Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2007)
D. Metaxas and J. Popovic (Editors)

Group Behavior from Video: A Data-Driven Approach to Crowd Simulation

Kang Hoon Lee, Myung Geol Choi, Qyoun Hong and Jehee Lee

Seoul National University†

Abstract

Crowd simulation techniques have frequently been used to animate a large group of virtual humans in computer graphics applications. We present a data-driven method of simulating a crowd of virtual humans that exhibit behaviors imitating real human crowds. To do so, we record the motion of a human crowd from an aerial view using a camcorder, extract the two-dimensional moving trajectories of each individual in the crowd, and then learn an agent model from observed trajectories. The agent model decides each agent's actions based on features of the environment and the motion of nearby agents in the crowd. Once the agent model is learned, we can simulate a virtual crowd that behaves similarly to the real crowd in the video. The versatility and flexibility of our approach is demonstrated through examples in which various characteristics of group behaviors are captured and reproduced in simulated crowds.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Three-Dimensional Graphics and Realism]: Animation

1. Introduction

The group behavior of human crowds in the real world varies significantly depending on time, place, the level of stress, the age of people, and many other social and psychological factors. These variations in group behavior are often characterized by observable features such as interpersonal space, the fluidity of formation, the level of energy, the uniformity of distribution, the style of interpersonal interactions, and so on.

The most popular approach to crowd simulation is based on agent models. Each individual agent perceives nearby environments and other agents, and decides its actions based on a set of simple rules with a few tunable parameters. A big advantage of agent-based models is that the integration of individual rule-based actions exhibits collective group behavior of enormous complexity and subtlety. The downside to this approach is that the crowd behavior is affected indirectly and unintuitively by tuning parameters and changing rules. This makes it difficult to produce a desired style of group behavior using agent-based models.

† e-mail: {zoi,mingle,mani83,jehee}@mrl.snu.ac.kr

SCA 2007, San Diego, California, August 04 - 05, 2007
© 2007 ACM 978-1-59593-624-4/07/0008 $ 5.00

We present a data-driven method of simulating a crowd of virtual humans that exhibit behavior imitating real human crowds. Our simulation method is also agent-based, but the behavior patterns of each individual agent are learned (rather than hard-coded) from a video clip that recorded a crowd of people from an aerial view. From the video, we would like to understand what each person perceived and how he/she acted in the perceived situation at every time instance. To do so, we track the two-dimensional moving trajectories of people and extract the context of the motion using vision-based tracking techniques. As a result of visual tracking, we acquire a large collection of state-action pairs, from which our virtual human learns which actions to take for any given situation.

Learning group behavior from videos is a challenging problem because the motion of each individual is influenced by many factors that are not captured in the video. We assume that each person in the video has separate mechanisms for deciding his/her high-level behavior and low-level motion. The low-level motion is assumed to be controlled based only on features of the environment and the motion of nearby individuals in the crowd. In this work, we focus on learning an agent model that controls the low-level motion of each agent

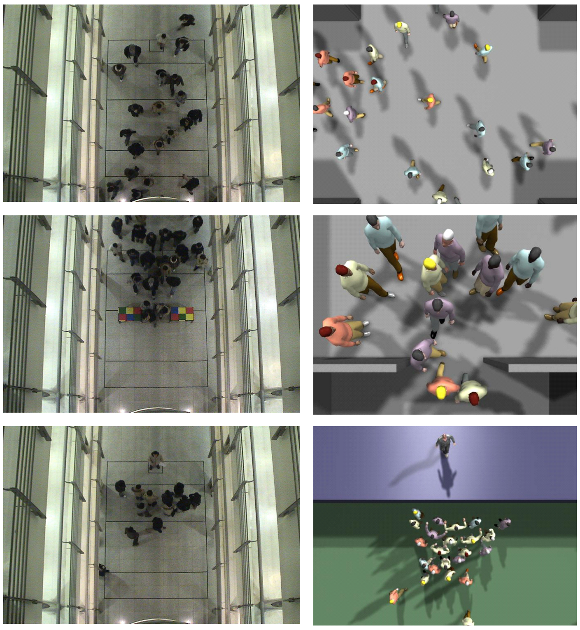

Figure 1: *Group behavior models are learned from crowd videos.*

in the crowd. Our learning algorithm is based on a locally weighted linear regression [AMS97], which is further elaborated to cope with complex group behaviors in a variety of environments.

Although multiple object tracking has been extensively studied in computer vision, automatic tracking of densely populated pedestrians is still challenging in practice. Instead, our system offers a semi-automated tracker and a keyframe-based interactive interface to users for allowing user intervention. Most of vision-based tracking algorithms track targets forward in time starting from their initial locations. This approach often suffers from the "hijacking" problem, that is, a tracker missing its own target and being attracted by another target. Our user interface system allows the user to refine the (possibly flawed) tracking result by inserting keyframes incrementally in a top-down manner.

We demonstrate the power of our approach through examples, in which subtle characteristics of captured group behaviors are reproduced, and through comparison with Reynolds' flocking algorithm. Our data-driven behavior model can be learned to imitate the rule-based flocking algorithm. We also demonstrate that captured insect behaviors can be adapted for simulating human crowds.

2. Related Work

Crowd simulation and group behavior control have gained significant interests in computer graphics, robotics, and urban planning. Since the rule-based agent model was introduced in the seminal work of Reynolds [Rey87], many variants of agent models have been explored in research community and employed in commercial products such as Massive [Mas06]. A number of researchers addressed the problem of creating a "smarter" agent that can decide its actions based on social, psychological, cognitive reasoning [MT97, POSB05, FTT99, MH04, ST05], and have pathfinding capability facilitated by global path planning methods [LC03, BLA03, LD04, PdHCM*06, GKM*01, KO04, SKG05]. Some researchers focused on an alternative approach that makes the environment "smarter" by embedding a repertoire of control modules into environment objects so that an agent can be provided with control strategies appropriate to its location and situation in the environment [SGC04, ACT05, LCL06]. We adopt this approach to allow the user to construct high-level group behavior scenarios flexibly by arranging a set of learned behavior models in the environment.

Crowd modeling has also been studied in the urban planning and fluid dynamics communities. Helbing and his colleagues [HMFB01] suggested a social force model of individual pedestrian dynamics. Kirchner and Schadschneider [KS02] simulated the evacuation process of a human crowd in a panic situation using cellular automata. Hughes [Hug03] formulated the behavior of a human crowd as a flowing continuum. Treuille et al. [TCP06] further elaborated Hughes' continuum crowd model and suggested a numerical method for simulating and visualizing crowd flow. Chenney [Che04] explored a way of designing stationary flow fields by tiling small rectangular regions of velocity fileds.

The computer vision community has developed a number of algorithms to track multiple moving objects in a video. To name only a few of them, Comaniciu et al. [CRM03] presented a kernel-based object tracking method that represents targets by feature histograms and tracks the targets using gradient-based optimization. Zhao and Nevatia [ZN04] addressed tracking multiple humans and inferring their coarse three-dimensional postures from a prior locomotion model. Khan et al. [KBD05] proposed a MCMC-based multi-target tracker that addressed a variable number of interacting targets. Egerstedt et al. [EBD*05] applied this MCMC-based tracker to trace the moving trajectories of ants and built a model that informs what the ants were doing in the video.

Reproducing observed movement patterns in a simulated environment has been explored in computer animation, roboics, and pedestrian research. Lai et al. [LCF05] presented a data-driven method of creating and dynamically updating group formations from a set of formation patterns. Brogan and Johnson [BJ03] and Fajen et al. [FWTK] showed that human walking data can be used to improve the quality of planned paths by adapting their steering models to the observed data. In pedestrian research community, several researchers addressed the problem of reconstructing crowd phenomena from video by tuning the parameters of their microscopic pedestrian models [Sti00, Tek02]. We share the basic motivation with these approaches. The main advantage of our approach is the ability to imitate a wide range of group behavior patterns and styles by using a regression-based learning method.

3. Video Capture and Processing

Human crowds are ubiquitous. Interesting crowd behaviors can be found in any downtown streets and buildings. We installed a camcorder above the crowds as high up as possible to observe them from an aerial view with minimal parallax effects, as long as each individual in the crowds is recognizable in the video. This setup makes the tracking procedure less painful because of minimal overlap of people. We also recorded crowd videos in a controlled environment. In recorded videos, about 40 volunteers displayed a variety of group behavior scenarios as instructed. The scenarios include pedestrians crossing and streaming through a variety of environment features, lining up in front of a ticket booth, watching a performance at streets, wandering around, chatting in small groups, just standing idly, and so on.

The crowd video includes a lot of information. From the video, we need to identify environment features (such as walls and obstacles), track two-dimensional trajectories of individuals, and understand their high-level behaviors. At a preprocessing phase, per-pixel median filtering is applied to the video for background subtraction. Then, the user annotates environment features manually on a video frame. Since our videos do not include any dynamically moving objects other than people, the manual annotation is not a burden.

Tracking individual trajectories from the video is a notoriously difficult problem. Inspired by the keyframe-based tracking interface of Agarwala et al. [AHSS04], we built a simple keyframe-based user interface system that allows the user to track multiple targets in the video semi-automatically. Our system employs a kernel-based tracking algorithm proposed by Comaniciu et al. [CRM03] for tracking targets. This algorithm works well in a short term duration, but often fails for an extended duration of tracking. In order to achieve better reliability, our system requires the user to specify both initial and final locations of each individual in the video. Then, the tracker is used to trace the target trajectory forward in time from the initial location and also trace backward in time from the final location. The forward trajectory and the backward trajectory are linearly blended to produce a more reliable trajectory. If this trajectory is unsatisfactory, the user may browse video frames and specify the location of the target at an intermediate video frame. The bi-directional tracking is then applied to two separate (initial-to-intermediate and intermediate-to-final) time intervals. In this way, the user can refine the trajectory adaptively until a satisfactory result is obtained. Additionally, Gaussian filtering is applied to resultant trajectories in order to smooth out noisy oscillations.

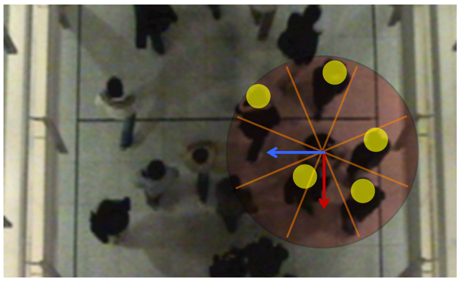

Figure 2: *The neighborhood formation is encoded with respect to the local coordinate system shown in red and blue arrows. The subjects were instructed to pretend that they are walking along corridors.*

Understanding high-level behaviors of individuals is even more challenging. At a ticket booth, for instance, some people would wait in line while some others would pass by or wander around. Although we have not yet found a general method of classifying different high-level behaviors automatically from the video, we found that a set of simple rules can be applied to identify specific behaviors effectively. For example, the moving speed and relative distance between nearby agents can be used to differentiate individuals waiting in line, passing by, and just wandering around. In our experiments, we manually annotated individual trajectories with the type of high-level behaviors facilitated by these heuristic rules.

4. State-Action Trajectories

Our crowd simulation system creates a number of agents that decide actions based on their limited perception capability. The behavior model of each agent is learned from state-action samples $\{(\mathbf{s}_i, \mathbf{a}_i)\}$ obtained from crowd videos. The state $\mathbf{s} \in \mathbb{R}^N$ of each agent reflects the motion of nearby agents, its own motion, and environment features. The action $\mathbf{a} \in \mathbb{R}^2$ of each agent is simply a two-dimensional vector that corresponds to the instantaneous speed and moving direction of the agent. To learn behaviors effectively, it is very important to have a low-dimensional state space, in which various aspects of group behaviors are compactly characterized. We selected four features to model the agent's state vector:

- **Self speed:** The instantaneous speed of the agent is computed as the distance between the current position of the agent and its position at the previous time instance divided by the time interval.
- **Neighborhood formation:** The motion of the agent is greatly influenced by the formation of nearby agents in the crowd. Closer agents would have stronger influence on the

behavior model and agents farther than a user-provided threshold distance r have no influence. The threshold distance r depends on the density of the crowd. To effectively represent this information by a fixed and relatively low dimensional vector, we divide the space around the agent into a set of eight radial regions (see Figure 2). Within each radial region, the distance to the nearest agent is a determining factor. The feature value for each radial region is

$$f = \begin{cases} (r-d)^2/r^2, & \text{if } d < r \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where d is the distance to the nearest agent in the radial region. In order to reflect the temporal change of the neighborhood formation, the state vector includes feature values from two successive (the current and the previous) frames. Thus, the neighborhood formation takes 16 entries in the state vector.

- **Pivot:** Behaviors, such as passing through gates and waiting in line, would take place with respect to the relevant objects. We assume that each behavior we intend to model can have at most one pivot object. In that case, the state vector includes the position and orientation of the agent with respect to the pivot object and its neighborhood formation is also represented with respect to the coordinate system attached to the pivot object.
- **Intended moving direction:** Each individual in a human crowd behaves with intention, which is an important factor of understanding individual behaviors. The intention is often implicit and may not be captured in crowd videos. We approximately estimate the intended moving direction of a pedestrian by averaging the moving directions in a window of past and future frames. The window size was ten frames (one second) in our experiments. This estimation is rough, but still useful for high-level control of pedestrians.

Each feature is empirically weighed according to its relative influence on the group behaviors (see Table 1). When the crowd is dense, we weighs the neighborhood formation over the other features in order to avoid collisions between agents. The intended moving direction is the most important feature for locomotion.

The observed states are represented by 21-dimensional vectors. We use a principle component analysis (PCA) to reparameterize the state space in a low-dimensional (8 to 12 dimensional in our experiments) space for efficient learning of behavior models.

5. Group Behavior Model

We model the group behavior in the video with a two-level hierarchical structure consisting of high-level behavior models and low-level action models (see Figure 3 left). The behavior model consists of a set of low-level action models and decides the transitioning between them to reproduce globally plausible group behavior patterns in crowd videos. Each low-level model describes a primitive action that can be represented by a simple learning model.

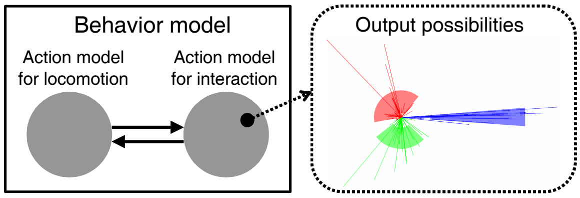

Figure 3: *Overview of our group behavior model. (Left) A high-level behavior model consists of low-level action models, depicted as the gray circles. At runtime, the behavior model decides the transitioning between low-level action models. (Right) For steering each agent, the corresponding active low-level model sorts output possibilities (shown as radial lines) into groups and selects a plausible output statistically. The translucent radial windows show the range of allowable perturbations.*

The state-action trajectories acquired from the processed video are segmented and classified into groups according to the annotated behavior patterns, and each group is used to learn a corresponding action model. The action model can be considered as a function that takes the state of an agent as input and produces a desired action of the agent at the next time instance. Given novel state s observed in simulation, our action model has to decide an action with respect to the training data. There are many approaches addressing this problem. For example, we can simply choose the nearest sample from the training set and select its associated action as output. This approach is simple and efficient, but the disadvantage is that the mapping between inputs and outputs are discontinuous. So, drastically different output actions can be produced for similar input states.

An alternative approach is to select nearby samples and combine them to produce an output. To do so, we employ a locally weighted linear regression method [AMS97], which requires that the model be *Markovian*. In other words, the agent's decision as to how to act should depend only on the current state. This condition is not satisfied in our group behavior learning problem. In crowd videos, each individual used to make different decisions at the same perceived state. We address this problem by clustering output possibilities and selecting a plausible output probabilistically (see Figure 3 right). We focus on the construction of action models in this section and discuss high-level behavior models in Section 6.1.

5.1. Neighborhood Search and Clustering

The basic idea of locally weighted regression is to search a small set of state-action samples that are similar to the current query state and estimate a linear regression model from the neighboring samples to determine an appropriate output action for the query state. The neighborhood search is the most computation-demanding component of locally weighted learning. In order to locate k-nearest neighbors efficiently, we store training data in a kd-tree and search k-nearest neighbors approximately within a small error bound using ANN library [MA06].

The neighboring samples thus obtained may represent a non-Markovian behavior that can produce more than one output possibility. In order to identify multiple output possibilities, we use k-means clustering algorithm, which sorts the neighboring samples into clusters at run time (see Figure 3 right). Each cluster includes samples with similar output actions. The number of clusters to be sorted out depends on the diversity of behaviors displayed in the video. The probability of selecting a cluster is inversely proportional to the mean deviations from the cluster selected at the previous time step. More specifically, the probability p_i of selecting the i-th cluster of mean $\mathbf{m}_i$ is

$$p_i = \frac{1/\|\mathbf{m}_i - \mathbf{m}_{prev}\|}{\sum_{j=1}^{K} 1/\|\mathbf{m}_j - \mathbf{m}_{prev}\|} \tag{2}$$

where $\mathbf{m}_{prev}$ is the mean vector of the previously selected cluster and K is the number of clusters.

5.2. Locally Weighted Linear Regression

Given samples $\{(\mathbf{s}_i, \mathbf{a}_i)\}_{i=1}^{k}$ where $\mathbf{s}_i$ is a vector-valued input state and $\mathbf{a}_i = (x_i, y_i)$ is a two-dimensional output, we first consider a regression model that produces the x-coordinates of output vectors. A regression model for y-coordinates can be built similarly. A linear fitting model can be described as a matrix equation $S\beta = \mathbf{x}$, where S is a matrix whose i-th row is $\mathbf{s}_i^T$, $\mathbf{x}$ is a vector whose i-th element is x_i, and β is a vector of the model parameters. Locally weighted linear regression estimates the model parameters

$$\beta = (S^T W S)^{-1} S^T W \mathbf{x}, \tag{3}$$

where W is a diagonal weight matrix with $W_{ii} = \exp\big(-\frac{1}{2\sigma}(\mathbf{s}-\mathbf{s}_i)^T(\mathbf{s}-\mathbf{s}_i)\big)$. This regression weighs near samples more than farther samples. Bandwidth σ determines how weights fall off with distance from $\mathbf{s}$.

5.3. Attraction

The strategy of reacting similarly at similar situations may fail if the situation encountered by an agent is significantly different from any state-action samples in the training data. In that case, the nearest samples are not really similar and a linear regression model would produce extremely extrapolated outputs. To avoid excessive extrapolations, we make a small perturbation to the output of locally weighted linear regression. The range of perturbation is limited by a constant multiple of standard deviations from the mean direction and distance (see Figure 3). Within the range, a perturbation is decided in such a way that the moving direction is steered toward the nearest sample configuration. To do so, we regularly sample output possibilities in the perturbation region and select the one that minimizes the distance to the nearest sample in the training data. Attraction to the nearest sample keeps the local formation of agents not to drift away from sample formations observed from the video.

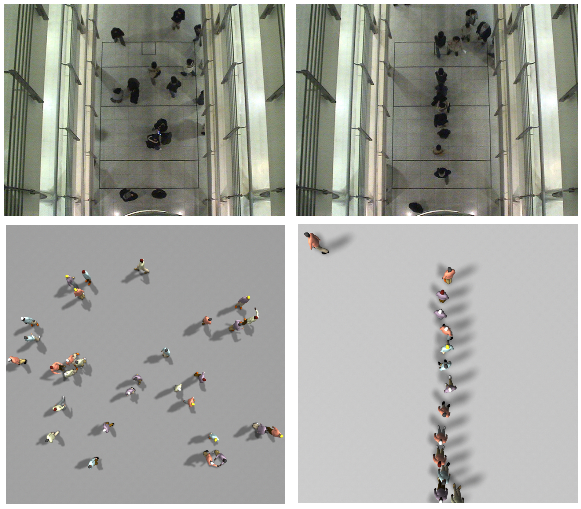

Figure 4: *Various interaction patterns in group behavior. (Top) We captured our subjects chatting in small groups and waiting in line. (Bottom) The learned action models produced the extended formations of corresponding styles.*

6. Crowd Simulation

6.1. High-Level Behavior

To make our agents not only react immediately with respect to perceived situations but also decide their high-level behaviors actively, we provide them with a high-level decision model composed of a set of learned low-level action models. In our system, this scheme is implemented as a finite state machine, in which each action model corresponds to a state. To reproduce global behavior patterns observed in training video, we encode the transition conditions between action models and control parameters such as the average duration of staying in each action model. These control parameters are measured from crowd video.

A typical example of high-level behavior is observed in a form of repeated transition between locomotion and group interaction. An agent moves around, joins a small group of agents for interaction, and leaves the group, repeatedly (see

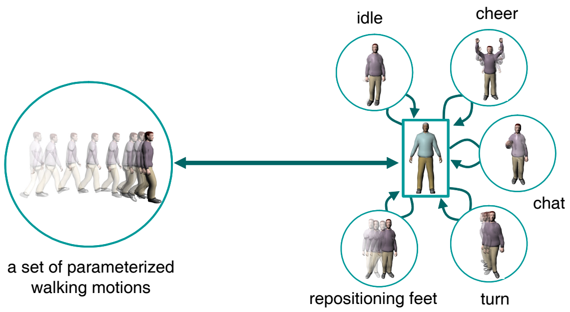

Figure 5: *A transition graph for full-body motion synthesis.*

Figure 4). These groups can be formed and disappeared dynamically. From the video, we measure the average size of agent groups, the average duration of staying, and the average interpersonal distance in groups. A new group is formed when more than two agents meet within the average interpersonal distance. If an agent approaches a group within its average interpersonal distance, the agent may be accepted to join the group probabilistically not to make the group too large beyond the average group size. Some action models enforce spatial constraints for transitioning. For example, the group of agents forms a line in front of a ticket booth. This group allows new members to join only at the end of the line and only the member at the front of the line can leave the group.

In order to simulate crowds in complex virtual environments, each agent need to adapt to their surroundings such as narrow gates, corridors and crosswalks. Our system allows the user to create the layout of an environment and annotate the environment what behavior models are appropriate in which part of the environment. Each agent in the crowd chooses its behavior model according to the annotation.

6.2. Full-body Motion Synthesis

For visually appealing and realistic crowd behaviors, we synthesize the full-body motion of human-like characters along the two-dimensional point trajectories of simulated agents. Since detailed full-body motions are not captured from the video, the context of motions along the trajectories are partly inferred from the trajectories and partly obtained from manual annotations.

We employ data-driven motion synthesis methods that utilize motion capture data. Two groups of motion data, locomotion and in-place motion, were collected to synthesize full-body animation of individual agents. In order to build locomotion library, our motion capture subject walked in various speeds and turning angles. The in-place motion library includes chatting, standing idly, cheering, turning, and repositioning feet.

The locomotion of a full-body character is parameterized by speeds and turning angles, and can be synthesized to walk along a two-dimensional trajectory. For parameterizing and blending example motions, we use an on-line motion blending method suggested by Park et al. [PSKS04]. This technique allows us to blend motion segments precisely at the granularity of frames. The in-place motions and the parameterized walking motions are integrated into a transition graph so that transitions between them can be made immediately (see Figure 5).

Synthesizing full-body motions along a given trajectory requires a lookahead search to determine which connecting transition to choose among many transition possibilities. When the computational cost is a major concern, a short period of lookahead time (0.66 seconds in our experiments) is used to decide transitions and the root node (the pelvis) of each articulated character is simply made to follow its target trajectory while allowing its feet to slide on the ground. In order to maximize the animation quality, each character searches a longer period of time to make a better decision of transitioning and prevents feet from sliding by using hierarchical displacement mapping [LS99].

7. Experimental Results

We recorded a variety of group behavior scenarios of about an hour in a rectangular region of 10m x 10m (see Table 1). In the videos, about forty volunteers walked in a variety of environment setups including corridors, corners and crossroads (Locomotion), passed through narrow gates (Passing gates), stopped and watched a street performance (Stage), waited in line (Line), chatted in small groups (Chat), and walked in various styles (Commuter, Stroller, Tourist). For each video, we tracked individual trajectories and annotated walls and pivots (such as the gate and the stage) if necessary. It took about thirty minutes to postprocess two hundred frames. In each individual frame, about twenty people are traced. The coordinate system was decided appropriately depending on the presence of pivots and the type of behaviors. For each query at runtime, we found one hundred of the nearest samples from training data and sorted them into three groups. The bandwidth of the kernel used in locally weighted linear regression was adaptively adjusted to the distance to the fifth nearest sample.

Styles. We acquired a collection of videos that recorded human crowds exhibiting different behavioral patterns, including the styles of busy commuters, relaxed strollers, and wandering tourists. Our behavior model captured these stylistic variations and reproduced them successfully in simulated crowds (see Figure 6).

Simulated crowds. To access the performance, we used our data-driven method to imitate Reynolds' rule-based algorithms. We collected a set of 4617 state-action samples from a simulated crowd controlled by Reynolds' leader-follower

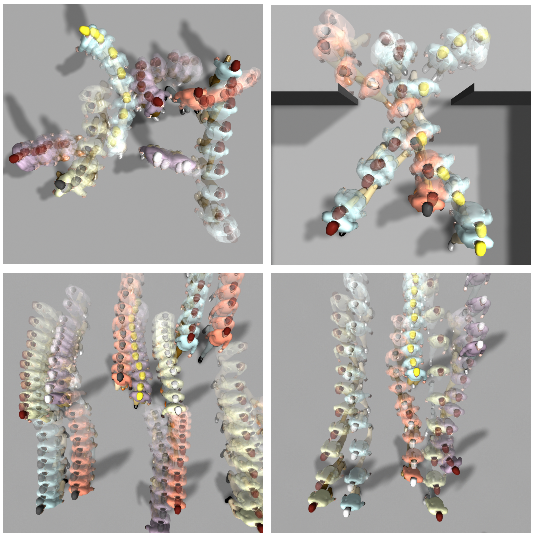

Figure 6: *The snapshots of crowd simulation reveal subtleties in group behaviors. (Top left) Various collision avoidance strategies are implicitly encoded in our data-driven behavior model and reproduced in the simulation. (Top right) Three characters meet at the gate. A similar situation observed in the video is reproduced so that the character at the middle gives way to the other two and waited until they pass through the gate. (Bottom) The stylistic variations in group behavior are reproduced in the simulated crowds. The agents in the left figure tend to keep pace with nearby agents whereas the agents in the right try to pass the others at variable speed.*

Figure 7: *The locomotion and interaction behavior models are learned from the ants video on the left and these behavior models are used to simulate virtual human crowds on the right. The ants video used courtesy of BioTracking project at Georgia Institute of Technology.*

model. Our behavior model learned from simulated training data similarly reproduced the characteristics of the rule-based behavior.

Ants. The group behaviors of animals and insects can also be applied to humans, thus creating a virtual human crowd that exhibit their behaviors. In the ants video, twenty ants are roaming around, approaching other ants, and interacting with each other by antennal contact. We learned an agent model from the ants video and applied this model to simulate a virtual human crowd (see Figure 7). To do so, we classified observed ant states into either locomotion or interaction automatically using simple rules; A group of ants staying within a threshold distance for more than one second are considered to be interacting with each other. All the other behaviors are classified as locomotion. The size and the velocity of ant trajectories were scaled appropriately in order to compensate for the scale difference between humans and ants.

Evacuation. Our approach is useful in simulating an evacuation scenario, in which a realistic group behavior plays an important role. In Figure 8, 80 agents are initially scattered in the five rooms, which are connected to the corridors leading to the building exit. Appropriate behavior models were annotated on the environment layout in such a way that the agents can leave the room and rush to the exit for evacuation. Similar to real emergency situations, bottlenecks were formed at narrow gates and junctions in simulation.

Small town performance. We built a small virtual town populated by 300 residents (see Figure 9). In a public square in the town, break dancers show their performance. The resident people are wandering around on the streets, chatting, gathering to watch the performance, and cheering. Generating 1000 frames of crowd simulation took about 5 minutes with 3D rendering disabled.

8. Discussion

The primary advantage of our approach is its capability of reproducing realistic human group behaviors in simulated environments. In our experiments, we observed the complexity and subtle details of simulated behavioral patterns that cannot easily be accomplished by using rule-based agent models. Our approach is especially useful for applications that need to simulate a large repertoire of realistic interactions among human characters.

The memory and computation costs of our regression-based learning algorithm increase with the amount of training data. The memory cost is not generally a problem, because it increases linearly with the size of training data. The computation cost is more serious, because the controller performs a neighborhood search for regression at run time. The computational cost was mitigated by maintaining data in *k*d-trees, which facilitate efficient spatial query processing.

We have visually compared the original videos and simulated crowds to see if human group behaviors are successfully captured and reproduced in the simulated environment. Though this visual comparison is an effective way of evaluating the quality and similarity of group behaviors in a subjective point of view, we also need a quantitative method of evaluating and characterizing group behaviors. The statistical measures, such as the average interpersonal space,

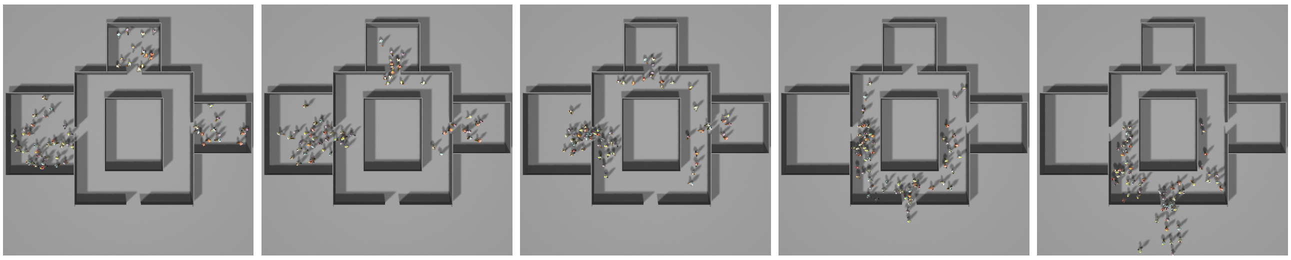

Figure 8: *Evacuation*

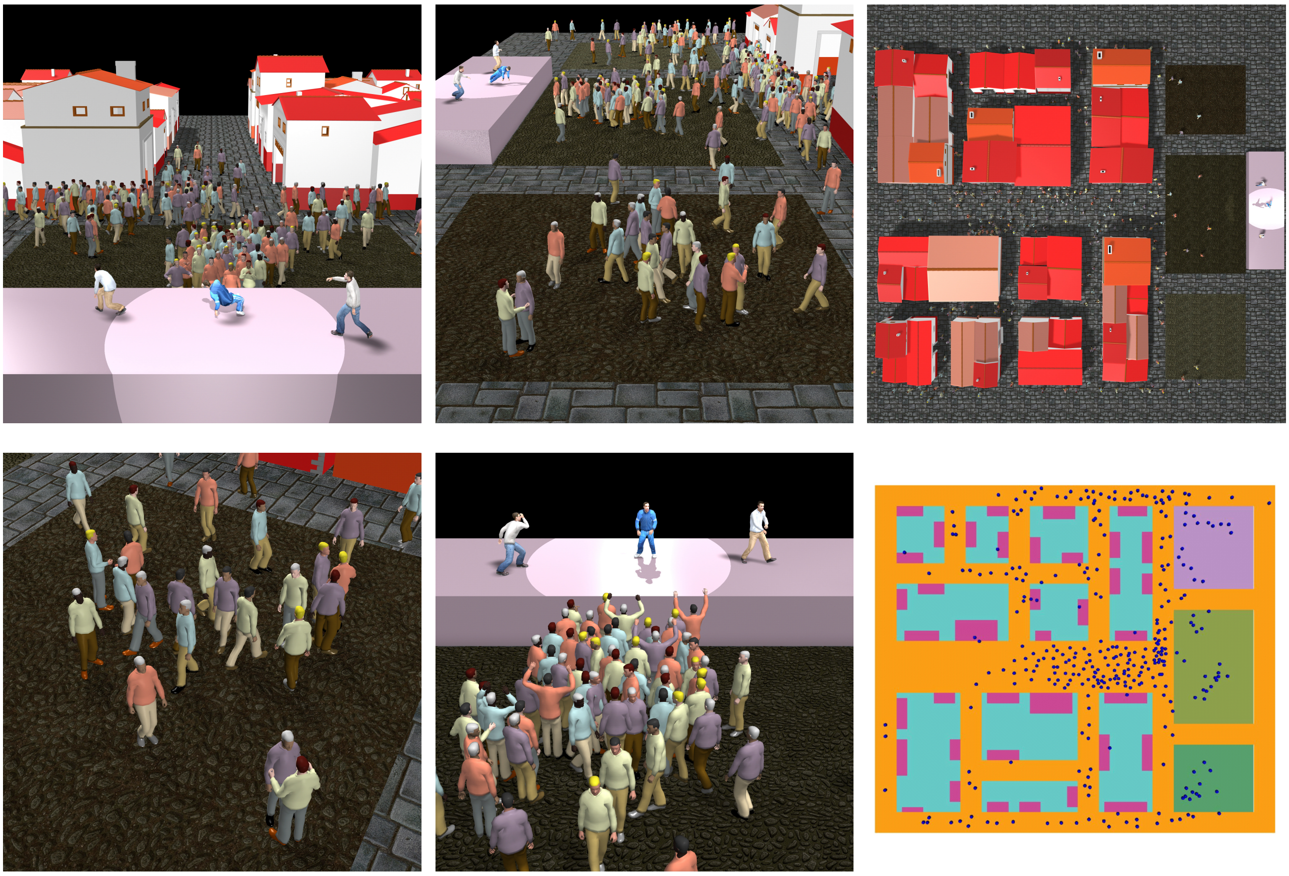

Figure 9: *Break dancer performance in the small town. The bottom right image is the layout of the town annotated by embedded behavior models. The characters in the blue region is directed to move to the gates and the characters in the pink region is provided with the "Passing gate" behavior model. The green, light green, purple annotations correspond to "chat", "ants", and "stage" behavior models, respectively. Building geometry models used courtesy of Pascal Mueller and Simon Haegler, copyright ETH Zurich.*

Behavior Models	# of video frames	# of (s,a) samples	Coordinate system	Neighborhood radius	Feature weights			
					Speed	Formation	Pivot	Direction
Locomotion	3000	2204	local	1.5 to 2 (m)	1	100	0	1
Passing gate	400	1216	pivot	1 (m)	1	100	10	0
Chat	1200	1041	pivot	1 (m)	1	100	0	0
Line	1200	927	pivot	1 (m)	1	10	10	0
Stage	600	640	pivot	1 (m)	1	100	1	0
Ants	3277	(interaction) 936	pivot	3 (cm)	1	100	0	0
		(locomotion) 1244	local	3 (cm)	1	10	0	100
Reynolds	10000	4617	local	2 (m)	1	1	500	0
Commuter	700	963	local	1 (m)	1	1	0	1
Stroller	700	1277	local	1 (m)	100	1	0	100
Tourist	1500	1226	local	1 (m)	1	50	0	1

Table 1: *Experimental data. The videos are sampled at the rate of 10 frames per second. The state of each individual is represented with respect to either a local, moving coordinate system or a fixed coordinate system. The radius of a perceived neighborhood was determined based on the density of the crowd. The feature weights for self speed, neighborhood formation, pivot, and intended moving direction are provided.*

the density and uniformity of distribution, the regularity of formation, and the temporal and spatial coherence of group interactions, might be used to quantify the characteristics of group behaviors.

The flexibility of crowd simulation could be increased by parameterizing group behaviors learned from video. A promising scenario is to install multiple camcorders at fixed locations for an extended period of time and collect video clips that record crowd behaviors at a variety of time, seasons, weather conditions, and spatial locations. These collection would allow us to build a parameterized behavior model that can be adapt for environment factors.

Acknowledgements

We would like to thank the Georgia Tech BORG laboratory for allowing us to experiment with the ants video and the trajectory data from the BioTracking project. We are also very grateful to Pascal Mueller and Simon Haegler for granting us to use their urban model. This research work was partly supported by the IITA under the ITRC support program.

References

[ACT05] ABACI T., CÍGER J., THALMANN D.: Planning with smart objects. In *WSCG (Short Papers)* (2005), pp. 25–28. 2

[AHSS04] AGARWALA A., HERTZMANN A., SALESIN D. H., SEITZ S. M.: Keyframe-based tracking for rotoscoping and animation. *ACM Transactions on Graphics (SIGGRAPH 2004) 23*, 3 (2004), 584–591. 3

[AMS97] ATKESON C. G., MOORE A. W., SCHAAL S.: Locally weighted learning. *Artificial Intelligence Review 11*, 1-5 (1997), 11–73. 2, 4

[BJ03] BROGAN D., JOHNSON N.: Realistic human walking paths. In *CASA '03: Proceedings of the 16th International Conference on Computer Animation and Social Agents* (2003), pp. 94–101. 2

[BLA03] BAYAZIT O. B., LIEN J.-M., AMATO N. M.: Better group behaviors in complex environments using global roadmaps. In *ICAL 2003: Proceedings of the eighth international conference on Artificial life* (2003), pp. 362–370. 2

[Che04] CHENNEY S.: Flow tiles. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), pp. 233–242. 2

[CRM03] COMMANCIU D., RAMESH V., MEER P.: Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Maching Intelligence 25*, 5 (2003), 564–577. 2, 3

[EBD*05] EGERSTEDT M., BALCH T., DELLAERT F., DELMOTTE F., KHAN Z.: What are the ants doing? vision-based tracking and reconstruction of control programs. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)* (2005). 2

[FTT99] FUNGE J., TU X., TERZOPOULOS D.: Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), pp. 29–38. 2

[FWTK] FAJEN B. R., WARREN W. H., TEMIZER S., KAELBLING L. P.: A dynamical model of visually-guided steering, obstacle avoidance, and route selection. *International Journal of Computer Vision*. 2

[GKM*01] GOLDENSTEIN S., KARAVELAS M., METAXAS D., GUIBAS L., AARON E., GOSWAMI

A.: Scalable behaviors for crowd simulation. *Computer & Graphics 25*, 6 (2001), 983–998. 2

[HMFB01] HELBING D., MOLNAR P., FARKAS I. J., BOLAY K.: Self-organizing pedestrian movement. *Environment and Planning B: Planning and Design 28* (2001), 361–383. 2

[Hug03] HUGHES R. L.: The flow of human crowds. *Annual Review of Fluid Mechanics 35* (2003), 169–182. 2

[KBD05] KHAN Z., BALCH T., DELLAERT F.: Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 11 (2005), 1805–1918. 2

[KO04] KAMPHUIS A., OVERMARS M. H.: Finding paths for coherent groups using clearance. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), pp. 233–242. 2

[KS02] KIRCHNER A., SCHADSCHNEIDER A.: Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A: Statistical Mechanics and its Applications 312(1–2)* (2002), 260–276. 2

[LC03] LI T.-Y., CHOU H.-C.: Motion planning for a crowd of robots. In *ICRA '03: Proceedings of IEEE international conference on Robotics and Automation* (2003), pp. 4215–4221. 2

[LCF05] LAI Y.-C., CHENNEY S., FAN S.: Group motion graphs. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), pp. 281–290. 2

[LCL06] LEE K. H., CHOI M. G., LEE J.: Motion patches: building blocks for virtual environments annotated with motion data. *ACM Transactions on Graphics (SIGGRAPH 2006) 25*, 3 (2006), 898–906. 2

[LD04] LAMARCHE F., DONIKIAN S.: Crowd of virtual humans: A new approach for real time navigation in complex and structured environments. *Computer Graphics Forum (Eurographics 2004) 23*, 3 (2004), 509–518. 2

[LS99] LEE J., SHIN S. Y.: A hierarchical approach to interactive motion editing for human-like figures. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), pp. 39–48. 6

[MA06] MOUNT D., ARYA S.: Ann: Library for approximate nearest neighbor searching, http://www.cs.sunysb.edu/ algorith/implement/ann/distrib/index1.html, 2006. 5

[Mas06] MASSIVE SOFTWARE: http://www.massivesoftware.com/, 2006. 2

[MH04] METOYER R. A., HODGINS J. K.: Reactive pedestrian path following from examples. *The Visual Computer 20*, 10 (2004), 635–649. 2

[MT97] MUSSE S. R., THALMANN D.: A model of human crowd behavior: Group inter-relationship and collision detection analysis. In *Computer Animation and Simulation '97* (1997), pp. 39–51. 2

[PdHCM*06] PETTRE J., DE HERAS CIECHOMSKI P., MAIM J., YERSIN B., LAUMOND J.-P., THALMANN D.: Real-time navigating crowds: scalable simulation and rendering. *Computer Animation and Virtual Worlds 17*, 3–4 (2006), 445–455. 2

[POSB05] PELECHANO N., O'BRIEN K., SILVERMAN B., BADLER N.: Crowd simulation incorporating agent psychological models, roles and communication. In *V-CROWDS '05: Proceedings of the First International Workshop on Crowd Simulation* (2005), pp. 24–25. 2

[PSKS04] PARK S. I., SHIN H. J., KIM T., SHIN S. Y.: On-line motion blending for real-time locomotion generation. *Computer Animation and Virtual Worlds 15*, 3 (2004), 125–138. 6

[Rey87] REYNOLDS C. W.: Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987), pp. 25–34. 2

[SGC04] SUNG M., GLEICHER M., CHENNEY S.: Scalable behaviors for crowd simulation. *Computer Graphics Forum (Eurographics 2004) 23*, 3 (2004), 519–528. 2

[SKG05] SUNG M., KOVAR L., GLEICHER M.: Fast and accurate goal-directed motion synthesis for crowds. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), pp. 291–300. 2

[ST05] SHAO W., TERZOPOULOS D.: Autonomous pedestrians. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2005), pp. 19–28. 2

[Sti00] STILL G. K.: *Crowd Dynamics*. PhD thesis, University of Warwik, 2000. 3

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

[Tek02] TEKNOMO K.: *Microscopic Pedestrian Flow Characteristics: Development of an Image Processing Data Collection and Simulation Model*. PhD thesis, Tohoku University, 2002. 3

[ZN04] ZHAO T., NEVATIA R.: Tracking multiple humans in crowded environment. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2* (2004), 406–413. 2