion variables:

$$\{\vec{o}_{e_2}, e_2, clip(e_2), t_{e_2}\} = \Omega_2(C^{a\&f})$$

$$\vec{o}=\Omega^o_2(\vec{p}_s, \vec{p}_{me}) = \frac{\vec{p}_s-\vec{p}_{me}}{\|\vec{p}_s-\vec{p}_{me}\|}$$

$$e_2 = \Omega^e_2(e_{base}, rand(), \vec{p}_s, \vec{p}_{me})$$
$$= e_{base} + personalize(rand()) + \lambda_0 \exp\{-\lambda_1\cdot(\vec{p}_s - \vec{p}_{me)}\}$$

$$clip_{candidate} = \Omega^c_2(\widetilde{escape2rush}, e_2) = matching(\widetilde{escape2rush}, e_2)$$

$$clip_{member} = \Omega^c_2(\widetilde{shake2fight}, e_2) = matching(\widetilde{shake2fight}, e_2)$$

$$\vec{v}_{candidate} = \Omega^v_2(\vec{o}, clip_{candidate}) = \vec{o}\cdot velocity(clip_{candidate})$$

$$\vec{v} = \Omega^v_2(\vec{o}, clip) = \vec{o}\cdot velocity(clip)$$

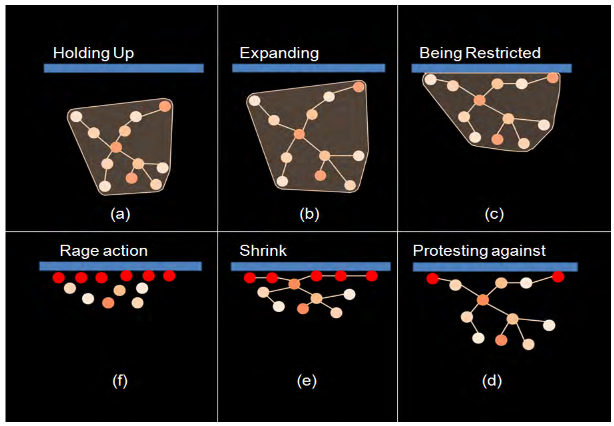

FIGURE 6. Performance of the Anger Tree. Angry nodes form an anger tree while extending. When nodes come into contact with the obstacle and provoke rage, the anger tree is initiated and the instigator is fought. (a), (b) Candidate nodes in the anger tree extend; (c) some nodes come into contact with the obstacle; (d) they turn into tree members and express their anger. (e), (f) The whole anger tree fights against the instigator.

4) TRUST AND DISGUST

Trust is attributable to relationships maintained within and between social groups. Trust is defined as follows:

- *A trustee's willingness to be obedient to the actions of another trustee.*

- *Confidence that the trusted individual will behave in a way that benefits the trustee.*

- *An absence of enforcement or control over actions performed by the trustee.*

When trusted individuals focus on a certain trustee, leadership forms. Leadership based on trust, cohesion, and team constraints in a human group differs from phenomena that align animals, which tend to form long-term relationships with others. Elements of trust can account for many of the structures we see in organized groups. We use the trust tree to describe behaviors of guided marching. Commands of the tree and of corresponding function Ω_3 are denoted as shown in Table 7. And the performance of the tree is shown in Fig. 7.

D. MOTION MATCHING SYSTEM

The energy level of an emotion defines the intensity of a performance. For facial animation, it manifests as different expressions; for cognation systems, it manifests as various decisions made; in neurological research, it is measured from the intensity of brain waves. When studying crowd behavior, we use vivid motion clips to reflect degrees of emotion.

According to the emotion cone, the vertical dimension represents the intensity level, and the circle denotes levels of similarity between emotions. We divide the clips into groups of different types. Each group is used as a sub-database for an emotional component. Clips of a group are distinguished by frequency, scope, elongation levels, etc. We then match these measurements with levels of emotion strength for motion selection as is shown in Fig. 8.

TABLE 7. Emotion paradigm and command for trust and disgust.

Emotion paradigm and command 4: Trust and Disgust

T&D Emotion Paradigm:

$EP_{t\&d}=\{Leader, QueueTail, Strength, Marching, TrustPeriod\}$

T&D Commands for candidates and members:

$C^{t\&d}_{candidates}=\{\vec{p}_{qt}, \vec{p}_{qt}, e_3, walk(e_3), t_{max}\}$

$C^{t\&d}_{members}=\{\vec{p}_{l}, \vec{p}_{l}, e_3, marching(e_3), t_{e_3}\}$

Motion variables:

$\{\vec{o}_{e_3}, e_3, clip(e_3), t_{e_3}\}=\Omega_3(C^{t\&d})$

$\vec{o}=\Omega_3^o(\vec{p}_o,\vec{p}_{me})=\frac{\vec{p}_o-\vec{p}_{me}}{\|\vec{p}_o-\vec{p}_{me}\|}$

$e_3=\Omega_3^e(e_{base}, rand(), \vec{p}_l, \vec{p}_{me})$
$\quad = e_{base}+personalize(rand())+\lambda_0 exp\{-\lambda_1\cdot(\vec{p}_l-\vec{p}_{me})\}$

$clip_{candidate}=\Omega_0^c(\widetilde{run2walk}, e_3)=matching(\widetilde{run2walk}, e_3)$

$clip_{member}=\Omega_3^c(\widetilde{marching}, e_3)=matching(\widetilde{marching}, e_3)$

$\vec{v}_{candidate}=\Omega_3^v(\vec{o}, clip_{candidate})=\vec{o}\cdot velocity(clip_{candidate})$

$\vec{v}=\Omega_3^v(\vec{o}, clip)=\vec{o}\cdot velocity(clip)$

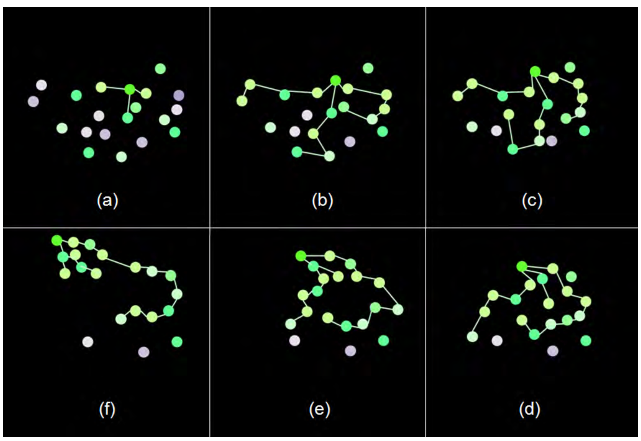

FIGURE 7. Performance of the Trust Tree. The leader (trustee) is first set as the root of the trust tree. It then absorbs nearby nodes close with more trust than the threshold and places them in the candidate list, commanding them to gather. When candidates are close enough to the end of the queue, they are turned into members. A new point that influences and attracts outside nodes is always set as the end of the marching queue.

To prevent the inconsistent selection of coterminous clips defined by levels of emotion, we must delay the emotion updating period until the current motion clip has been executed. An algorithm of motion synthesis is used to compose continuous motions and it involves finding the key frame (Algorithm 4 is shown in Table 8) and keyframe-based motion synthesis (Algorithm 5 is shown in Table 9).

For different emotions (major emotion component changes), agents must move between EmotionTrees. Thus, for the sub-database of motion clip changes, a simple motion graph is created for transitions as is shown in Fig. 9.

V. IMPLEMENTATION AND RESULTS

A. SIMULATOR

This transplantable Emotion Network is relatively independent from low level physical models. Our model is based on the simulation of a force model for collision avoidance and a potential field for navigation. We use simple interaction methods and a task decision system as a cognition model. The edited skin color corresponds to that of the wheel of emotion to perform psychological states.

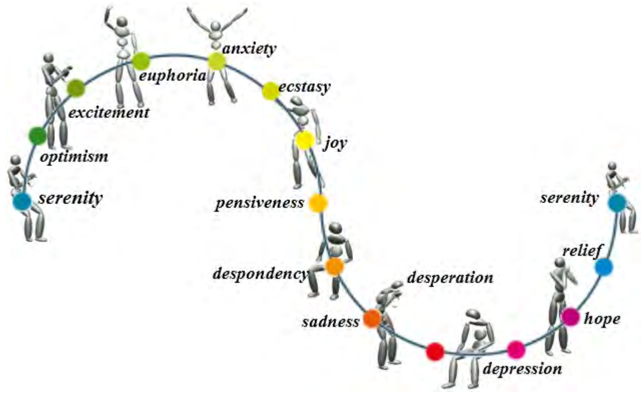

FIGURE 8. Matching motion clips with emotional strength.

TABLE 8. Algorithm for finding key frames.

Algorithm 4: *Find Key Frame*
Input: f, $last$, T, δ
Output: *KeyArray*
/* *KeyArray* is the array that stores key frames, f is the frame list, T is the number of frames in the list, and δ is a threshold set to move out of the loop. */
Add $f[0]$ into *KeyArray;*
$last \leftarrow 1$;
$t \leftarrow 2$;
While $t \leq T$ **do**
  $dist \leftarrow \|f(t)\text{-}f(last)\| \leftarrow \sum_{i=1}^{n} d(\hat{q}(t), \hat{q}(last))$
  /* Here α_i is weighting coefficient, $i = 1, 2..., n$; $\hat{q}(t)$ is the quaternion of the frame $f(t)$; and $d(q_1, q_2)$ is the difference between q_1 and q_2. */
  If $dist > \delta$ **then**
    add $f(t)$ to *KeyArray;*
    $last \leftarrow t$;
  End
  $t \leftarrow t + 1$;
End

TABLE 9. Algorithm for motion synthesis.

Algorithm 5: Motion Synthesis
Input: *KeyArray*, *num*
Output: $\vec{p}(t)$, $\hat{q}_i(t)$
/* *KeyArray* is the array storing key frames given by Algorithm1, and *num* is the number of key frames in it. */
$k = 0$;
While $k < num\text{-}1$ **do**
  $t_1 \leftarrow KeyArray[k]$;
  $t_2 \leftarrow KeyArray[k+1]$;
  Forall $t_1 < t < t_2$ **do**
    $\mu = (t - t_1)/(t_2 - t_1)$;
    compute root translation $\vec{p}(t) \leftarrow (1\text{-}\mu)\,\vec{p}(t_1)+\mu\vec{p}(t_2)$;
    Foreach *joint i* **do**
      compute rotation
      $\hat{q}_i(t) \leftarrow Slerp(\hat{q}_i(t_1), \hat{q}_i(t_2); \mu)$;
    End
  End
  $k \leftarrow k+1$;
End

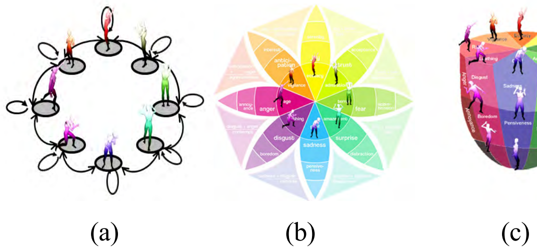

FIGURE 9. A motion graph for transitions between emotions.

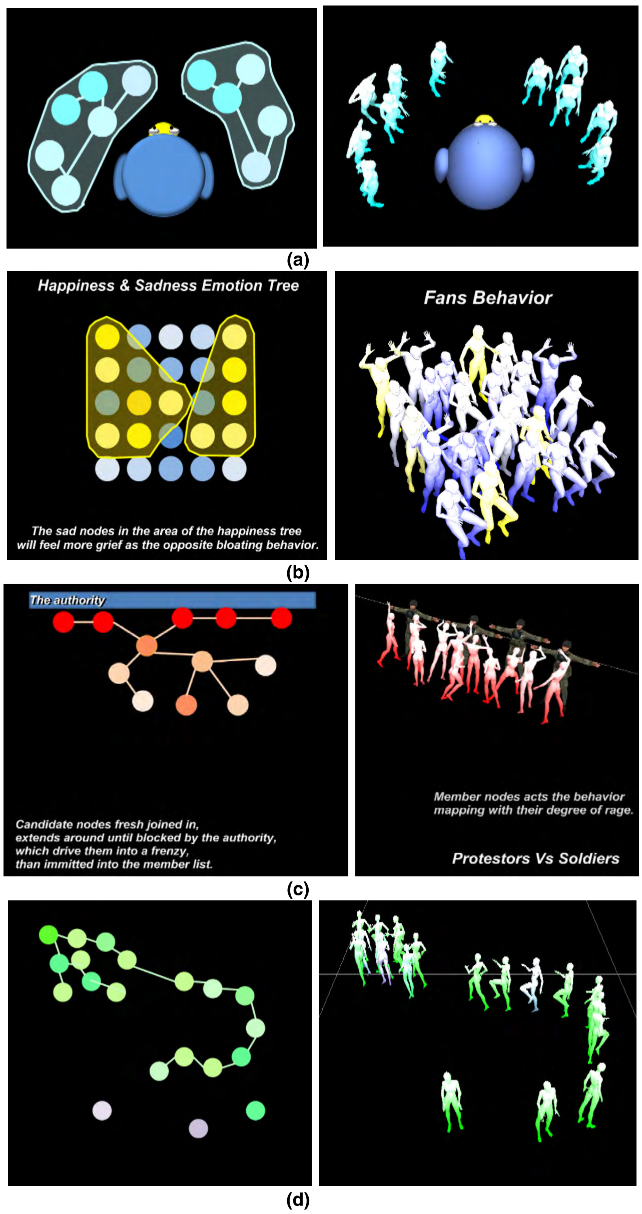

FIGURE 10. Emotional crowds generated by EmotionTrees. Emotional colors correspond to those shown in the wheel of emotion.
(a) Bystanders generated by an interest tree. (b) Fan behavior generated by a happiness and sadness tree. (c) Violent protestors generated by an anger tree. (d) Guided marching behavior generated by a trust tree.

There are some of the benefits of using our Emotion Network to study crowd behaviors (as is shown in Fig. 10). Firstly, our model can produce dynamic phenomena with emotional features while the individual model produces random behavioral changes and says nothing of situations not involving emotions; secondly, our method can be used to simulate autonomous crowd behaviors with various emotional features. Additionally, our method can be integrated into a physical crowd simulator as a high-level emotion manager providing intelligent support to a low-level driver.

We illustrate its performance through various scenarios:

B. BYSTANDERS

This scenario depicts the formation of haphazard crowds of bystanders (Fig. 11). There are two houses in the scene where pedestrians are produced. Pedestrians leave one house to enter another house. A flag (Fig. 11) in the scene represents a haphazard event. Several pedestrians are attracted first, and then more become interested and surround the scene. We also apply a mouse interaction event that disrupts the simulation by placing a cartoon model in the scene. This results in the autonomous formation of a crowd of bystanders (Fig. 12). An emotion potential mesh shows the energy level of interest as the circle is formed (Fig. 16(a)).

FIGURE 11. Crowding Bystanders: In the left sub-figure, agents around the flag act differently (swaying arms, looking left and right, tiptoeing, shrugging, etc.) The right subfigure shows a snapshot of the semicircle formed by the bystanders.

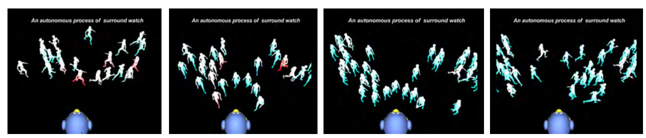

FIGURE 12. Crowd of bystanders forms autonomously. Pedestrians move until a cartoon model settles in the scene with the mouse. Some pedestrians change direction to see what has occurred, and then more and more stop. A crowd of bystanders forms autonomously. Minutes later, the crowd disperses.

C. AUDIENCE

In this scenario, we simulate the behaviors of an audience watching a football game on TV (Fig. 13). The audience has been divided into two groups by initial major emotion components representing the fans of the two football teams. Throughout the game, the emotions of the audience rise and fall and are expressed in the form how to generate Emotion-Trees of clapping, cheering, heckling, and crying. An emotion potential mesh is created to show the two opposing emotions when a goal is scored (Fig. 16(b)).

D. VIOLENT PROTESTORS

In the scenario shown in Fig. 14, we simulate a crowd protesting at a town hall. Police prevent the angry crowd from

FIGURE 13. Passionate Fans: An audience is watching a football game on television. When a goal is scored by one team, its fans stand up and cheer.

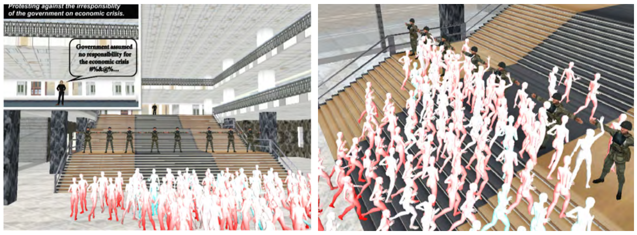

FIGURE 14. Violent Protestors: This scenario depicts a crowd of violent protestors at a town hall. Police hold a line in front of the crowd. The angry protestors shout, shake their fists and struggle against the defensive wall.

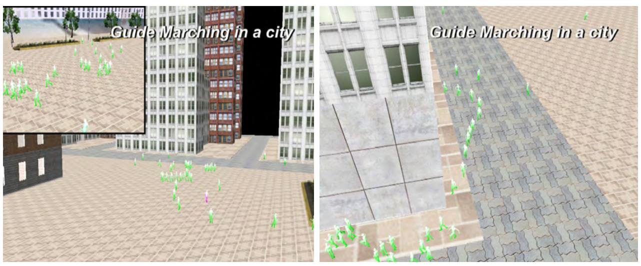

FIGURE 15. Guided Marching: A marching procession forms autonomously in the street. A guide leads the followers in a procession

rushing towards the politicians. Once the angry protestors are stopped, they violently attack the officers. Some of the protestors break through the defensive wall. An emotion potential mesh is figured to illustrate the expression of anger as the anger tree is extended (Fig. 16(c)).

E. GUIDED MARCHING

This scenario depicts a procession formed autonomously on a street (Fig. 15). In real life, parades are always led by a few initiators and are followed by accepting participators. We simulate this situation by designating an initiator of loitering pedestrians. Upon experiencing trust, surrounding followers march in a procession. An emotion potential mesh illustrates the expression of trust during the formation of the procession (Fig. 16(d)).

F. COMPARISON

To illustrate the advantages of the Emotion Network, we conducted a simulation involving bystanders under three different conditions: original simulation without the emotion model, simulation with emotion factors in agents and with corresponding maps on behavior and simulation with the Emotion Network (shown in Fig. 17). In the simulation

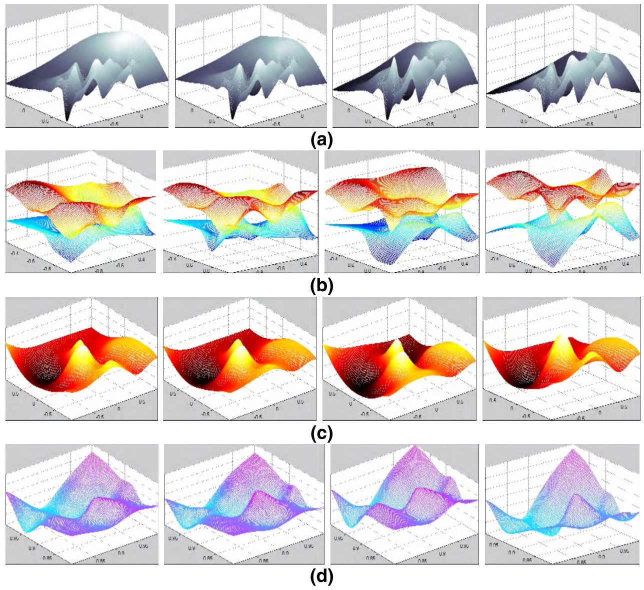

FIGURE 16. Emotion potential mesh. (a) Interest mesh. (b) Happiness and Sadness mesh. (c) Anger mesh. (d) Trust mesh.

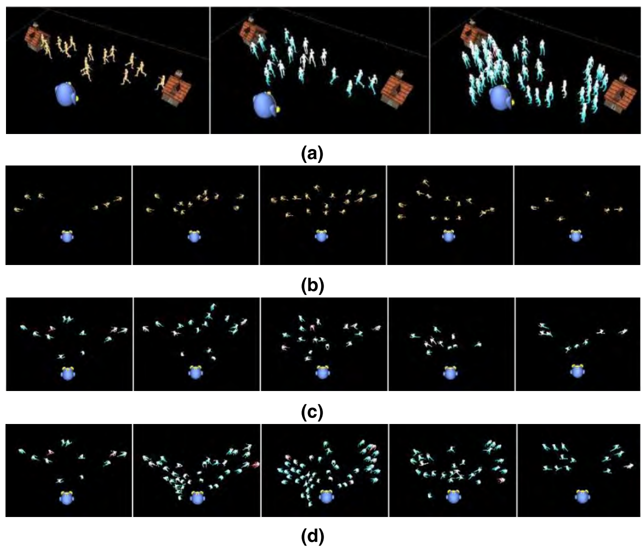

FIGURE 17. Comparative experiment on bystander behaviors. (a) Snapshot of three experiments conducted at the same time. (b) Without the emotion model. (c) Individual emotion factors with motion maps. (d) Emotion network.

without an emotion layer, pedestrians ignore the event and move toward their targets. Implanting emotional properties into agents and setting corresponding influence maps causes some pedestrians to stop but only impresses scattered onlookers. For the simulation involving the Emotion Network, a crowd of bystanders forms naturally, remains for a period of time and then gradually disperses.

VI. CONCLUSION

We introduce an Emotion Network to address the challenging problem of simulating socially emotional crowds. Inspired by theories of social networks and intergroup emotions, we build individual emotion spaces as network nodes, embody their psychological linkages with relationship arcs, and cluster emotionally homogeneous crowds into EmotionTrees. Natural behaviors with collective constraints and emotional characteristics are generated by extracting emotion rules and by constraining behaviors with tree commands.

By eliciting emotion rules and mapping emotion rules into paradigms, we construct tree commands based on emotion factors (e.g., the event source, object, strength level and period). From intergroup emotion theory, we apply subgroup constraints to emotion commands. Emotion nodes are divided into three phases: free nodes, candidates and members. Finally, motion variables can be calculated from emotion commands (emotion source-oriented orientation, velocity and emotion strength-based motion clip selection).

We successfully construct emotion paradigms for four basic emotions and we simulate phenomena related to bystanders, football fans, violent protestors and guided marches. Compared to simulations not involving emotions and compared to the individual emotion model, our model can produce dynamic phenomena with emotional features while the individual model produces random behavioral changes and says nothing of situations not involving emotions.

Our model is limited in its depiction of overlaps between subgroups. It is difficult to address multi-identity issues of emotional self-categorization even in real life. Furthermore, we would like to explore the behavioral features of crowds with complex mixed emotions as well as interactions between groups expressing different emotions.

REFERENCES

[1] D. Thalmann, "Populating virtual environments with crowds," in *Proc. ACM Int. Conf. Virtual Reality Continuum Appl.* Hong Kong: Chinese Univ. Hong Kong, Jun. 2006, p. 11.

[2] N. Pelechano, J. M. Allbeck, and N. I. Badler. *Virtual Crowds: Methods, Simulation, and Control*. San Rafael, CA, USA: Morgan & Claypool, 2008.

[3] F. Durupınar, "From audiences to mobs: Crowd simulation with psychological factors," Ph.D. dissertation, Dept. Comput. Eng., Bilkent Univ., Ankara, Turkey, 2010.

[4] F. Martinez-Gil, M. Lozano, I. García-Fernández, and F. Fernández, "Modeling, evaluation, and scale on artificial pedestrians: A literature review," *ACM Comput. Surv.*, vol. 50, no. 5, pp. 72–107, 2017.

[5] R. L. Hughes, "The flow of human crowds," *Annu. Rev. Fluid Mech.*, vol. 35, no. 1, pp. 169–182, 2003.

[6] A. Treuille, S. Cooper, and Z. Popović, "Continuum crowds," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1160–1168, 2006.

[7] R. Narain, A. Golas, S. Curtis, and M. C. Lin, "Aggregate dynamics for dense crowd simulation," *ACM Trans. Graph.*, vol. 28, no. 5, p. 122, 2009.

[8] S. Chenney, "Flow tiles," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, 2004, pp. 233–242.

[9] X. Jin, J. Xu, C. C. L. Wang, S. Huang, and J. Zhang, "Interactive control of large-crowd navigation in virtual environments using vector fields," *IEEE Comput. Graph. Appl.*, vol. 28, no. 6, pp. 37–46, Nov. 2008.

[10] S. Patil, J. van den Berg, S. Curtis, M. C. Lin, and D. Manocha, "Directing crowd simulations using navigation fields," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 2, pp. 244–254, Feb. 2011.

[11] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 25–34, Jul. 1987.

[12] C. W. Reynolds, "Steering behaviors for autonomous characters," in *Proc. Game Developers Conf.*, 1999, pp. 1–14.

[13] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 51, no. 5, p. 4282, 1995.

[14] D. Helbing, I. Farkas, and T. Vicsek "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487–490, 2000.

[15] S. R. Musse and D. Thalmann, "A model of human crowd behavior: Group inter-relationship and collision detection analysis," in *Computer Animation and Simulation*. Vienna, Austria: Springer-Verlag, 1997, pp. 39–51.

[16] J. Pettré, J.-P. Laumond, and D. Thalmann, "A navigation graph for real-time crowd animation on multilayered and uneven terrain," in *Proc. 1st Int. Workshop Crowd Simulation*, 2005, pp. 81–90.

[17] M. Sung, M. Gleicher, and S. Chenney, "Scalable behaviors for crowd simulation," *Comput. Graph. Forum*, vol. 23, no. 3, pp. 519–528, 2004.

[18] S. Wei and D. Terzopoulos, "Autonomous pedestrians," *Graph. Models*, vol. 69, nos. 5–6, pp. 246–274, 2007.

[19] S. Paris, J. Pettré, and S. Donikian, "Pedestrian reactive navigation for crowd simulation: A predictive approach," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 665–674, 2007.

[20] H. Yeh, S. Curtis, S. Patil, J. van den Berg, D. Manocha, and M. Lin, "Composite agents," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, 2008, pp. 39–47.

[21] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 655–664, 2007.

[22] T. Kwon, K. H. Lee, J. Lee, and S. Takahashi, "Group motion editing," *ACM Trans. Graph.*, vol. 27, no. 3, p. 80, 2008.

[23] E. Ju, M. G. Choi, M. Park, J. Lee, K. H. Lee, and S. Takahashi, "Morphable crowds," *ACM Trans. Graph.*, vol. 29, no. 6, p. 140, 2010.

[24] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. J. Robot. Res.*, vol. 21, no. 3, pp. 233–255, 2002.

[25] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[26] J. P. van den Berg and M. H. Overmars, "Roadmap-based motion planning in dynamic environments," *IEEE Trans. Robot.*, vol. 21, no. 5, pp. 885–897, Oct. 2005.

[27] J. P. van den Berg and M. H. Overmars, "Path planning in repetitive environments," in *Proc. Int. Conf. Methods Models Autom. Robot.*, 2006, pp. 657–662.

[28] S. Rodriguez, J.-M. Lien, and N. M. Amato, "Planning motion in completely deformable environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2006, pp. 2466–2471.

[29] M. Zucker, J. Kuffner, and M. Branicky, "Multipartite RRTs for rapid replanning in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, Apr. 2007, pp. 1603–1609.

[30] M. G. Choi, M. Kim, K. L. Hyun, and J. Lee, "Deformable motion: Squeezing into cluttered environments," *Comput. Graph. Forum*, vol. 30, no. 2, pp. 445–453, 2011.

[31] S. Levine, Y. Lee, V. Koltun, and Z. Popović, "Space-time planning with parameterized locomotion controllers," *ACM Trans. Graph.*, vol. 30, no. 3, pp. 1–11, May 2011.

[32] S. J. Guy, J. Chhugani, S. Curtis, P. Dubey, M. Lin, and D. Manocha, "PLEdestrians: A least-effort approach to crowd simulation," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, 2010, pp. 119–128.

[33] C. Mousas, P. Newbury, and C.-N. Anagnostopoulos, "The minimum energy expenditure shortest path method," *J. Graph. Tools*, vol. 17, nos. 1–2, pp. 31–44, 2013.

[34] J. Allbeck and N. Badler, "Toward representing agent behaviors modified by personality and emotion," in *Proc. Embodied Conversational Agents AAMAS*, 2002, pp. 1–6.

[35] N. Pelechano, J. M. Allbeck, and N. I. Badler. *Virtual Crowds: Methods, Simulation, and Control*. San Rafael, CA, USA: Morgan & Claypool, 2008.

[36] Q. Yu and D. Terzopoulos, "A decision network framework for the behavioral animation of virtual humans," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, 2007, pp. 119–128.

[37] F. Durupinar, U. Güdükbay, A. Aman, and N. I. Badler, "Psychological parameters for crowd simulation: From audiences to mobs," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 9, pp. 2145–2159, Sep. 2016.

[38] J. Funge, X. Tu, and D. Terzopoulos, "Cognitive modeling: Knowledge, reasoning and planning for intelligent characters," in *Proc. 26th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 1999, pp. 29–38.

[39] F. Garat and D. Thalmann, "Guiding and interacting with virtual crowds in real-time," in *Proc. Eurograph. Workshop Animation Simulation*, vol. 33, no. 11, 1999, pp. 23–34.

[40] E. de Sevin and D. Thalmann, "A motivational model of action selection for virtual humans," in *Proc. Int. Conf. Comput. Graph.*, 2005, pp. 213–220.

[41] D. Helbing, I. J. Farkas, P. Molnár, and T. Vicsek, "Simulation of pedestrian crowds in normal and evacuation situations," in *Pedestrian and Evacuation Dynamics*. Berlin, Germany: Springer-Verlag, 2002, pp. 21–58.
[42] F. Durupinar, J. Allbeck, N. Pelechano, and N. Badler, "Creating crowd variation with the OCEAN personality model," in *Proc. Int. Joint Conf. Auto. Agents Multiagent Syst. (DBLP)*, 2008, pp. 1217–1220.
[43] X. Pan, C. S. Han, and K. H. Law, "A multi-agent based simulation framework for the study of human and social behavior in egress analysis," in *Proc. Int. Conf. Comput. Civil Eng.*, 2005, pp. 12–15.
[44] D. Cañamero, "Designing emotions for activity selection," Dept. Comput. Sci., Univ. Aarhus, Aarhus, Denmark, Tech. Rep. DAIMI 545, 2000, vol. 29.
[45] L. Fu, W. Song, W. Lv, and S. Lo, "Simulation of emotional contagion using modified SIR model: A cellular automaton approach," *Phys. A, Stat. Mech. Appl.*, vol. 405, pp. 380–391, Jul. 2014.
[46] L. Fu, W. Song, W. Lv, X. Liu, and S. Lo, "Multi-grid simulation of counter flow pedestrian dynamics with emotion propagation," *Simul. Model. Pract. Theory*, vol. 60, pp. 1–14, Jan. 2016.
[47] M. Cao, G. Zhang, M. Wang, D. Lu, and H. Liu, "A method of emotion contagion for crowd evacuation," *Phys. A, Stat. Mech. Appl.*, vol. 483, pp. 250–258, Oct. 2017.
[48] M. Hoffman, *Empathy and Prosocial Behavior*, 3rd ed. New York, NY, USA: Guildford Press, 2008, pp. 440–455.
[49] D. M. Mackie, E. R. Smith, and D. G. Ray, "Intergroup emotions and intergroup relations," *Social Personality Psychol. Compass*, vol. 2, no. 5, pp. 1866–1880, 2008.
[50] S. Freud, *The Ego and the Id*. London, U.K.: Hogarth, 1949.
[51] S. G. Barsade, "The ripple effect: Emotional contagion and its influence on group behavior," *Administ. Sci. Quart.*, vol. 47, no. 4, pp. 644–675, Dec. 2002.
[52] R. Plutchik, "The nature of emotions," *Amer. Sci.*, vol. 89, no. 4, p. 344, Aug. 2001.
[53] J. M. Digman, "Personality structure: Emergence of the five-factor model," *Annu. Rev. Psychol.*, vol. 41, pp. 417–440, Feb. 1990.
[54] C. Loscos, D. Marchal, and A. Meyer, "Intuitive crowd behavior in dense urban environments using local laws," in *Proc. IEEE Theory Pract. Comput. Graph.*, Jun. 2003, pp. 122–129.
[55] I. Karamouzas, N. Sohre, R. Narain, and S. J. Guy, "Implicit crowds: Optimization integrator for robust crowd simulation," *ACM Trans. Graph.*, vol. 36, no. 4, p. 136, 2017.

LEI LYU received the Ph.D. degree in computer application technology from the University of Chinese Academy of Sciences in 2013. He is currently an Associate Professor of the School of Information Science and Engineering, Shandong Normal University, Jinan, China. His current research interests include virtual reality and multimedia.

JINLING ZHANG received the Ph.D. degree in computer application technology from the Beijing University of Posts and Telecommunications. She is currently an Associate Professor of the School of Information, Renmin University of China, Beijing, China. Her current research interests include virtual reality, and network and information security.