

# Integrating Information Theory in Agent-Based Crowd Simulation Behavior Models

CAGATAY TURKAY<sup>1,\*</sup> EMRE KOC<sup>2</sup> AND SELIM BALCISOY<sup>2</sup>

<sup>1</sup>University of Bergen, Bergen, Norway

<sup>2</sup>Sabanci University, Istanbul, Turkey

\*Corresponding author: cagatay.turkay@ii.uib.no

**Crowds must be simulated believable in terms of their appearance and behavior to improve a virtual environment's realism. Due to the complex nature of human behavior, realistic behavior of agents in crowd simulations is still a challenging problem. In this paper, we propose a novel behavioral model which builds analytical maps to control agents' behavior adaptively with agent-crowd interaction formulations. We introduce information theoretical concepts to construct analytical maps automatically. Our model can be integrated into crowd simulators and enhance their behavioral complexity. We made comparative analyses of the presented behavior model with measured crowd data and two agent-based crowd simulators.**

*Keywords:* crowd simulation; information theory

*Received 14 July 2010; revised 24 January 2011*

Handling editor: Erol Gelenbe

## 1. INTRODUCTION

A crowd constitutes a critical component in many virtual environment and entertainment applications. Believable behavior and appearance of a crowd improves a virtual environment's realism. Recent advances in graphics hardware address the issue of photo-realistic rendering of crowds. However, due to the complex nature of human behavior, realistic behavior of agents in crowd simulations is still a challenging problem. Previous approaches propose either (i) global solutions with high-level formulations [1]—which can simulate large numbers of agents however are not suitable for creating complexity in the crowd or (ii) low-level scripted, complex agent-based methods—which are computationally expensive and require expertise and effort in the production phase [2].

In this paper, we are proposing an analytical agent-based behavioral model that integrates global knowledge about crowd formation into local, agent-based behavior control. We use analytical representations of a crowd's activities, called *behavior maps*, which are constructed with a statistical framework based on information theory. Our model proposes an agent definition responsive to behavior map values and agent-crowd interaction formulations. Agents behaving in realistic, variable and complex manners can be achieved

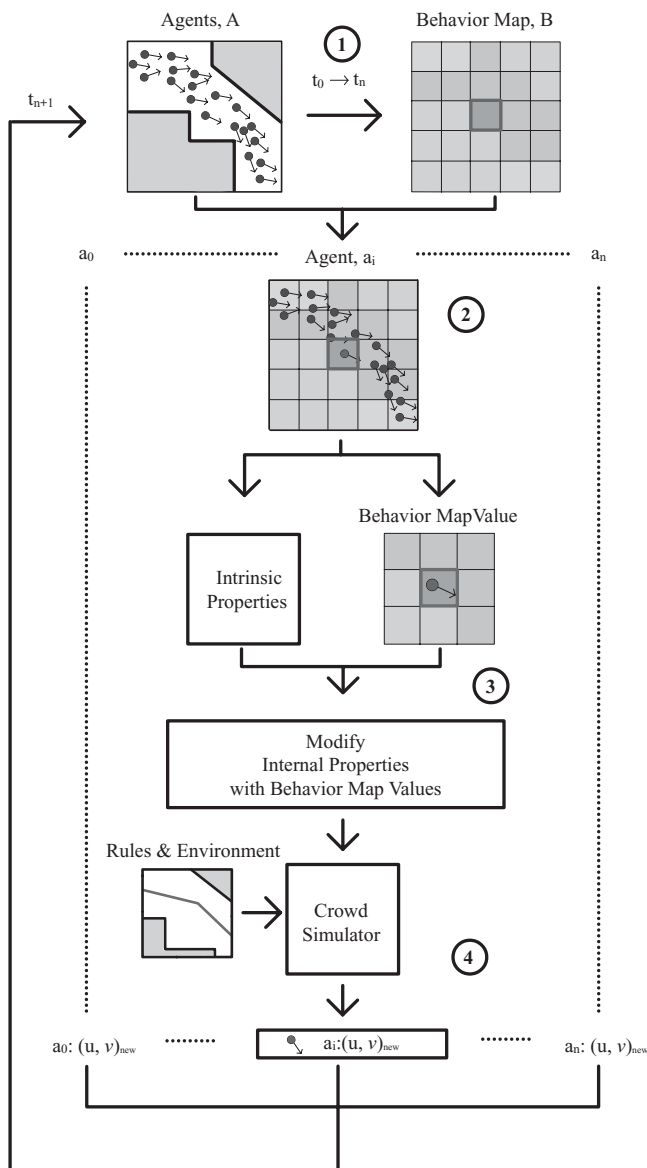
with our behavioral model without the need for low-level scripting.

Our proposed model is founded on analytical maps to represent activities of the crowd. We borrow ideas from behavioral mapping techniques used in psychology research to build an analytical model of the environment. These techniques involve place-centered maps, which keep track of behavior of individuals within a specific space and time. These maps display how and when a place is being populated [3]. In our model, we utilize the notion of place-centered maps in crowd simulations and propose *behavior maps*. Behavior maps are automatically generated statistical analysis maps that record and represent both spatial and temporal dynamics of agents. Methods to construct behavior maps are derived from scene analysis methods introduced in [4].

*Information theory* constitutes the theoretical foundation of behavior map construction. We use information theory quantities, i.e. *information entropy* and *Kullback–Leibler (KL) divergence*, to construct behavior maps. At each time step of a simulation, physical properties of agents, i.e. position and velocity, are listed and classified. The resulting behavior map conveys probabilistic and statistical information of agents' locomotion.

Agents access behavior maps and modify their intrinsic properties. How agents respond and behave according to their intrinsic properties and behavior maps are defined with agent–crowd interaction formulations.

Beneath all this high-level structure, we utilize a multi-agent navigation system to solve agent–agent and agent–environment interactions through collision detection and path planning algorithms. Our model (Fig. 1) can extend any existing agent-based crowd simulator. In this paper, we extend reciprocal velocity obstacles (RVO) multi-agent navigation system introduced in [5].



**FIGURE 1.** Overall structure of our model. (1) Behavior map is produced. (2) Agents are assigned a specific cell value. (3) Agents' intrinsic properties are modified with behavior map value. (4) Agents' locomotion is determined by simulator.

We believe that our methodology can aid the utilization of emergency response [6] and decision support systems [7, 8]. Owing to capabilities of our model to monitor and evaluate the simulation environment, it can be integrated into these systems to enhance their overall performance. In Section 4, we discuss the performance of our model in an evacuation scenario simulation.

The rest of the paper is organized as follows. In Section 2, we review the related literature. In Section 3.1, we introduce behavior maps. In Section 3.2, the general agent representation is given. In Section 3.3, agent–crowd interactions are formulated. Section 4 discusses the results obtained with the model. To conclude, final remarks are made in the conclusion section.

## 2. RELATED WORK

An overall idea of the challenges and improvements in crowd simulation can be obtained in [9] by Thalmann and Raupp Musse. There are several behavioral models proposed in the literature and a survey by Kasap and Magnenat-Thalmann [10] covers most of these studies. There have been many studies on agent-based crowd models to create human-like behaviors. Seminal works of Reynolds used behavioral models considering local rules [11] and created emergent flocking [12] behaviors. There is considerable work on agent-based crowd simulators incorporating psychological models and sociological factors. In [13], Luo *et al.* model social group and crowd-related behaviors. Their main focus is a layered framework to reflect the natural pattern of a human-like decision-making process. Gelenbe *et al.* [14] utilize an information-fusing strategy to determine the behavior of autonomous agents. They incorporate agents' pursuits and aims, the physical setting and their collective social behavior. In [15], Rymill and Dodgson tried to improve the quality of agent behavior by adding theories from psychology. In their work, they tried to produce more realistic collision avoidance responses. Musse and Thalmann [2] developed virtual human agents with intentions, beliefs, knowledge and perception to create a realistic crowd behavior. In [16], Pelechano *et al.* assigned psychological roles and communication skills to agents to produce diverse and realistic behaviors. Gelenbe *et al.* [17] use biological metaphors to solve the interaction of agents. In their work, they utilize G-networks where agents are represented as normal cells, bad cells and immune defense cells. In a later study, Gelenbe and Wang [18] utilize G-networks to construct a mathematical model to efficiently represent a system of agents and analyze their behavior.

In a recent work, [19] created an improved model by using psychological and geometrical rules with a social and physical forces model. There are studies that model the virtual environment as maps to guide agents' behaviors. In [20] Shao and Terzopoulos modeled the environment with topological,

perception and path maps to generate autonomous agents. Gayle *et al.* [21] used adaptive roadmaps, which evolve with the dynamic nature of the environment. In [22], Sung *et al.* assign situations and behaviors directly to environment rather than the agents themselves. Kaplan and Gelenbe [23] propose a technique where they fuse terrain information and agent goals. This technique increases the performance of simulations with many complex obstacles. Dimakis *et al.* [24] propose a system called DBES, which incorporates a distributed simulation model to reduce the simulation time. In their system, they define two different type of agents, namely simulators and simulated agents. Simulator agents are active agents and perform certain strategies, and simulated agents follow these strategies. They also utilize a graph-based simulator to put focus on certain areas of the simulation.

The concept of behavior maps has been used in robotics and vision field. Dornhege and Kleiner [25] defined behavior maps as encoding context information of the environment and used these maps to autonomously navigate a robot on rough terrain. Berclaz *et al.* [26] used behavior maps to encode probabilities of moving in a certain direction on a specified location and also to track trajectories of people and detect anomalies in people's behaviors. In their study, they used expectation maximization algorithms to detect anomalies. Autonomous agents have also proven to be beneficial in augmented reality applications in a study by Gelenbe *et al.* [27]. In their work, the simulation environment is displayed in a real-world setting and the agents react to this environment realistically.

We integrated theories from behavioral modeling and borrowed ideas from studies representing the environment with guidance maps. To compute these maps, we employed quantities from information theory. Information theory has been introduced into computer graphics field by Feixas *et al.* [28] which proposes a framework to measure scene complexity by using information theory quantities. In a recent study, Turkay *et al.* [4] used information theory-based formulations to automatically control the virtual camera in a crowded environment. They extended their information theory-based model to control how agents behave in a crowd simulation [29, 30]. We improve and elaborate the behavioral model proposed in [30] and develop concrete formulations and methods to use this behavioral model to extend any crowd simulator's variability and realism.

### 3. ANALYTICAL BEHAVIORAL MODEL

Our model provides global knowledge on crowd's activities and enables the crowd simulator to incorporate agent-crowd interactions to modify agents' behavior. Behavior maps constitute the foundation of our model. They record and analytically represent crowd's activities. The second element of our model is a generic agent representation to access behavior maps. The final element in our model is a set of formulations

to link the underlying crowd simulator with behavior maps. We customize the agent representation to fit into the current crowd simulator's features before developing these formulations.

#### 3.1. Crowd representation with behavior maps

Behavior maps are analytical representations of a crowd which span over the whole virtual environment and monitor agents' locomotion during the simulation. We formulate agent-crowd interactions through using behavior maps in our calculations. A behavior map  $B$  is a 2D grid, consisting of  $w$  rows and  $h$  columns, where each cell is a square with side of length  $l$ . In this work, we assume that the environment consists of only horizontal paths. Figure 2 illustrates how different types of behavior maps are constructed.

Let  $A = \{a_1, a_2, \dots, a_n\}$  be the set of agents present in a simulation, where  $a_i$  represents a single agent. Physical properties of an agent can be described as  $a_i = \{u, \vec{v} : u, \vec{v} \in \mathbb{R}^2\}$ , where  $u$  defines the position and  $\vec{v}$  defines the velocity of agent  $a_i$ . The activity of an agent is described by: (i) its position and (ii) the direction and magnitude of its velocity vector. The activity of a single agent is added into our model in two steps. First, the agent's position is registered to the corresponding cell in  $B$  by mapping its position to the  $w \times h$  grid and, secondly, its velocity vector is quantized to be added as a sample in probabilistic distributions.

Figure 2 illustrates behavior map construction. A behavior map is constructed as a convex combination of two information theory-based maps called *entropy map* and *expectance map*. In [4], these maps have proven to represent the temporal and spatial dynamics of a crowd's locomotion.

##### 3.1.1. Behavior map construction

Information theory deals with the quantification of information. *Information entropy* is the key measure in information theory [31] and it is the first quantity we use in our model to construct behavior maps. Let  $X$  be a discrete random variable which takes values from set  $\chi$  with probability distribution  $p(x) = Pr[X = x]$ ,  $x \in \chi$ . Entropy  $H(x)$  of random variable  $X$  is defined by

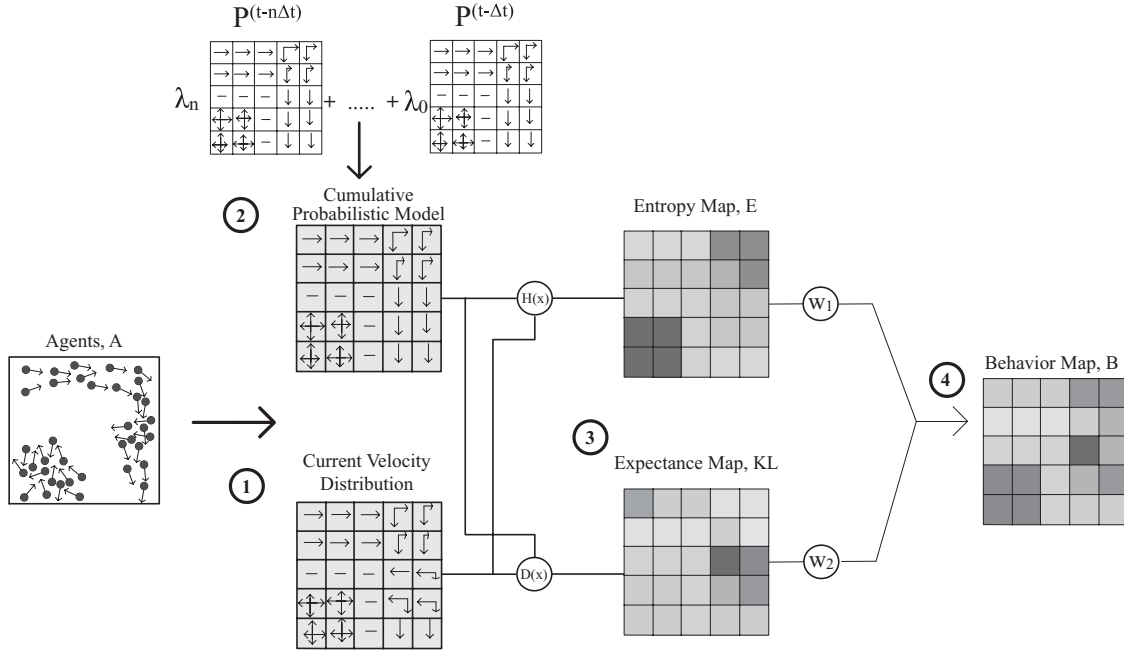
$$H(x) = - \sum_{x \in \chi} p(x) \log p(x). \quad (1)$$

Entropy provides insight about how likely a system produces varied outcomes; namely, it is a measure of uncertainty of a random variable.

The other quantity we have utilized is *KL divergence* [32]. Take two probability mass functions (pmf)  $p(x)$  and  $q(x)$ ; then the divergence between pmf's  $p(x)$  and  $q(x)$  is given by

$$D(p||q) = - \sum_{x \in \chi} p(x) \log \frac{p(x)}{q(x)}, \quad (2)$$

which is a non-symmetric metric expressing the difference between two probability distributions. Specifically, it represents



**FIGURE 2.** Behavior map construction. (1) Current distribution of crowd locomotion. (2) Older distributions are merged with a temporal filter. (3) *Entropy* and *expectance maps* are calculated. (4) Behavior map is constructed with convex combination using user-defined weights.

how probable the distribution defined by  $q(x)$  is likely to occur when  $p(x)$  is given.

To use these information theory quantities, we use a pmf denoted as  $P_{\vec{v}}$ , to characterize the distribution of a crowd's locomotion. The velocity of each agent is added as a sample to the 2D pmf. As  $P_{\vec{v}}$  is taking samples over a period of time, we use a temporally updated and filtered pmf as defined in [4]. This pmf is denoted as  $P_{\vec{v}}^{t_1 \rightarrow t_2}$ , where  $t_1$  and  $t_2$  are two distinct time steps throughout the simulation. These information theory quantities, probability distribution functions and the temporal filter mechanism are utilized to construct the components of a behavior map, which are the *entropy* and *expectance maps*.

**Entropy map.** Entropy measures the uncertainty of a random variable. If locomotion of an agent is considered as the random variable, entropy values represent the magnitude of predictability of a crowd's movements. Entropy values denote whether agents move independently or in a group. Locations with smaller entropy values denote where agents move with similar velocities. Conversely, locations with higher entropy values represent disorder in agents' locomotion. To build an entropy map  $E$  we begin by considering a random variable  $X_{i,j}$  ( $i, j$  indicating location on  $E$ ), drawn according to pmf  $(P_{\vec{v}}^{(t-n\Delta t) \rightarrow t})_{i,j}$ . Then,  $E$  can be defined as

$$E^t = \{H(X_{i,j}) : 0 \leq i < w, 0 \leq j < h\}, \quad (3)$$

where  $H(X_{i,j})$  is the entropy of  $X_{i,j}$  as defined in Equation (1).

**Expectance map.** Probability distribution of crowd's activities defines the characteristics of locomotion that are likely to occur at specific locations. We define the distribution of a crowd's locomotion from time  $(t - n\Delta t)$  to  $(t - \Delta t)$  by pmf  $P_{\vec{v}}^{(t-n\Delta t) \rightarrow (t-\Delta t)}$  and the current distribution of a crowd's locomotion at time  $t$  by  $P_{\vec{v}}^t$ . We use these two pmfs in Equation (2) to calculate KL divergence values. These values constitute the second type of behavior map called the *expectance map*. Expectance map KL is defined as

$$KL^t = \{(D(P_{\vec{v}}^{(t-n\Delta t) \rightarrow (t-\Delta t)} \| P_{\vec{v}}^t))_{i,j} : 0 \leq i < w, 0 \leq j < h\}. \quad (4)$$

KL values indicate the difference between the current distribution and the cumulative distribution of a crowd's locomotion. Use of KL divergence values to indicate *surprise* is proposed in [33], where they use KL divergence values to discover *surprising* events in video. They employed a principled approach to prove that KL is a powerful measure to represent *surprise*. We use KL values to indicate unexpected, surprising crowd formations. In an expectance map, cells with high KL values denote *surprising* activities taking place at those locations. At cells with lower KL values the state of the crowd remains as *expected*. Both of these maps address different aspects in the locomotion of a crowd and each map has certain impacts on agents' behavior. Therefore, a behavior map  $B$  is the convex combination of the values of *entropy* and *expectance*

maps. This map can be formulated by

$$B^t = \{w_1 e_{i,j}^t + w_2 k l_{i,j}^t : 0 \leq w_n < 1, w_1 + w_2 = 1, 0 \leq i < w, 0 \leq j < h\}, \quad (5)$$

where each  $w_i$  represents user-defined weight values to determine the contribution of each map. These user-defined weights provide a mechanism to control how a behavior map is constructed according to the simulation scenario. A study to illustrate how  $w$  values alter simulation outcomes is described in Section 4.

### 3.2. Agent representation

Agent-based crowd simulators have access to several motion engines and animation sets which define behavioral output types. These types can range from basic behaviors like changing direction to complex behaviors like spreading shoulders to clear its path. The feature set of the crowd simulator and the underlying agent model define the complexity of agent behavior. In our behavioral model, we need a generic agent representation to fit into any type of agent-based crowd engine. Our agent representation includes two properties, (i) *behavior state* which enables interaction between agents and behavior maps and (ii) *behavior constants* to determine agents' behaviors in combination with the behavior state.

Behavior state  $\beta$  is the behavior map cell value assigned to an agent. Agents on the same cell of the map share the same behavior state. As behavior map values are altered temporally and spatially, these values are used in agent–crowd interaction formulations to adaptively control agents' behavior. Behavior constants  $f$  are agent-specific values which are evaluated as personality attributes. Each feature of an agent which we want to control adaptively is paired with a behavior constant. By assigning an  $f$  value, behavioral complexity of an agent is extended and, by varying  $f$  values, responses of agents to behavior map values are varied. Behavior constants can be regarded as a mechanism to create complexity and variation in a crowd. To wrap up these concepts with an example, assume a crowd simulator where agents have the feature of sweating, which we denote as  $p_0$ . In our representation, a behavior constant  $f_0$  defines how easily an agent sweats. And  $\beta$  values adaptively control when and where an agent will sweat. The agent representation is extended to include these properties, in addition to physical properties, which are position  $u$  and velocity  $v$ :

$$a_i = \{u, \bar{v}, \beta, \langle f_0, p_0 \rangle, \dots, \langle f_n, p_n \rangle : \beta, f_n \in [0, 1] \forall n\}. \quad (6)$$

Here  $p_n$  is a symbolic representation to indicate a feature associated with  $a_i$ . A single  $\langle f_n, p_n \rangle$  pair represents that  $p_n$  is controlled by  $f_n$ . Note that, for each  $\langle f_n, p_n \rangle$  pair, a formulation should be developed to define how the  $\beta$  and  $f_n$  values control  $p_n$ .

Our model is designed as a feedback mechanism as depicted in Fig. 1. In this mechanism, behavior maps contain only global information and this information is fed to the underlying crowd simulator to be utilized to modify local interactions. Thus local changes in the system alter the global state, and behavior maps are reconstructed with the altered global information. And finally, the behavior maps are fed back into the system to close the feedback loop is. To maintain the flow of this feedback mechanism, local interactions are considered as an output of our model rather than an input.

### 3.3. Extending a crowd simulator

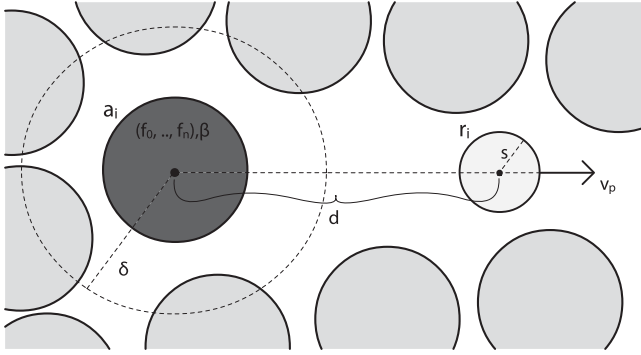
Any crowd simulator can be extended with our behavioral model. Our model introduces agent–crowd interactions into agent-based crowd simulators. To integrate our model, we first need to customize the agent definition given in Equation (6) according to the capabilities of the crowd simulator. This representation is then accompanied by formulations to define how agents handle behavior map values.

In this study, we use the RVO multi-agent navigation system introduced in [5]. We extended this system by implementing *composite agents* proposed in [34]. A composite agent  $a_i$  is a special agent equipped with a proxy agent  $r_i$  to model a number of emergent behaviors realistically. A proxy agent is a virtual agent, which is visible to all agents in the simulation except its parent  $a_i$ . Here  $r_i$  moves according to  $a_i$ 's preferences. For example, if  $a_i$  wants to move in a certain direction,  $r_i$  is placed in that direction to clear  $a_i$ 's path. With this mechanism  $a_i$  can display particular behaviors. The features  $p_n$  of the underlying simulation system (i.e. RVO) can be listed as follows:

- (i)  $d$ : distance between the proxy agent's position  $r_i[u]$  and  $a_i$ 's position  $u$ . The longer the distance, the further  $a_i$  can proceed with less collisions.
- (ii)  $s$ : radius of the circular area  $r_i$  occupies. The larger the area, the easier  $a_i$  can move.
- (iii)  $\bar{v}_p$ : this is the *preferred velocity* of an agent  $a_i$ . It is the optimal velocity that would bring the agent to its goal. We modify  $v_p$ 's direction with a normalized velocity vector  $\bar{v}_b$ , which is calculated with respect to behavior map values. Here  $\bar{v}_b$  is calculated as a vector leading to lower value zones found as a result of a local search.
- (iv)  $m$ : indicates the agent speed.
- (v)  $\delta$ : indicates the safety factor which is the range considered by an agent while calculating possible future collisions. With a high safety factor, an agent considers a higher number of possible collisions and behaves more carefully to make less collisions.

The agent representation proposed in Equation (6) is customized with respect to the features of the underlying simulator:

$$a_i = \{\text{type}, u, \bar{v}, r_i, \beta, \langle f_0, d \rangle, \langle f_1, s \rangle, \langle f_2, \bar{v}_p \rangle, \langle f_3, m \rangle, \langle f_4, \delta \rangle : \bar{v}_p \in \mathbb{R}^2; f_n, \beta, d, s \in \mathbb{R}\}, \quad (7)$$



**FIGURE 3.** A composite agent  $a_i$ , its associated proxy agent  $r_i$  and certain features of agent representation.

where type indicates whether the agent is a composite or proxy agent. Each  $f$  value with their associated feature is given as pairs. A figure to illustrate the customized agent definition can be seen in Fig. 3. The next step is developing the formulations to include the behavior state,  $\beta$ , and behavior constant,  $f$ , values. We develop formulations to represent agent–crowd interactions for agent  $a_i$  as follows:

$$\begin{aligned} \beta &= B_{i,j}, \\ d &= \sqrt{f_0 \beta} d_{\max} + d_{\min}, \\ s &= \sqrt{f_1 \beta} s_{\max} + s_{\min}, \\ \vec{v}_p &= (\vec{v}_p^o + \sqrt{f_2 \beta} \vec{v}_b) (\sqrt{f_3 \beta} m_{\max} + m_{\min}), \\ \delta &= \sqrt{f_4 \beta} \delta_{\max} + \delta_{\min}, \end{aligned} \quad (8)$$

where  $B_{i,j}$  is the current behavior map value at cell  $\{i,j\}$  and  $\vec{v}_p^o$  is the optimal velocity leading to the agent's goal. Each property has a user-defined min and max value to keep the values in a certain range.

## 4. RESULTS AND TEST CASES

Our behavioral model extends the variability and complexity of a crowd simulator. We run a number of tests to demonstrate how our model enhances a crowd simulator's performance. The tests were run on a system with Intel QuadCore 2.8 GHz and Nvidia GeForce GTX-280. Formulations in our behavioral model are constituted by simple calculations, therefore we observed that integration of our model into a crowd simulator does not bring significant computational overload. The number of agents that can be simulated with our model is limited by the crowd simulator used in our simulations.

### 4.1. Test environment

We define analogies between the interpretations of analytical maps with  $f$  values in order to produce realistic crowd

**TABLE 1.** Analytical maps and their interpretation.

Analytical map	Behavioral interpretation
Entropy	Predictability
Expectance	Surprise

**TABLE 2.** Behavior types, related features and  $f$  values associated with these features.

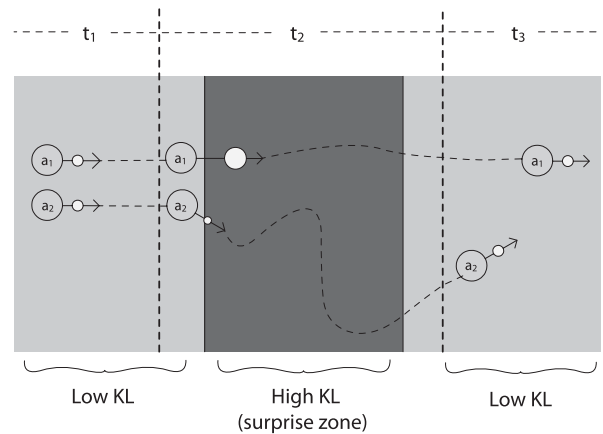
Behavior type	Feature	$f$
Carefulness	$\delta$	$f_4$
Aggressiveness	$d, s, \vec{v}_p, m$	$f_0, f_1, f_2, f_3$

simulations. We interpret the analytical maps of our model as seen in Table 1.

In the test environment, agents can have aggressiveness and/or carefulness properties. To create certain agents which are aggressive and careful, we relate features of agents and formulations with  $f$  and  $\beta$  values. In Table 2, these behavior types with their related features and  $f$  values are listed.

The interpretations of behavior maps are used to define how agents respond to them. In areas with high entropy, where agents' locomotions are diverse, agents become more careful to avoid collisions, and they become more aggressive to get through these regions as quickly as possible. As the expectance map indicates the level of *surprise* in a specific location, aggressive agents do not panic and behave more goal-oriented by preserving their optimal velocity  $\vec{v}_p^o$  and enlarge  $s$ ,  $d$  and  $m$  values in order to display their aggressiveness. On the other hand, high  $KL$  values make an agent less careful. Note that, while carefulness is proportional to entropy values, it is inversely proportional to expectance values.

Figure 4 illustrates how agents respond to the expectance map ( $w_1 = 0$ ;  $w_2 = 1$  in Equation (5)) at the micro level. In



**FIGURE 4.** Responses of agents to the expectance map.

this figure,  $a_1$  is an aggressive agent and  $a_2$  is a calm agent. In time interval  $t_1$ ,  $a_1$  and  $a_2$  behave identical. In  $t_2$ , they enter a high KL zone;  $a_1$  responds by enlarging  $s$  and  $d$  values to keep its  $\bar{v}_p$  as close to optimal as possible. However,  $a_2$  mimics a panicking behavior and behaves in an unexpected manner. At  $t_3$ , agents return to their initial state. Note that at the end of  $t_3$ ,  $a_1$  proceeds further.

In addition to the test cases, we also include certain plots (Figs. 5–7) to determine how certain parameters affect our

system’s outcome. This sort of analysis is required to assess the performance and quality of the behavioral model in evacuation and crowd simulations. Similar analysis of behavior models can be found in studies by Filippopolitis et al. [8, 35].

To illustrate how  $f$  values affect agents’ behavior, we run the same simulation with varying  $f$  values. Figure 5 displays the average speed of the agents when the simulation is run with different values for  $f_0, f_2, f_3$  and  $f_4$ . We observe that  $f_3$  values are positively correlated with the average speed as it directly alters the speed of an agent. On the other hand,  $f_4$  values which alter the *carefulness* of an agent negatively affect agents’ speed. As  $f_4$  increases, the agents become more and more careful. This makes them harder to move as they avoid other agents more than before. It is also clear from the image that  $f_2$  values do not have an impact on agents’ speed as it modifies the direction of the agent rather than its speed.

4.2. Test cases

Our model and the underlying crowd simulator require a number of parameters to be set before performing a test. We build a graphical user interface (GUI) based editor to interactively enter behavior constants and crowd simulator parameters. This authoring tool enables the designer to disperse  $f$  values over the agents to create variation in a crowd interactively. The  $w_n$  values in Equation (5) determine the contribution of the behavior map’s components, therefore weights of each map should be adjusted according to the simulation scenario. To investigate the effect of  $w$  values on behavior map construction, we run two separate scenarios using different  $w$  combinations and present the results in Fig. 6. In this plot,  $w_1$  modifies the entropy map contribution and  $w_2 = 1 - w_1$ . Scenario-1 in this test is the scenario in Test-4 where two groups of agents meet and scenario-2 is the one in Test-1 where a group of agents evacuate a room. We can clearly see that a high  $w_1$  value is more suitable for the first scenario because when these groups meet, a high entropy zone is formed and the contribution of the entropy map on  $B$  should be emphasized. On the other hand,  $w$  values does not alter simulation times in the second scenario;  $w$  values should be decided prior to the simulation with respect to the simulation scenario. In the following tests, we observe that equal  $w_n$  ( $w_1 = 0.5; w_2 = 0.5$ ) values for each map performed successfully in most of the scenarios. Results of the following tests can be found in the video available at [http://students.sabanciuniv.edu/emrekoc/iscis2010/iscis2010\\_turkey\\_et\\_al.avi](http://students.sabanciuniv.edu/emrekoc/iscis2010/iscis2010_turkey_et_al.avi). A screenshot from our test environment can be seen in Fig. 8.

*Test-1.* We perform a test to prove the validity of our approach by a comparison with a real-world scenario. In this test, our behavior model is integrated into an RVO library and the resulting simulations are compared with real-world data. We used room evacuation videos and data produced in Research Center Jülich, Germany and made available in [36]. These

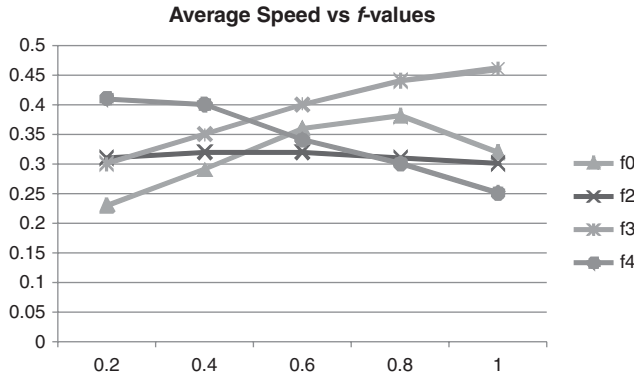


FIGURE 5. Average speed of agents over varying  $f$  values.

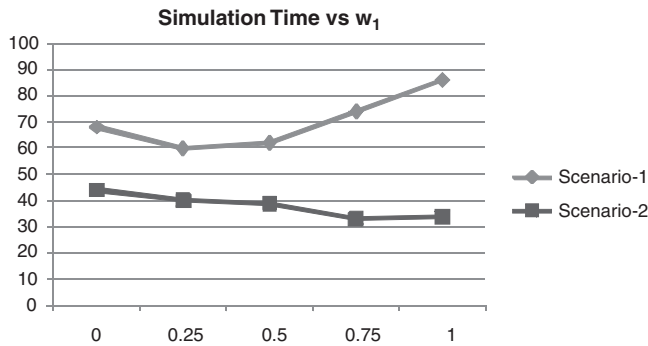


FIGURE 6. Simulation times of two scenarios under different  $w_1$  assignments;  $w_1$  determines the contribution of the entropy map in behavior map construction.

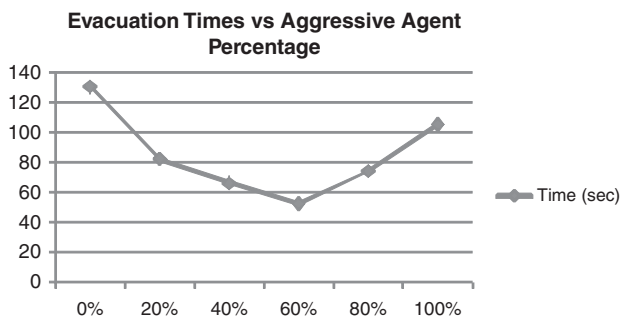
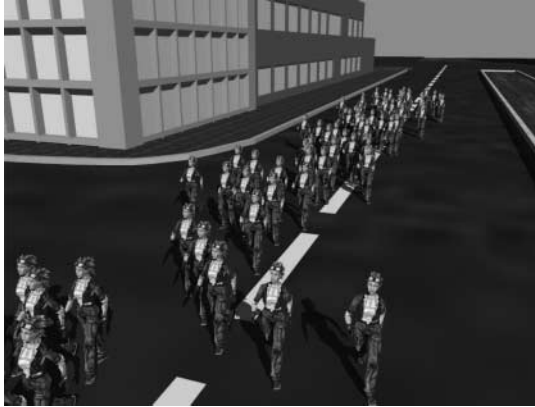


FIGURE 7. Simulation times of the evacuation scenario with different aggressive agent percentages.

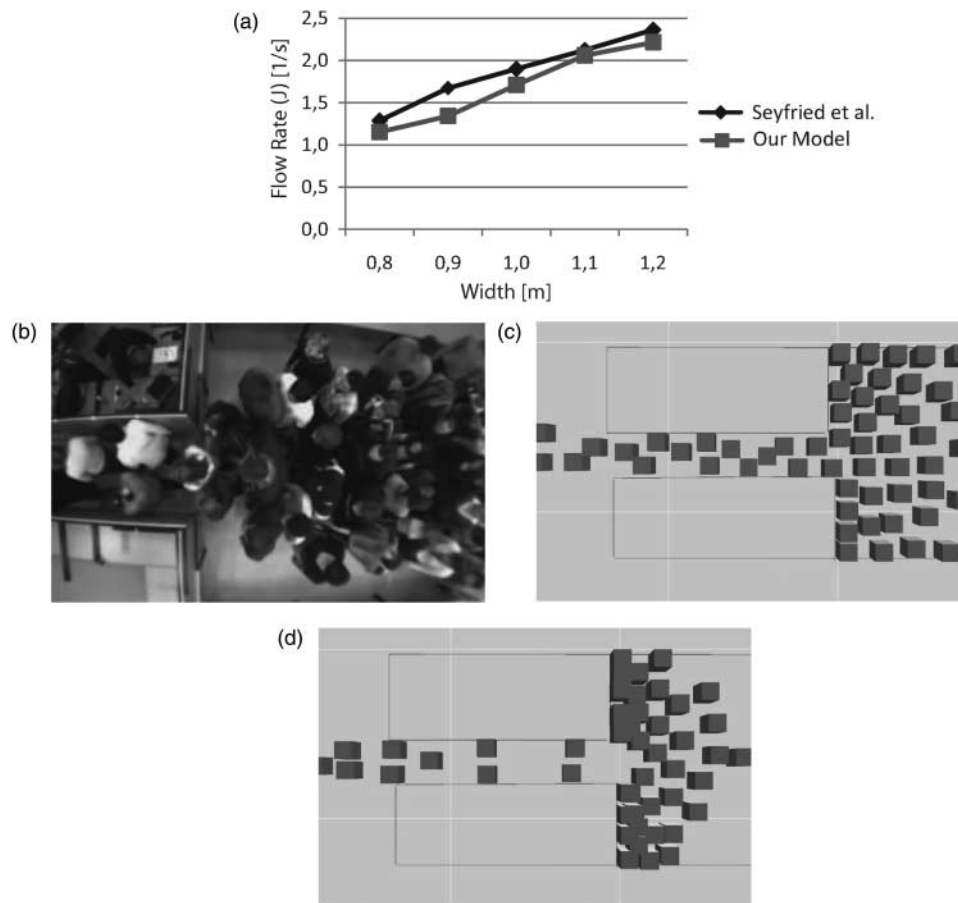


**FIGURE 8.** A screenshot from our test environment.

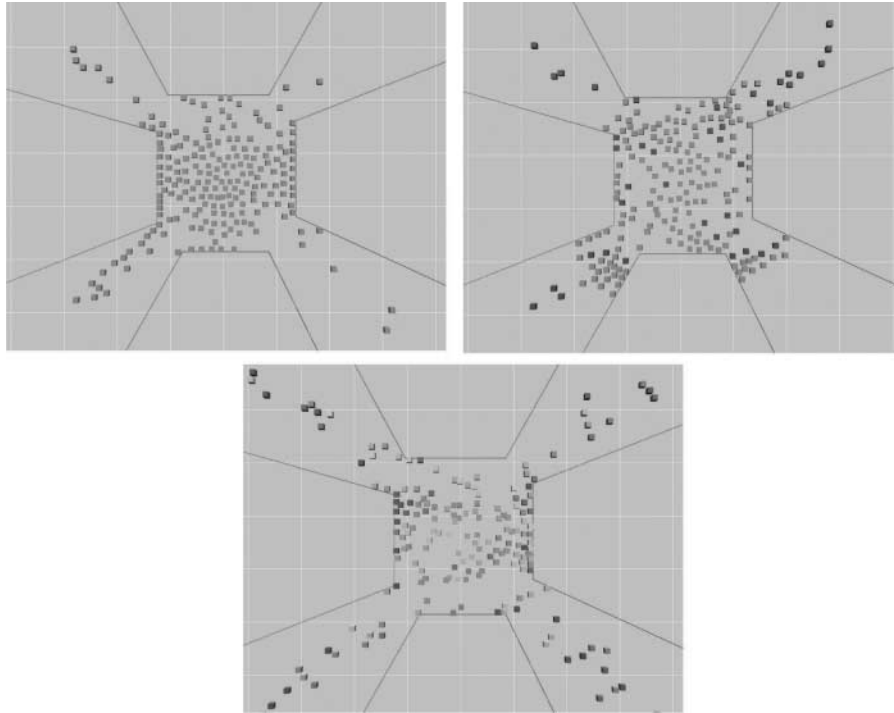
videos measure the flow of 60 students while evacuating a room with a variable exit width. We measure the flow of our agents with the formula  $J = \Delta N / \Delta t$ , where  $N$  is the number of agents and  $\Delta t$  is calculated as the difference between the evacuation times of the first and the last agent. As the video incorporates

students evacuating the room calmly, we set low aggressiveness to our agents. We observe that our results are consistent with the real-world case (Fig. 9). We made further studies with this scenario setting and instead of adding calm agents, we add aggressive agents into the room. Agents are competing more to get out quickly in this case; as a result, clogging occurred at the exit (Fig. 9). An evaluation of how aggressive agents affect simulation times is displayed in Fig. 7. When there are only calm agents in the simulation, the final evacuation times are higher as agents act very slowly. As we add more aggressive agents, the evacuation times are lowered as aggressive agents proceed through the gate more quickly. However, after a certain percentage (around 60%) of aggressive agents, the evacuation times become greater as their aggressiveness causes clogging around the exit.

*Test-2.* We made comparison tests with two agent-based crowd simulators. The first one is the flocking model developed by Reynolds [37] and the second is the RVO library, which we also used as the underlying navigation library in this paper [5]. These comparative tests incorporate a scenario where four groups of agents walk through at a piazza. Throughout these tests, we



**FIGURE 9.** (a) Flow vs. width of exit. (b) Real-world scenario. (c) Our test environment with less aggressive agents. (d) Clogging occurs when agents are more aggressive.



**FIGURE 10.** Agent diversity and complexity of crowd behavior can be adjusted with our model (left to right: uniform, 80% calm 20% aggressive, diverse agents).

create a crowd with varied  $f$  values in our crowd simulator, and this creates diversity in the crowd's behaviors. In other models, agents do not respond to the dynamics of the crowd and behave identically.

*Test-3.* We run the same scenario from Test-2 incorporating a crowd consisting of (i) only calm (not aggressive) agents, (ii) 20% aggressive agents and (iii) agents with various  $f$  values. Figure 10 displays the results of these tests. We see that only by varying the dispersion of  $f$  values, our model is capable of creating diverse and realistic results without requiring any additional scripting or editing effort.

*Test-4.* We adopt a scenario where two groups of agents move toward each other. This scenario highlights the function of entropy maps ( $w_1 = 1$ ;  $w_2 = 0$  in Equation (5)). Before these groups meet, they do not display aggressive behavior as they produce a behavior map zone with low entropies. However, when these groups meet, there is a high level of disorder and entropy values increase. This variance in crowd formation adaptively modifies agents' responses and they start behaving aggressively.

## 5. CONCLUSION

In this paper, we presented a novel analytical behavioral model which automatically builds behavior maps to control

agents' behavior adaptively with agent-crowd interaction formulations. Probabilistic methods incorporating information theory quantities have been used to produce these behavior maps.

We did a comparative analysis of the presented behavior model with measured crowd data and two agent-based crowd simulators. We also run several well-known test scenarios to demonstrate the performance of our model.

The presented behavioral model can be integrated into existing agent-based crowd simulators and improve the complexity of resulting crowd behavior. In most of the crowd simulators, low-level scripts are developed to drive complex agent behaviors. The analytical maps produced in our model are utilized to control these behaviors automatically. An important advantage of the proposed model lies in reducing the time spent on creating agents displaying diverse behaviors.

As a future work, we will expand the scope of behavior map construction methods with different quantities from information theory and related fields. These maps can broaden our model with new interpretations and results. In this paper, we only integrated our model into agent-based simulators and used behavior maps to control individual agents. We will try to integrate our model into simulators which solve crowd simulations with global approaches [1]. We believe that our analytical maps will also provide information to control crowds globally.

## REFERENCES

- [1] Treuille, A., Cooper, S. and Popović, Z. (2006) Continuum crowds. *ACM Trans. Graph.*, **25**, 1160–1168.
- [2] Musse, S.R. and Thalmann, D. (2001) Hierarchical model for real time simulation of virtual human crowds. *IEEE Trans. Vis. Comput. Graphics*, **7**, 152–164.
- [3] Sommer, R. and Sommer, B. (2002) *A Practical Guide to Behavioral Research: Tools and Technique* (5th edn). Oxford University Press.
- [4] Turkay, C., Koc, E. and Balcisoy, S. (2009) An information theoretic approach to camera control for crowded scenes. *Vis. Comput.*, **25**, 451–459.
- [5] van den Berg, J., Lin, M. and Manocha, D. (2008) Reciprocal Velocity Obstacles for Real-Time Multi-agent Navigation. *IEEE Int. Conf. on Robotics and Automation, ICRA 2008*, May 19–23, Pasadena, CA, USA, pp. 1928–1935.
- [6] Filippopolitis, A., Loukas, G., Timotheou, S., Dimakis, N. and Gelenbe, E. (2009) Emergency Response Systems for Disaster Management in Buildings. *Proc. NATO Symp. C3I for Crisis, Emergency and Consequence Management*, Bucharest, Romania, May, pp. 11–12.
- [7] Filippopolitis, A. and Gelenbe, E. (2009) A Decision Support System for Disaster Management in Buildings. *Proc. Summer Computer Simulation Conf. (SCSC 09)*, Istanbul, Turkey, July, pp. 141–147. SCS, San Diego, CA.
- [8] Filippopolitis, A. and Gelenbe, E. (2009) A Distributed Decision Support System for Building Evacuation. *HSI'09: Proc. 2nd Conf. Human System Interactions*, Piscataway, NJ, USA, pp. 320–327. IEEE Press.
- [9] Thalmann, D. and Raupp Musse, S. (2007) *Crowd Simulation*. Springer.
- [10] Kasap, Z. and Magnenat-Thalmann, N. (2008) Intelligent Virtual Humans with Autonomy and Personality: State-of-the-Art. *New Advances in Virtual Humans*. Studies in Computational Intelligence. Springer.
- [11] Reynolds, C. (1999) Steering Behaviors for Autonomous Characters. *Game Developers Conf.*, San Jose, CA, USA.
- [12] Reynolds, C. (1987) Flocks, Herds and Schools: A Distributed Behavioral Model. *Proc. 14th Annual Conf. Computer Graphics and Interactive Techniques*, Anaheim, CA, USA, pp. 25–34.
- [13] Luo, L., Zhou, S., Cai, W., Low, M.Y.H., Tian, F., Wang, Y., Xiao, X. and Chen, D. (2008) Agent-based human behavior modeling for crowd simulation. *Comput. Animat. & Virtual Worlds*, **19**, 271–281.
- [14] Gelenbe, E., Kaptan, V. and Hussain, K. (2004) Simulating the Navigation and Control of Autonomous Agents. *Fusion 2004: 7th Int. Conf. Information Fusion; Stockholm*. International Society of Information Fusion, ONERA-DTIM, BP 72, 29 Av. de la Division Leclerc, Chatillon, 92320, France.
- [15] Rymill, S. and Dodgson, N. (2005) A Psychologically-Based Simulation of Human Behaviour. *Theory and Practice of Computer Graphics*, Canterbury, Kent, UK, pp. 35–42.
- [16] Pelechano, N., O'Brien, K., Silverman, B. and Badler, N. (2005) Crowd Simulation Incorporating Agent Psychological Models, Roles and Communication. *First Int. Workshop on Crowd Simulation*, Lausanne, Switzerland.
- [17] Gelenbe, E., Kaptan, V. and Wang, Y. (2004) Biological Metaphors for Agent Behavior. *Computer and Information Sciences-ISCIS 2004*, Kemer-Antalya, Turkey, Vol. 3280, pp. 667–675.
- [18] Gelenbe, E. and Wang, Y. (2006) A mathematical approach for mission planning and rehearsal. *Proc. SPIE Def. Secur.*, **6249**, 197–207.
- [19] Pelechano, N., Allbeck, J.M. and Badler, N. I. (2007) Controlling Individual Agents in High-Density Crowd Simulation. *Proc. 2007 ACM SIGGRAPH/Eurographics Symp. Computer Animation*, San Diego, CA, USA, pp. 99–108. Eurographics Association.
- [20] Shao, W. and Terzopoulos, D. (2007) Autonomous pedestrians. *Grav. Models*, **69**, 246–274.
- [21] Gayle, R., Sud, A., Andersen, E., Guy, S., Lin, M. and Manocha, D. (2009) Interactive Navigation of Heterogeneous Agents Using Adaptive Roadmaps. *IEEE Trans. Vis. Comput. Graphics*, **15**, 34–48.
- [22] Sung, M., Gleicher, M. and Chenney, S. (2004) Scalable Behaviors for Crowd Simulation. *Computer Graphics Forum*, Grenoble, France, pp. 519–528.
- [23] Kaptan, V. and Gelenbe, E. (2006) Fusing Terrain and Goals: Agent Control in Urban Environments. *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, Orlando, FL, USA, Vol. **6242**, 71–79.
- [24] Dimakis, N., Filippopolitis, A. and Gelenbe, E. (2010) Distributed building evacuation simulator for smart emergency management. *Comput. J.*, **53**, 1384–1400.
- [25] Dornhege, C. and Kleiner, A. (2007) Behavior Maps for Online Planning of Obstacle Negotiation and Climbing on Rough Terrain. *IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2007. IROS 2007*, San Diego, CA, USA, pp. 3005–3011.
- [26] Berclaz, J., Fleuret, F. and Fua, P. (2008) Multi-camera Tracking and Atypical Motion Detection with Behavioral Maps. *ECCV '08: Proc. 10th European Conf. Computer Vision*, Berlin, Heidelberg, pp. 112–125. Springer.
- [27] Gelenbe, E., Hussain, K. and Kaptan, V. (2005) Simulating autonomous agents in augmented reality. *J. Syst. Softw.*, **74**, 255–268.
- [28] Feixas, M., del Acebo, E., Bekaert, P. and Sbert, M. (1999) An Information Theory Framework for the Analysis of Scene Complexity. *Computer Graphics Forum*, pp. 95–106. John Wiley & Sons.
- [29] Turkay, C., Koc, E., Yuksel, K. and Balcisoy, S. (2009) Adaptive Behavioral Modeling for Crowd Simulations. *Proc. Int. Conf. Computer Animation and Social Agents (Short Papers)*, Amsterdam, The Netherlands, pp. 65–68.
- [30] Turkay, C., Koc, E. and Balcisoy, S. (2010) An Information Theory Based Behavioral Model for Agent-Based Crowd Simulations. *Computer and Information Science: Proc. 25th Int. Symp. Computer and Information Sciences*, London, UK, pp. 379–382. Springer.
- [31] Shannon, C.E. (1948) A mathematical theory of communication. *Bell Syst. Tech. J.*, **27**, 623–656.

- [32] Kullback, S. (1997) *Information Theory and Statistics (Dover Books on Mathematics)*. Dover Publications.
- [33] Itti, L. and Baldi, P. (2005) A Principled Approach to Detecting Surprising Events in Video. *IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, San Diego, CA, USA, pp. 631–637. IEEE Computer Society.
- [34] Hengchin, Y., Sean, C., Sachin, P., Jur, V.D.B., Dinesh, M. and Ming, L. (2008) Composite Agents. *Symp. Computer Animation—SCA'08*, Dublin, Ireland.
- [35] Filippoupolitis, A., Hey, L., Loukas, G., Gelenbe, E. and Timotheou, S. (2008) Emergency Response Simulation Using Wireless Sensor Networks. *Proc. 1st Int. Conf. Ambient Media and Systems*, Quebec, Canada, pp. 1–7. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [36] Ped-Net (2010). <http://ped-net.org/>.
- [37] OpenSteer (2010). <http://opensteer.sourceforge.net/>.