An Information Theory Based Behavioral Model for Agent-Based Crowd Simulations

Cagatay Turkay[1], Emre Koc[2], and Selim Balcisoy[2]

[1] University of Bergen, Bergen, Norway
[2] Sabanci University, Istanbul, Turkey
Cagatay.Turkay@ii.uib.no, emrekoc@su.sabanciuniv.edu,
balcisoy@sabanciuniv.edu

Abstract. 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.

1 Introduction

Crowd constitutes a critical component in many virtual environment and entertainment applications. Believable behavior and appearance of a crowd improves a virtual environments 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 either propose 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 requiring 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 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 analytic 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 which 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 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 (Figure 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].

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 at the conclusion section.

2 Related Work

An overall idea of the challenges and improvements in crowd simulation can be obtained in [6] by Thalmann et.al.. There are several behavioral models proposed in the literature and a survey by Kasap et.al. [7] 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 [8] and create emergent flocking [9] behaviors. There is considerable work on agent-based crowd simulators incorporating psychological models and sociological factors. In [10], Luo et.al. model social group and crowd related behaviors. Their main focus is a layered framework to reflect the natural pattern

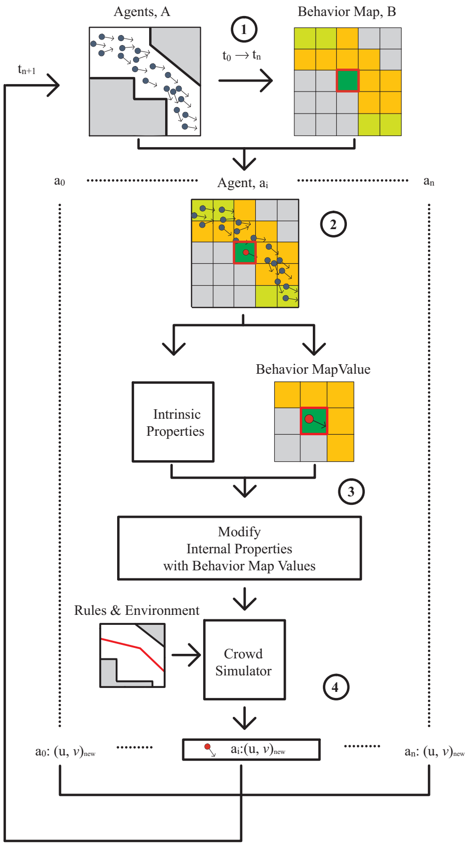

Fig. 1. Overall structure of our model. 1) Behavior map is produced 2) Agents are assigned a specific cell value 3) Agent's intrinsic properties are modified with behavior map value 4) Agents' locomotion is determined by simulator

of human-like decision making process. In [11], Rymill et.al. 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 et.al. [2] developed virtual human agents with intentions, beliefs, knowledge and perception to create a realistic crowd behavior. In [12], Pelechano et.al. assigned psychological roles and communication skills to agents to produce diverse and realistic behaviors. In a more recent work, [13] created an improved model by using psychological and geometrical rules with a social and physical forces model. There are studies which model the virtual environment as maps to guide agents' behaviors. In [14] Shao et.al. modeled the environment with topological, perception and path maps to generate autonomous agents. [15] used adaptive roadmaps, which evolve with the dynamic nature of the environment. In [16], Sung et.al. assign situations and behaviors directly to environment rather than the agents themselves. The concept of behavior maps have been used in robotics and vision field. [17] defined behavior maps as encoding context information of the environment, and use these maps to autonomously navigate a robot on rough terrain. Berclaz et.al. [18] used behavior maps to encode probabilities of moving in a certain direction on a specified location and used these maps to track trajectories of people and to detect anomalies in people's behaviors. In their study, they used expectation maximization algorithms to detect anomalies.

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 have been introduced into computer graphics field by Feixas et.al. [19] 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. In [20], they extended their information theory based model to control how agents behave in a crowd simulation. We improve the behavioral model proposed in [20] 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. 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 the 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 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 $A = \{a_1, a_2, ..., a_n\}$ 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 $a_i = \{u, \boldsymbol{v} \ : u, \boldsymbol{v} \in \mathbb{R}^2\}$ where u defines the position and $\boldsymbol{v}$ defines the velocity of agent a_i. The activity of an agent is described by: i) its position ii) direction and magnitude of its velocity vector. 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 wxh 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 crowd's locomotion.

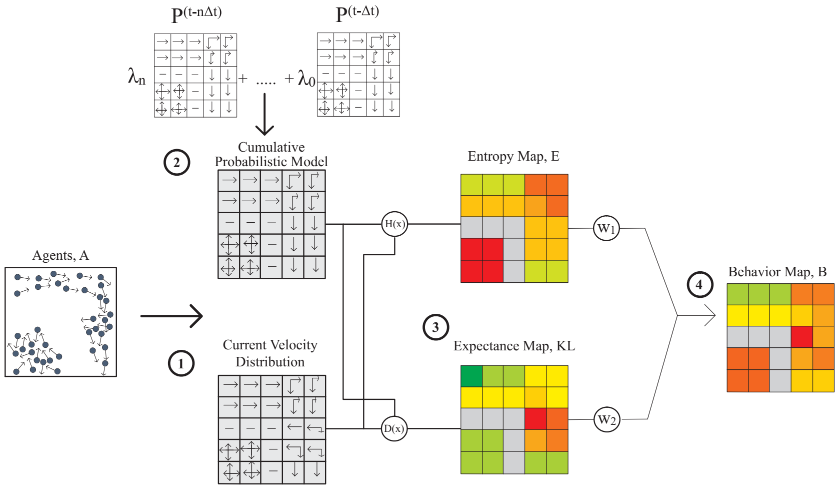

Fig. 2. Behavior map construction. 1) Current distribution crowd locomotion 2) Older distributions are merged with a temporal filter 3) *Entropy* and *expectance maps* are calculated 4) Behavior map is constructed with convex combination using user-defined weights

Behavior Map Construction

Information theory deals with the quantification of information. *Information entropy* is the key measure in information theory [21] 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 χ 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 *Kullback-Leibler divergence* (KL) [22]. Take two probability mass functions *(pmf)* $p(x)$ and $q(x)$, divergence between *pmf*s $p(x)$ and $q(x)$ is given by:

$$D(p\|q) = -\sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \tag{2}$$

which is a non-symmetric metric expressing the difference between two probability distributions. Specifically, it represents how probable the distribution defined by $q(x)$ is likely to occur when $p(x)$ is given.

In order to use these information theory quantities, we use a *pmf* denoted as $P_{\boldsymbol{v}}$, to characterize the distribution of crowd's locomotion. Velocity of each agent is added as a sample to the 2D *pmf.* As $P_{\boldsymbol{v}}$ 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 ${P_{\boldsymbol{v}}}^{t_1 \to t_2}$ 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 *entropy* and *expectance map.*

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 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* $({P_{\boldsymbol{v}}}^{(t-n\Delta t)\to t})_{i,j}$. Then, E can be defined as;

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

, 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 crowd's locomotion from time $(t - n\Delta t)$ to $(t - \Delta t)$ by *pmf* ${P_{\boldsymbol{v}}}^{(t-n\Delta t)\to(t-\Delta t)}$ and the current distribution of crowd's locomotion at time t by ${P_{\boldsymbol{v}}}^{t}$. We use these two *pmf*s in Equation 2 to calculate KL divergence values. These values constitute the second type of behavior map called *expectance map.* Expectance map KL is defined as;

$$KL^t = \{(D({P_{\boldsymbol{v}}}^{(t-n\Delta t)\to(t-\Delta t)} \| {P_{\boldsymbol{v}}}^{t}))_{i,j} :\; 0 \le i < w,\; 0 \le j < h\} \tag{4}$$

KL values indicate the difference between the current distribution and the cumulative distribution of crowd's locomotion. Use of KL divergence values to indicate *surprise* is proposed in [23], 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 remain as *expected*. Both of these maps address different aspects in the locomotion of crowd and each map has certain effects on agent's behavior. Therefore, a behavior map, B, is the convex combination of the values of *entropy* and *expectance* maps. This map can be formulated by;

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

, where each w_i represents user-defined weight values to determine the contribution of each map. These user-defined weights provides a mechanism to control how a behavior map is constructed according to the simulation scenario.

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 behavior state.

Behavior state, β, 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 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 easy an agent sweats. And β 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:

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

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

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. In order 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 with formulations to define how agents handle behavior map values.

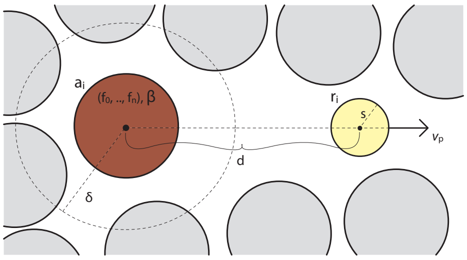

Fig. 3. A composite agent a_i, its associated proxy agent r_i and certain features of agent representation

In this study, we use Reciprocal Velocity Obstacles (RVO) multi-agent navigation system introduced in [5]. We extended this system by implementing *composite agents* proposed in [24]. 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. 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;

- d : Distance between 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.
- s : Radius of the circular area r_i occupies. The larger the area, the easier a_i can move.
- $\boldsymbol{v_p}$: 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, $\boldsymbol{v_b}$, which is calculated with respect to behavior map values. $\boldsymbol{v_b}$ is calculated as a vector leading to lower value zones found as a result of a local search.

- m : Indicates agent speed.
- δ : Indicates 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 careful to make less collisions.

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

$$\begin{aligned} a_i = \{ & type, u, \boldsymbol{v}, r_i, \beta, \langle f_0, d \rangle, \langle f_1, s \rangle, \langle f_2, \boldsymbol{v}_p \rangle, \\ & \langle f_3, m \rangle, \langle f_4, \delta \rangle : \boldsymbol{v}_p \in \mathbb{R}^2; f_n, \beta, d, s \in \mathbb{R} \} \end{aligned} \tag{7}$$

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 Figure 3. The next step is developing the formulations to include behavior state, β, and behavior constants, f, values. We develop formulations to represent agent-crowd interactions for agent a_i as:

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

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

4 Results & 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 constitute of simple calculations, therefore we observed that integration of our model into a crowd simulator does not bring significant computational overload. The number of agents which 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 simulations. We interpret the analytical maps of our model as seen in Table 1.

Table 1. Analytical maps and their interpretation

Analytical Map	Behavioral Interpretation
Entropy	Predictability
Expectance	Surprise

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 β 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, $\boldsymbol{v}_p^o$, 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. Notice that, while carefulness is proportional to entropy values, it is inversely proportional to expectance values.

Figure 4 illustrates how agents respond to expectance map ($w_1 = 0; w_2 = 1$ in Equation 5) at micro level. In 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 $\boldsymbol{v}_p$ as close as possible to optimal. However, a_2 mimics a panicking behavior and behaves in an unexpected manner. At t_3, agents return to their initial state. Notice that at the end of t_3, a_1 proceeds further.

Table 2. Behavior types, related features and f values associated with these features

Behavior Type	Feature	f
Carefulness	δ	f_4
Aggressiveness	$d, s, \boldsymbol{v}_p, m$	f_0, f_1, f_2, f_3

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 GUI-based editor to interactively enter behavior constants and crowd simulator parameters. This authoring tool

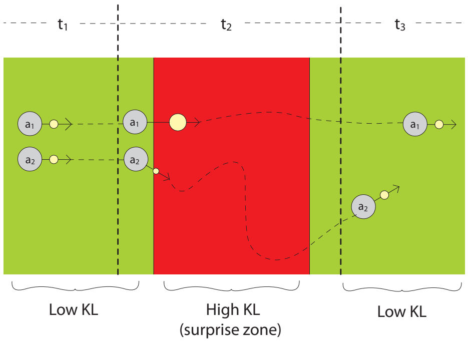

Fig. 4. Responses of agents to expectance map

enables the designer to disperse f values over the agents to create variation in crowd interactively. w_n values in Equation 5 determine the contribution of behavior map's components, therefore weights of each map should be adjusted according to the simulation scenario. In our 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 accompanying video. A screenshot from our test environment can be seen in Figure 7.

Test - 1 We perform a test to prove the validity of our approach by a comparison with a real world scenario. We used room evacuation videos and data produced in Research Center Jülich, Germany and made available in [25]. These 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 Δt 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 (Figure 5). 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 through the exit (Figure 5).

Test - 2 We made comparison tests with two agent-based crowd simulators. The first one is the flocking model developed by Reynolds [26] 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 test, we create a crowd with varied f values in our crowd simulator and this creates diversity in crowd's behaviors. In other models, agents do not respond to the dynamics of the crowd and behave identically.

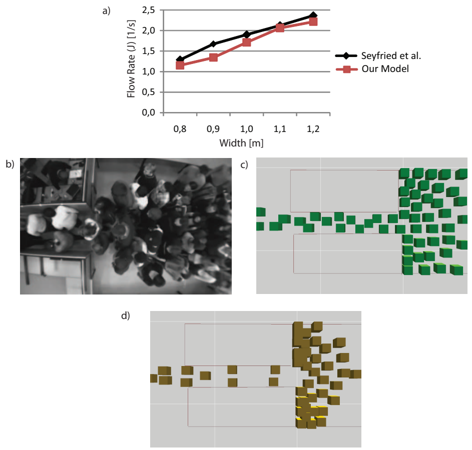

Fig. 5. a) Flow vs. width of exit b) Real-world scenario c) Our test environment with less aggressive agents d) Clogging occurs when agents are more aggressive

Test - 3 We run the same scenario from Test - 2 incorporating a crowd consisting of i) only calm (not aggressive) agents ii) 10% aggressive agents and iii) agents with various f values. Figure 6 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 towards 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 aggressive.

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 analyses 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.

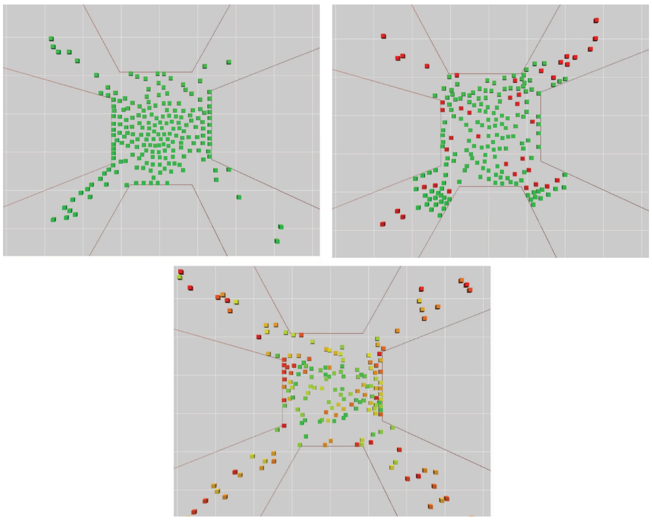

Fig. 6. Agent diversity and complexity of crowd behavior can be adjusted with our model (left to right: uniform, 20% calm 80% aggressive, diverse agents)

Fig. 7. A screenshot from our test environment

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 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. Adrien Treuille, Seth Cooper, and Zoran Popović. Continuum crowds. *ACM Trans. Graph.*, 25(3):1160–1168, 2006.
2. S. R. Musse and D. Thalmann. Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics*, 7:152–164, 2001.
3. R. Sommer and B. Sommer. *A Practical Guide To Behavioral Research: Tools and Technique*. Oxford University Press, 5 edition, 2002.
4. Cagatay Turkay, Emre Koc, and Selim Balcisoy. An information theoretic approach to camera control for crowded scenes. *Vis. Comput.*, 25(5-7):451–459, 2009.
5. J. van den Berg, M. Lin, and D. Manocha. Reciprocal Velocity Obstacles for real-time multi-agent navigation. In *Robotics and Automation, 2008.*, pages 1928–1935, 2008.
6. Daniel Thalmann and Soraia Raupp Musse. *Crowd Simulation*. Springer, 2007.
7. Z. Kasap and N. Magnenat-Thalmann. *Intelligent Virtual Humans with Autonomy and Personality : State-of-the-Art*, pages 43–84. Studies in Computational Intelligence, Springer, 2008.
8. C.W. Reynolds. Steering behaviors for autonomous characters. In *Game Developers Conference*, 1999.
9. C.W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, 1987.
10. Linbo Luo, Suiping Zhou, Wentong Cai, Malcolm Yoke Hean Low, Feng Tian, Yongwei Wang, Xian Xiao, and Dan Chen. Agent-based human behavior modeling for crowd simulation. *Comput. Animat. Virtual Worlds*, 19(3-4):271–281, 2008.
11. S.J. Rymill and N.A. Dodgson. A Psychologically-Based Simulation of Human Behaviour. In *Theory and Practice of Computer Graphics*, pages 35–42, 2005.
12. N. Pelechano, K. OBrien, B. Silverman, and N. Badler. Crowd simulation incorporating agent psychological models, roles and communication. In *First International Workshop on Crowd Simulation*, 2005.
13. N. Pelechano, J. M. Allbeck, and N. I. Badler. Controlling individual agents in high-density crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 99–108. Eurographics Association, 2007.
14. W. Shao and D. Terzopoulos. Autonomous pedestrians. *Graphical Models*, 69(5-6):246–274, 2007.
15. R. Gayle, A. Sud, E. Andersen, S.J. Guy, M.C. Lin, and D. Manocha. Interactive Navigation of Heterogeneous Agents Using Adaptive Roadmaps. *Visualization and Computer Graphics, IEEE Transactions on*, 15(1):34–48, 2009.

16. M. Sung, M. Gleicher, and S. Chenney. Scalable behaviors for crowd simulation. In *Computer Graphics Forum*, volume 23, pages 519–528, 2004.
17. C. Dornhege and A. Kleiner. Behavior maps for online planning of obstacle negotiation and climbing on rough terrain. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3005–3011, 2007.
18. Jérôme Berclaz, François Fleuret, and Pascal Fua. Multi-camera tracking and atypical motion detection with behavioral maps. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 112–125, Berlin, Heidelberg, 2008. Springer-Verlag.
19. M. Feixas, E. del Acebo, P. Bekaert, and M. Sbert. An information theory framework for the analysis of scene complexity. In *Computer Graphics Forum*, volume 18, pages 95–106. Citeseer, 1999.
20. Cagatay Turkay, Emre Koc, Kamer Yuksel, and Selim Balcisoy. Adaptive behavioral modeling for crowd simulations. In *Proceedings of International Conference on Computer Animation and Social Agents*, pages 65–68, 2009.
21. C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:623–656, 1948.
22. Solomon Kullback. *Information Theory and Statistics (Dover Books on Mathematics)*. Dover Publications, 1997.
23. Laurent Itti and Pierre Baldi. A principled approach to detecting surprising events in video. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 631–637. IEEE Computer Society, 2005.
24. Yeh Hengchin, Curtis Sean, Patil Sachin, van den Berg Jur, Manocha Dinesh, and Lin Ming. Composite agents. In *Symposium on Computer Animation - SCA'08*, 2008.
25. Ped-Net, May 2009. `http://ped-net.org/`.
26. OpenSteer, 2009. `http://opensteer.sourceforge.net/`.