ACUMEN: Activity-Centric Crowd Authoring Using Influence Maps

Athanasios Krontiris
Rutgers, the State University of New Jersey
TDK.krontir@gmail.com

Kostas E. Bekris
Rutgers, the State University of New Jersey
kb572@cs.rutgers.edu

Mubbasir Kapadia
Rutgers, the State University of New Jersey
mk1353@rutgers.edu

ABSTRACT

Heterogeneity in virtual crowds is crucial for many applications, including visual effects, games, and security simulations. Nevertheless, tweaking the behavior parameters of a character to achieve crowd heterogeneity is frequently hard. In particular, it is typically unclear how tuning some non-intuitive parameters at the agent level will eventually affect both the microscopic or macroscopic scale of the crowd. This paper proposes an activity-centric framework for authoring functional, heterogeneous virtual crowds in semantically meaningful environments. The specification of locations as environmental attractors and agent desires are used to compute "influence maps", which allow the emergence of heterogeneous behaviors in a large virtual crowd in a complex scene. The same framework can also facilitate the authoring of complex group behaviors, such as following behaviors or families, by treating moving agents as attractors. Accompanying results demonstrate the framework's potential by authoring crowds in different environments. The experiments highlight the ability to easily orchestrate purposeful, heterogeneous crowd activities both at a macroscopic and microscopic level with minimal parameter tuning.

Keywords

Agent-based, Crowd Behavior, Influence Maps, Simulation, Crowd Simulation, Multi-user/multi-virtual-agent interaction, Simulation techniques tools and platforms

Figure 1: A heterogeneous virtual crowd simulated in a semantically meaningful environment using influence maps. Agents exhibit a variety of unique behaviors, subject to their own individual desires and the influence of their surroundings, and other agents.

1. INTRODUCTION

Modeling heterogeneous behavior for agents in a large crowd is important in various domains, including virtual environments, robotics, transportation engineering, psychology, visual effects, and games. The focus of this work is on crowd simulation systems, which can be used for studying humans in crowded situations, social behaviors, for architectural and urban design, or for training and emergency evacuation.

CASA 2016, May 23 - 25, 2016, ,

ISBN 978-1-4503-4745-7/16/05. . . $15.00

DOI: http://dx.doi.org/10.1145/2915926.2915935

Crowds are ubiquitous and a realistic simulation of crowds necessitates heterogeneity in the movement and behavioral characteristics of individuals and groups, which contributes to the properties of the crowd at an aggregate scale. The environment, context, and the demographics play equally important roles in influencing the crowd dynamics. It has also been reported that humans exhibit noticeably different behavior in crowded environments, where the presence of the crowd triggers temporary changes in personality of individuals. To this end, it is crucial to model the influence of the environment and other crowd members on an individual's behavior, in addition to an agent's own personal desires, as a governing factor to generate plausible, heterogeneous crowd simulations.

Current approaches [1, 2] rely on simple rules and non-determinism to inject variety in the visual appearance, movement, and behavior of crowds. Authors are tasked with tuning parameters, which often may not be intuitive, to achieve the requisite behavior at either the microscopic or macroscopic scale. Operating at the microscopic level allows for authoring precision while macroscopic control provides ease of specification. It is typically the case, however, that it is not possible to achieve both. An animator tasked with authoring large crowds is forced to tweak many different parameters that alter their choice of destinations, movement and navigation decisions, and how agents interact with each other and the environment, frequently in complex and non-intuitive ways. Scripting these parameters is tedious and specific to a particular environment and crowd configuration, where even minor changes in the scene mandate an overhaul in the specification. These challenges are further exacerbated when authoring unique reactions of individuals or groups as a reaction to dynamic events, such as the arrival of a celebrity.

This paper proposes an activity-centric framework for authoring functional and heterogeneous crowds in semantically meaningful environments. End users annotate environments with semantics, such as locations, which attract specific populations, as well as the desires of each population. Environment attractors and agent desires are used to compute "influence maps", which encode the relative influence of an individual's surroundings on its behavior. Each agent elicits unique behavioral characteristics that satisfy its desires while accounting for the influence of the environment. Agents may influence each other, thus facilitating the authoring of complex groups behaviors, such as families, and crowds following a person of interest, or keeping safe distance from a suspicious individual. In particular, our work provides an accessible interface for authoring and synthesizing heterogeneous crowd activity.

Main Contributions. This paper provides the following main contributions: (1) An activity-centric approach for authoring heterogeneous crowd behaviors that decouples the specification of environment semantics (attractors) from the crowd definition (desires). (2) A computational model for encoding the influence of entities (environment locations or other members in the crowd) on an agent, and its mapping to behavior selection, including grouping behavior.

We demonstrate the potential of the influence maps approach by authoring crowds at various locations, such as a shopping mall, a museum and an airport. This paper highlights the framework's ability in being able to easily orchestrate purposeful, heterogeneous crowd activities, with control at both the macroscopic and microscopic level, with minimal parameter tuning.

2. RELATED WORK

The maturity of research in simulating and authoring crowds has produced a wide variety of approaches that represent different tradeoffs between authoring precision and ease of specification. We refer the readers to comprehensive surveys [1, 2] and provide a brief overview below.

Simulation-centric Authoring. Several techniques have been proposed to simulate the low-level dynamics of crowd movement. These include particle dynamics [3], social forces [4], velocity-based approaches [5, 6], and continuum models [7, 8]. Commercial software, such as Massive [9] and Golaem [10], are *simulation-centric* where animators author the responses of an autonomous agent to external stimuli and tweak simulation parameters to mould the emergent crowd behavior to conform to the required specifications. This mode of authoring requires the animator to work within the limits imposed by the simulation framework. As a result, animators often resort to manual methods that provide more precise control, at the expense of significantly increasing the authoring burden.

Data-centric Authoring. The work in [11, 12] synthesizes synchronized multi-character motions and crowd animations, by performing editing and stitching operations on a library of motion capture data. Using this approach, the user can interactively manipulate the motions of many characters, while having precise control over individual trajectories. However, the approach is limited to the database of pre-recorded clips as large deformations and time warping may yield unnatural results. Also, editing an individual motion may change the entire crowd animation, which is undesirable when orchestrating crowd activities with multiple constraints. Motion patches [13] are environmental building blocks that are annotated with motion data, which informs what actions are available for animated characters within the block. Motion patches can be edited and connected together [14], or precomputed by expanding a search tree of single character motions [15] to synthesize complex multi-character interactions. This concept is extended in [16] by annotating spatial regions with agent trajectories to create precomputed crowd patches, which can be connected to author large crowds.

Behavior-centric Authoring. In contrast, behavior-centric approaches use logical constructs [17] and complex models [18] to represent knowledge and action selection in agents. Parameterized Behavior Trees (PBT's) [19] are hierarchical, modular descriptions of coordinated activities between multiple actors. Smart Events [17] externalize behavior logic to authored events that occur in the environment. Improv [20] and LIVE [21] describe behaviors as rules, which govern how actors act based on certain conditions. These systems are reactive in nature, and typically produce predefined behaviors corresponding to the current situation. Cognitive approaches [18] use complex models, such as decision and neural networks to model knowledge and action selection in virtual agents. Normoyle et al. [22] proposed an agent decision model using a stochastic graph and a series of convex optimization problems in order to compute the parameters for the heterogeneous crowd, which is difficult to author. The use of domain-independent planners [23] is a promising direction for automated behavior generation, which provide automation while sacrificing authoring control.

Influence Maps. The concept has been used to model agent-based decision-making in computer games [24]. Svensson and Johansson [25] build an intelligent influence map-based controller for controlling "Ms. PacMan". The current framework is focused on computing the next desired location for each individual agent in the crowd by using the influence that the agent receives from the environment. As a result, each agent will decide to go to different location based on its own desires. Similarly, the robotics community have developed *potential fields* for crowd specification [4, 26], which encode attractors and repellers in the environment to model goal-directed collision avoidance. The notion of influences has also been defined for agents as well. The work in [27, 28] define proxy agents which attract or repel other agents to model group behaviors.

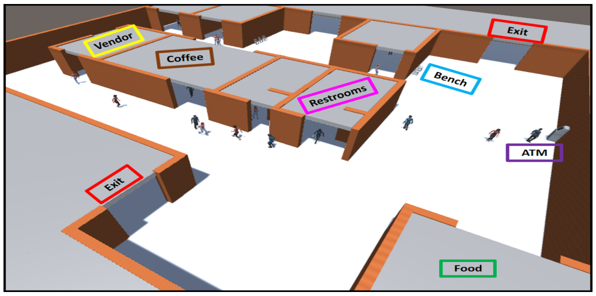

Figure 2: Semantically annotated environment, where locations are agent attractors.

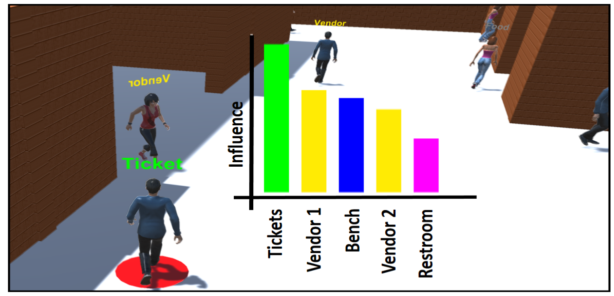

Figure 3: Influences of environment attractors on an agent.

3. INFLUENCE MAPS FOR CROWDS

Our framework establishes a relation between an environment and its inhabitants using *influences* which characterizes the degree to which a specific entity (an environment location or another agent) attracts an individual in a crowd. This relation is represented and computed using "influence maps", a well-known concept in the agent community that we are leveraging them for our purpose [25, 29, 30]. Agents exhibit behavior that is affected by the influences of their surroundings, subject to their own unique desires. An influence map for a specific agent represents the relative influences of the surrounding environment locations, other crowd members, and the agents final destination. A selection function queries the influence map of an agent to determine the agents next behavior choice, by choosing the next location to go, a crowd member to follow (or stay away from), or make its way to its final desired destination.

Figure 2 depicts potential locations that could be used as attractors for an agent inside a mall. Given these influence sources, an agent will select the location that exerts the greatest influence. Figure 3 shows an agent with its top five influences. Each bar represents a different attractor inside the environment, while the color of the bar shows the type of the attractor. Influences due to other agents are similarly modelled and explained below. Our framework is computationally efficient and influence sources may be added or removed on the fly, to facilitate dynamic environments that produce emergent crowd dynamics.

3.1 Preliminaries

Attractors are entities that exhibit an influence on an agent in the crowd, and maybe specific environment locations or other agents.

Environmental Attractor Type. The set $\mathbf{R}$ defines the different types of environmental attractors. Such types $r \in \mathbf{R}$ can be specified by the author depending on the environment. The desire of an agent to follow a group or head to one's final destination are always included as available environmental attractor types.

Agents. Assume a crowd of agents $\mathbf{A}$ is simulated inside a facility. Each agent $\alpha \in \mathbf{A}$ is defined as

$$\alpha = \langle \mathbf{p}_\alpha, \mathbf{p}_\alpha^d, t^d, \mathbf{W}, \mathcal{M} \rangle,$$

where $\mathbf{p}_\alpha$ is the current position of the agent, while $\mathbf{p}_\alpha^d$ specifies the final target that the agent has to be before the time t^d. $\mathbf{W}$ is a set of desires that the agent has for the different types of environmental attractors, $\mathbf{W} = \{w^r | \forall r \in \mathbf{R}$, s.t. $0 \leq w^r \leq 1\}$. Finally, $\mathcal{M}$ is the set of influence values that are computed online by all environmental attractors.

Locations. A location $l \in \mathbf{L}$ is the physical location of an attractor. It is defined as

$$l = \langle r, \mathbf{p}_l, o, o^{max}, t^{min}, t^{max} \rangle,$$

where r specifies the type of attractor this location is associated with, $\mathbf{p}_l$ is the position of this location, o specifies its current occupancy, while o^{max} provides the maximum capacity of this location. Each agent will spend some time $t^l \in [t^{min}, t^{max}]$ inside each location.

Groups. A group $\mathbf{g} \in \mathbf{G}$ specifies a set of agents ($\mathbf{g} \subseteq \mathbf{A}$), who will move together in the facility. In this way, each agent $\alpha_i \in \mathbf{g}$ becomes an attractor for the other group members.

Influence Map. An influence map $\mathcal{M}_\alpha$ is the entire set of influences that an agent α receives from the environment. The influence map for an agent α is defined as:

$$\mathcal{M}_\alpha = \langle \{I_l\}, \{I_{\mathbf{g}}\}, I_d \rangle \tag{1}$$

Where I_l represents the influences that the agent receives from all the locations $l \in \mathbf{L}$ of the environment, $I_{\mathbf{g}}$ describes the set of influences from the group members $\mathbf{g}$, if the agent belongs to a group, and finally, the agent experiences an influence I_d from its destination. The next section describes how we compute these influences.

3.2 Target Selection via Influences

The agent has to compute the influence map, $\mathcal{M}_\alpha$ eq.1, which is a collection of all the influences that the agent gets from the environment, in order to decide its next target. An agent α has to compute influences from all the static attractors, $\{I_l(\alpha, l) | \forall l \in \mathbf{L}\}$. If the agent belongs to a group $\mathbf{g}$, then it will have to compute influences from the members of this group, $\{I_{\mathbf{g}}(\alpha_i) | \alpha_i \in \mathbf{g}$, s.t. $\mathbf{g} \in \mathbf{G} \cap \alpha \subset \mathbf{g}\}$. Finally, the agent has to compute the influence from the destination, I_d.

The next target location $\mathbf{p}_\alpha^t$ for the agent α is the location of the attractor (environment location, group, or the agent's final destination) that currently has maximum influence over the influence map, $\max(\mathcal{M}_\alpha)$.It is possible for an agent to select $\mathbf{p}_\alpha^d$ and leave earlier than the specified t_α^d This could happen because all the attractors have small influence, as a result, the agent decides to leave the environment. Note that influence calculations and target selection is performed regularly (usually amortized over a few frames), and it may be possible for the maximal attractor to change as the agent is traveling to its current destination.

3.3 Location-Based Influences

Static locations in an environment attract agents based on three metrics: (1) the desire that an agent has for this region, (2) the occupancy level of the location, and (3) the time that the agent has before departing the environment.

Agent Desire: An agent's desire to visit different environment attractors is specified as $w^r_\alpha \in \mathbf{W}_\alpha$, where each location l in the environment is associated with a specific type r of environment attraction, and has a multiplicative effect on that location's influence. Agent desires embed unique traits in different crowd members, allowing authors to specify different crowd demographics, each with their own unique set of desires. Note that authors may choose to specify the desires of each individual in a crowd separately for extremely fine-grained control, or cluster crowd members together into categories of people, for ease of authoring.

Occupancy: The current occupancy o_l of a location l has a negative impact on its influence over an agent, which decreases linearly, based on the following relationship:

$$I_o(l) = \begin{cases} 1 - \frac{o_l}{o_l^{max}} & \textbf{if } o_l \le o_l^{max}; \\ -\infty & \textbf{otherwise} \end{cases} \qquad (2)$$

Time: Each agent has a specific amount of time t to spend inside the facility before the agent has to reach its final destination $\mathbf{p}^d_\alpha$ at time t^d_α. All the agents have to respect this rule and should not visit a location where they will have to spend more than their available time. The time that an agent α needs for a location l is computed as the sum of the following three terms: (1) $\mathtt{time}(\mathbf{p}_\alpha, \mathbf{p}_1)$, the time the agent needs to reach l from its current position, (2) $\mathtt{time}(\mathbf{p}_1, \mathbf{p}^d_\alpha)$, the time α needs to go from l to $\mathbf{p}^d_\alpha$, and (3) t^l_α, the time the agent will spend inside the location l.

In order to estimate time to travel between two locations $\mathtt{time}(\mathbf{p}_1, \mathbf{p}_2)$, we compute the path integral along the static optimal path between $\mathbf{p}_1$, $\mathbf{p}_2$ (computed using A* on a navigation mesh), and assume the agent travels along this path. The total time t_{need} needed by an agent α is thus computed as follows:

$$t_{need} = \mathtt{time}(\mathbf{p}_\alpha, \mathbf{p}_1) + \mathtt{t}^1_\alpha + \mathtt{time}(\mathbf{p}_1, \mathbf{p}^d_\alpha)$$

The time that the agent will have inside the environment after visiting the location l will be: $t = t^d_\alpha - (t_{curr} + t_{need})$. If $t < 0$, then the influence is negative and the agent will definitely avoid visiting this location. Otherwise, the influence is computed as follows:

$$I_l(\alpha, l) = \frac{t \cdot w^r_\alpha}{t+1} \cdot I_o(l). \qquad (3)$$

3.4 Group-Based Influences

Agents in a group $\mathbf{g}$ influence each other, and serve as moving attractors. An agent may be in multiple groups. The influence that an agent $\alpha_i \in \mathbf{g}$ applies on an agent α is computed by:

$$I_{\mathbf{g}}(\alpha) = w^{\mathbf{g}}_\alpha \cdot I_l(\alpha, l_i) \qquad (4)$$

where $w^{\mathbf{g}}_\alpha \in \mathbf{W}_\alpha$ is the desire of the agent α to follow the group $\mathbf{g}$, and $I_l(\alpha_i, l_i)$ computes the location-based influence that agent α gets from the location l_i that agent α_i is currently heading to. As a result, if the current destination of α_i is not attractive to α, then the influence from α_i will not be strong.

3.5 Destination Influence

Each agent aims to leave from the environment at time t^d_α. The desire to depart the environment increases as the available time to departure t decreases. As mentioned before, the destination is treated as an attractor, where each agent has a desire $w^d_\alpha \in \mathbf{W}_\alpha$ and the location will apply influence on the agent equals to:

$$I_d(\alpha) = \begin{cases} \infty & \textbf{if } t \le 0; \\ \frac{w^d_\alpha}{t} & \textbf{otherwise} \end{cases} \qquad (5)$$

where $t = t^d_\alpha - (t_{curr} + \mathtt{time}(\mathbf{p}_\alpha, \mathbf{p}^d_\alpha))$. If $t \le 0$, then the agent is already late and has to run towards $\mathbf{p}^d_\alpha$.

3.6 Dynamic Influences

The current framework allows the user to dynamically add new influence sources in the environment. In this way, the user can interact in real-time with the environment and change the decisions of the agents. The tool supports the interactive addition and modification of both location-based and group-based influences. For example, a new agent could appear in the scene (e.g., a celebrity), who may exude a strong influence to most of the other agents in the scene. As a result, most of them desire to follow this new agent. In addition, the user could add or remove an attractor from the environment. The agents will react to this change without any further involvement from the user. In case a new location with influence source appears, the agents will start getting influenced by this location. Removing one influence source the user makes the agents indifferent to this location.

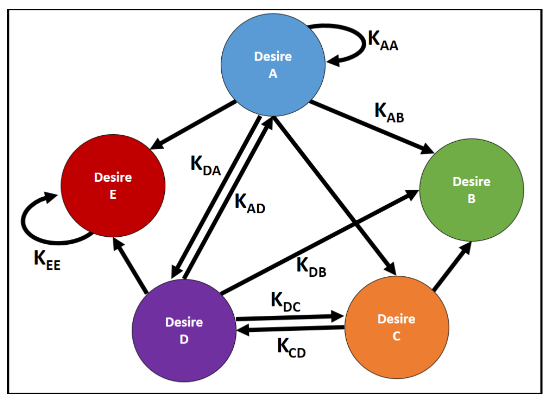

Figure 4: Dynamic dependencies between desires are modeled with a directed graph. Edges correspond to how a visit to an attractor of type A will effect the agent's desires.

In addition, the user can tune both the desires that effect the exo-centric influences and the desires that are used for the ego-centric influences on-the-fly. This can happen either through an authoring interface where the end user can change the desires of the agent as the user wishes, or through parameters that the user specifies. These parameters affect how much using a type of attraction affects the rest of the desires of the agent (see figure 4). The K_{ij} values show the rate of change that one type of attraction will change the other desires of the agent. These values can be either positive or negative. The new value for the desire j, after the agent visited an attractor of type i, will change based on the function:

$$\mathtt{W(j)} = \mathtt{W(j)} + \mathtt{K_{ij}} \cdot \mathtt{W(j)} \tag{6}$$

The end user can use this tool in order to control the order of attractions that the agent will visit. For example, after the agent visits a food store there is large probability that he would like to visit the restroom. Then the multiplier for using the food store and then restroom could be high enough to dominate all the other desires. Moreover, the end-user can reduce the desire for an influence type after the agent uses it once, and as a result, the desire to use the same location again or a location of the same influence type will be reduced.

4. AUTHORING PROCESS

Authoring a scenario entails three modular and independent steps: (1) Environment Semantic Specification, (2) Crowd Demographic Specification, and (3) Group Specification. A key benefit of our approach is that the environment and the crowd can be defined independently of each other, allowing authors to easily compare and contrast different crowd behaviors across environments.

Environment Semantic Specification. Content creators such as game level designers or artists define rich, complex virtual worlds and embed it with environment semantics which annotate the manner in which different environment locations may attract different subsets of the crowd. This is specified by associating a specific location l in the environment with a corresponding environment attractor label r. Optionally, the degree of influence of a specific environment attractor may also be specified. Each attractor has an influence type, its own maximum capacity and an interval of time a person will spend inside the attractor (fig.5). Authors can use our graphical user interface to easily add, remove or edit environment elements on the fly.

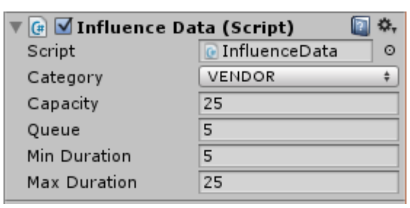

Figure 5: Attractor Specification.

Crowd Demographic Specification. The author first specifies different agent categories, each with their own unique set of desires, modelled as weights $\mathbf{W}_\alpha$, and desired final destinations. A specific crowd instance comprises a relative different distribution of the different categories of agents, to produce a varied crowd, that exhibits different behavioral characteristics based on the influence of its surroundings and its own individual desires. Note that an author may work at different levels of granularity depending on application needs, by either specifying unique desires for each individual in a crowd, or creating coarse categories of similar agents.

Group Specification. Authors may additionally create agent groups where members of the group influence each other. If an agent is in a group then a desire value $w_\alpha^{\mathbf{g}}$ defines the desire of the agent to follow its group members. For example, the author could model a family using the group behavior. The members of the family will belong to the same group and in order the members to stick together the $w_\alpha^{\mathbf{g}}$ value should be high. The leader of the group will have smaller desire to stay with the group, as a result, the leader will select to visit a different attractor and the rest of the member will follow this agent. A behaviour of a crowd following a guide at a museum can be modelled similarly. All the agents that desire to follow the guide will be in the same group with the guide and their desire to stay with the group will supersede other desires.

Specification Benefits. The proposed specification interface provides a minimal, yet sufficient abstraction to empower end users to efficiently author heterogeneous crowds. This is in contrast to existing solutions which require authors to manually specify intermediate destinations for each agent in the simulation. Additionally, prior methods do not have any means to enforce constraints such as the maximum occupancy of buildings, which is automatically satisfied using our method. Our approach also facilitates modular and iterative authoring of crowds and environments. End users can incrementally increase the complexity of crowds and environments by adding new attractors and desires without having to re-specify the behaviors of previously defined agent behaviors.

5. APPLICATIONS

We demonstrate the efficacy of our approach by authoring functional, purposeful crowds in a shopping mall, airport, and art museum. The examples presented in this paper were created with the Unity Game Engine [31] and show how the proposed technique can be applied to a variety of crowd simulation scenarios. Experiment details of all the environments are detailed below. Animations of the resulting experiments can be seen in the accompanying video.

5.1 Mall

This example shows a simulation of a crowd inside a mall. In the mall there are 11 attractions that the agents can go to based on their desires. There are six different types of attractors: vendors, restrooms, food stores, coffee shops, benches and ATMs (fig.2). The agents have a desired time to depart the facility. This is not a strict constraint in the case of a mall, so the desire to reach the destination is not as high. As a result the people will spend significant time inside the mall.

Default Behavior: The desires of the agents have been randomly selected between 0 and 1. This means that there is an equal probability for an agent to visit a location of each type. Moreover, the desire of an agent for an attraction drops after the agent has visited a location of that specific influence type. Finally, it is enforced that an agent will not visit the same exact location after leaving that location. Figure 6(top) depicts the default behavior of the agents inside the mall. A series of images taken at different points in time shows the accumulation of people inside different locations. In the end, all locations have similar density of visits except the center one, which seems to have more. This is due to an increased maximum capacity for this location, which is difficult to reach given the number of agents. The agents try to avoid locations with high occupancy, which results in many pedestrians prefering to visit this location instead the

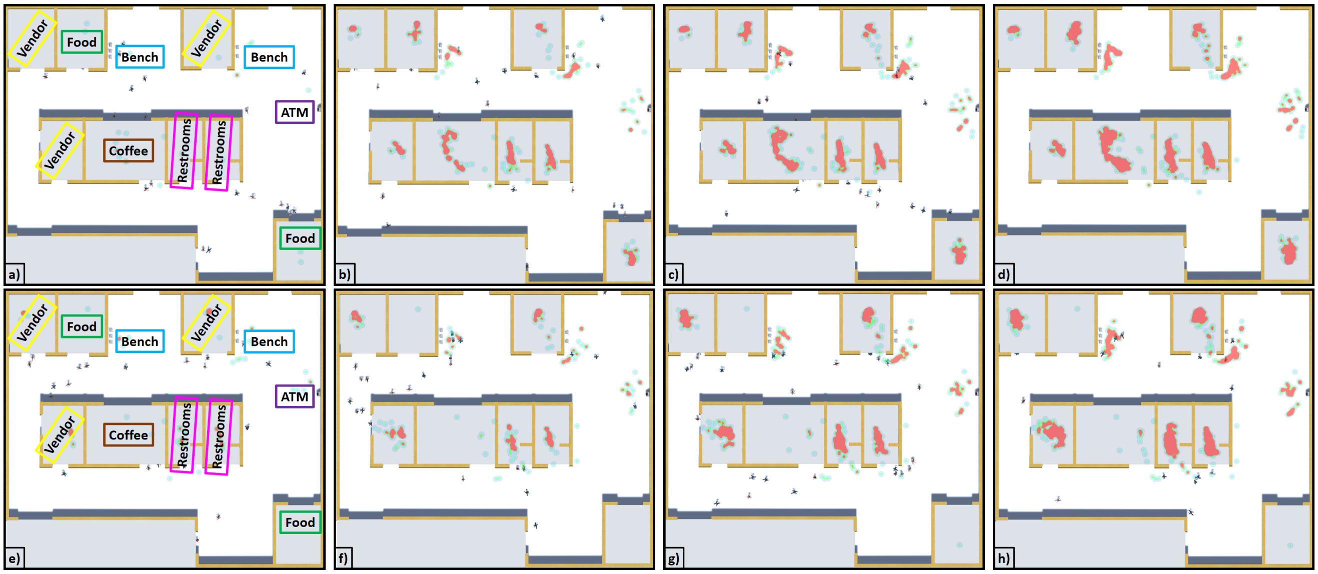

Figure 6: Change in density distributions of agents over time. Top: Default specification. Bottom: Reduced desire for coffee and food store.

other smaller attractors.

Groups: For this experiment there is 40% probability for an agent to be part of a group. The desire of an agent to follow the group is a random value between 0 and 0.5 instead 0 to 1. As result only a smaller number of groups are formed during the simulation.

Variations: By changing the distributions of desires for the different type of attractions, the crowd will adopt new behavior. In the following example the desire for all the attractions are selected between 0 and 1, except of the desires for the coffee and the food stores that are selected between 0 and 0.1. This means that the desire of the agents to visit one of these locations is quite low. Figure 6(bottom) shows that the density of visits to these 3 locations is limited. Only few people decide to visit these locations.

The dynamic influences are also tested in this environment. In the first scenario the agents were executing the default behavior of the experiment. During lunch time, however, the desires for the agents to visit a food store were increased. Figure 10(center) shows that during 1:00pm to 4:00pm there is an increased number of visits to the food stores while the visits to all the other locations is decreased. When people fulfill their desire for food the influence from the food stores drops and as a result, people will visit different locations inside the mall again. The second scenario, shown in Figure 10(right), also takes place during lunch time, but now the agents like to use the restroom or get a cup of coffee after their lunch. This scenario is created with the dynamic influences tool such that after an agent visits a food store, the desire to visit a restroom or the coffee shop is increased.

5.2 Art Museum

This example shows the capability of the framework for authoring complex group behaviors, such as following behaviors or families. The agents are simulated inside an art museum, where it is common for people to move in groups. It is not mandatory the semantic information to specify a location. For this example, the different influence types represent different artists, while the influence source is placed in front of the exhibit that the people would like to visit. This way the desires of the agents will be based on a specific artist and not based on the location of the exhibit. In the museum there are 6 different types of influences, i.e., artists, and 25 locations.

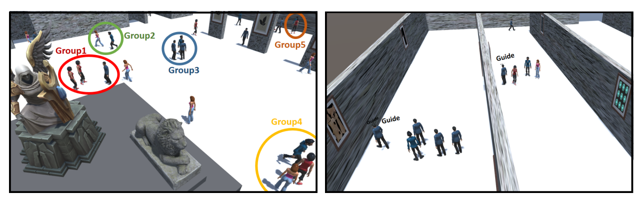

Figure 7: Left: Groups of people are formed in the museum. Right: two big groups of people follow the two guides inside the museum.

Default Behavior: Similar behavior to the mall is tested in this scenario. All the desires are randomly selected between 0 and 1 and the same rules for maximum capacity per location are followed.

Groups: Grouping behaviors can be achieved by treating some moving agents as attractors for other agents. The member of the group with the highest influence on the group will be the attractor for the other members. This agent will decide to go somewhere in the environment based on the influence map created by the attractors' influence and the agent's desires. All the other members of the group will select the position of the group leader as their target. Nevertheless, it is possible for a group member to be attracted by a different attractor in the environment and leave the group in order to visit this location. Given the influence of the leader of the group, the agent who deviated from the group might return.

In a museum there is a large probability to see families where the father or the mother is the leader of the group and the rest of the group members follow the leader. There are also groups of people moving all together and admiring the exhibits. Moreover, there are many groups of people following a guide, who will take the role of the group leader and will act as an attractor for the people that belong to this group.

This scenario examines the capabilities of the framework to form grouping behaviors. In order to see groups inside the facility the probability of an agent to form a group is increased to 80%, while the desire of an agent to follow the group is also selected between 0 and 1, in contrast to the previous example where the desire for grouping was selected between 0 and 0.5.

Variations: A similar test case as in the previous scenario where the desires are selected from 0 to 1 and then some desires reduced to 0.1 has been tested. Figure 8 left, shows that all the locations are uniformly visited with more density around the statue in the center and the art around the statue, that is made from the artist A. Figure 8 right, however, shows the same crowd with the same location semantics, but the desires of the agents to visit exhibits made by artist A reduced. As a result, these location do not attract many people this time.

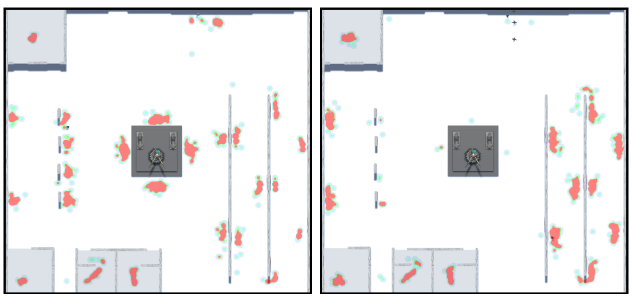

Figure 8: Left: the distribution of visits where all the desires are randomly selected between 0 and 1. Right: The desire of an agent to visit most popular locations is now quite low.

The last scenario shows two guides with each one followed by a group of people. All the members of the group, i.e., the visitors of the museum, will be attracted to the guide and will try to stay as close as they can to this guide, Fig.7 right. Each guide is specialized to exhibits made by a specific artist. To model this behavior each guide has different desires for each influence type.

5.3 Airport

The last example (Fig.9) shows that when the final destination has a high priority, the agents will not be attracted to intermediate locations. Instead, they will actively try to reach their target as soon as possible. For example, in an airport many people will try to get to their gate early, even if they still have time available inside the facility. Only some people will decide to wander around the airport and visit vendors before they go to their gate. This can achieved by specifying smaller desires for the rest of the attractors and increasing the desire of the agents to reach their target.

6. EVALUATION

In this section, we estimate the complexity of authoring a comparable crowd with the same extent of heterogeneity in

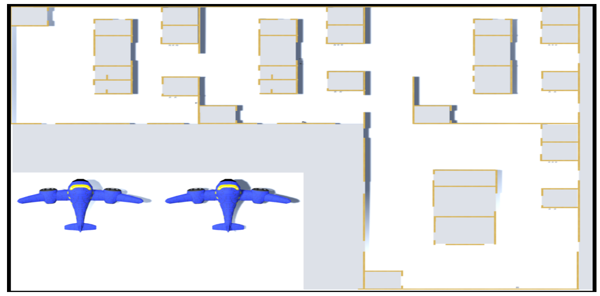

Figure 9: This environment has a lot of attractions; however, the agents have high priority to be at their target on time.

agent behavior, without using an activity-centric authoring paradigm. For an end-user to have a heterogeneous crowd, one will have to specify intermediate locations for each individual agent. On average, an agent visits 5 attractors during the simulations in the presented environments, see Section 5. In each environment about 200 agents are simulated and about 1000 intermediate targets have been selected. Figure 10 shows the visits per attraction type in different points in time. Something not so trivial for the end-user to respect is the maximum capacity value for each attractor. The end-user cannot blindly select intermediate targets for the agents. These intermediate points should be valid for all the agents. In addition, the end-user should not provide an intermediate target further away from the agent's exit, especially when there is not much time left before the departure of the agent from the facility. The proposed method can easily deal with these situations without needing any user involvement. Figure 10 shows that after the environment has the semantics the agents select to go to different locations. The deviation of the visits between attraction types is because the agents have to respect all the rules imposed in the environment, such as maximum capacity.

After the targets for the agents have been specified, the agents will not be able to adapt their decisions and visit new locations inside the environment. For example the scenario where the end-user decided to increase the desire of the agents for food store cannot happen. The end-user will have to know about this change before hand, in order to

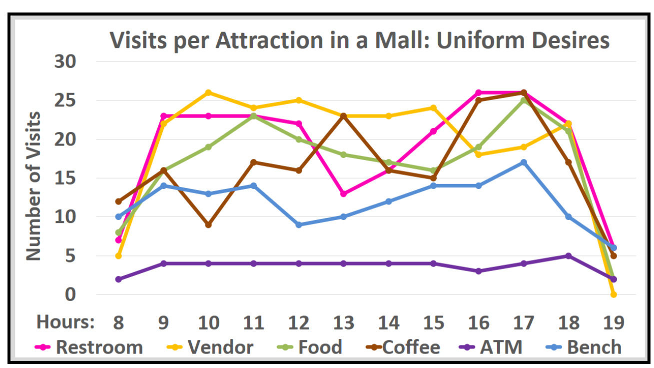

Figure 10: Number of visits per attraction type in the mall when agents' desires are selected randomly between $[0, 1]$ for all the type of attractions.

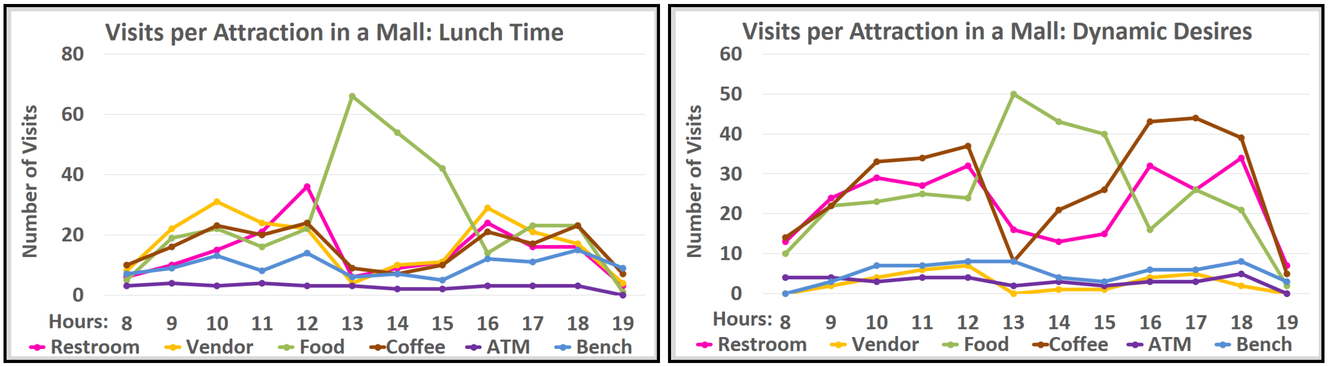

Figure 11: Number of visits per attraction type in the mall over time. (Left) Most people are attracted by a food vendor during lunch time. (Right) After the people fulfill their desire for food, dynamically change their desire to use the restroom or have a cup of coffee.

specify the intermediate targets of the agents given this information. In contrast, our proposed framework can easily handle this scenario by simply increasing the desire of the agents to visit a food store. Figure 11(left) clearly shows the increase of the desires during 1:00pm to 4:00pm. Most of the people decide to visit the food stores, while the density of the visits to the other attractors drops but it is not 0.

Another interesting scenario is depicted in Figure 11(right), where the desires of an agent dynamically change after the agent has visited the food store. For this example the rate of change (K_{ij}) between food store and coffee shop is 1, which means that the desire to visit a coffee shop will be doubled based on the equation 6, and 0.5 between food store and restroom. It is interesting to notice that after the pick for the visits to a food store people have already started visiting the coffee shop and the restroom more than before. Until after the lunch time where the desire of the people to visit a food store is now low, on the other hand the desire to visit the coffee shop or the restroom stays high.

	Mall	Museum	Airport
Early (min)	16.53	13.28	83.38
Late (min)	0.12	2.43	0
People Late	1	3	0

Table 1: The first row shows the average of how early before the time limit people have left the facility. The second and third show the average time people left after their time limit and the percentage of people that left late from the facility, respectively.

The primary task of the proposed method is to simulate heterogeneous crowds in a semantically meaningful environment. Nevertheless, the agents still need to reach their target on time. If the desire to leave the facility is not high enough the people will try to stay as long as they can. If there are no attractors for the agent to visit and also be on time out of the facility, then the agent will leave the facility earlier. Table 1 shows the average time that agents are leaving the environment and how many of them were late. For the tested environments, inside a mall only one agent was late by 0.3 minutes, while all the other agents were leaving on average 17 minutes earlier than their desire time to leave. For the art museum only 3 agents left late from the building, while in the airport that it is important for the agents to be at their destination on time the agents were at their destination on average 83.38 minutes earlier.

Computational Performance. All the experiments ran in real-time (> 60 fps) on an Intel Core Duo 1.6 GHz with 8GB of RAM. The only computation that is needed for the agent to select the next target is the computation of the influence map. Overall, the computational performance of the method scales linearly with the number of agents and environment locations.

7. CONCLUSION

In this paper, we propose an activity-centric approach to authoring heterogeneous, functional, purposeful, crowds in semantically rich virtual environments. Central to our approach is the specification and modelling of how environments and other crowd members influence the behavior of each individual, contingent upon its own personal desires. Our approach affords several benefits. Authors have the flexibility of designing environments and crowds independent of one another. Large, yet varied crowds can be synthesized by specifying only a handful of intuitive parameters, and expert users still have the flexibility of fine-grained control over specific individuals in a crowd, or the ability to author complex group dynamics such as leader following, and families. We demonstrate the benefits of our approach by authoring three scenarios including a mall, a museum, and an airport.

For future work, we would like to conduct a usability study to evaluate the tradeoffs between authoring fidelity and ease of specification. The use of computational intelligence (e.g., domain-independent planners) to facilitate the authoring process is an interesting avenue of future exploration. Finally, our focus in this work is to author crowds of background characters. Complementary to our work is the sunthesis of foreground character activities that conform to narrative constraints.

8. REFERENCES

[1] N. Pelechano, J. M. Allbeck, and N. I. Badler. *Virtual Crowds: Methods, Simulation, and Control*. Morgan and Claypool Publishers, 2008.

[2] D. Thalmann and Soraia Raupp Musse. *Crowd Simulation, Second Edition*. Springer, 2013.

[3] C. W. Reynolds. Steering behaviors for autonomous characters. *Game Developers Conference*, 1999.

[4] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 1995.
[5] J. Berg, S.. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *Robotics Research*. Springer Berlin Heidelberg, 2011.
[6] S. Paris, J. Pettré, and S. Donikian. Pedestrian reactive navigation for crowd simulation: a predictive approach. *Comput. Graph. Forum*, 2007.
[7] A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. In *ACM SIGGRAPH*, 2006.
[8] R. Narain, A. Golas, S. Curtis, and M. C. Lin. Aggregate dynamics for dense crowd simulation. In *ACM SIGGRAPH Asia*, 2009.
[9] Stephen Regelous. Massive, 2001.
[10] Stephane Donikian. Golaem, 2009.
[11] M. Kim, K. Hyun, J. Kim, and J. Lee. Synchronized multi-character motion editing. In *ACM SIGGRAPH*, 2009.
[12] J. Kim, Y. Seol, T. Kwon, and J. Lee. Interactive manipulation of large-scale crowd animation. *ACM Transactions on Graphics*, 2014.
[13] K. H. Lee, M. G. Choi, and J. Lee. Motion patches: building blocks for virtual environments annotated with motion data. In *ACM SIGGRAPH*, 2006.
[14] M. Kim, Y. Hwang, K. Hyun, and J. Lee. Tiling motion patches. In *ACM SIGGRAPH / Eurographics SCA*, 2012.
[15] P. H. Shum, T. Komura, M. Shiraishi, and S. Yamazaki. Interaction patches for multi-character animation. In *ACM SIGGRAPH Asia*, 2008.
[16] K. Jordao, J. Pettré, M. Christie, and M. Cani. Crowd Sculpting: A space-time sculpting method for populating virtual environments. *Computer Graphics Forum*, 2014.
[17] C. Stocker, L. Sun, P. Huang, W. Qin, J. M. Allbeck, and N. I. Badler. Smart events and primed agents. In *Intelligent Virtual Agents*, IVA, 2010.
[18] Q. Yu and D. Terzopoulos. A decision network framework for the behavioral animation of virtual humans. In *ACM SIGGRAPH/Eurographics SCA*, 2007.
[19] A. Shoulson, F. M. Garcia, M. Jones, R. Mead, and N. I. Badler. Parameterizing behavior trees. In *Motion in Games*, 2011.
[20] K. Perlin and A. Goldberg. Improv: A system for scripting interactive actors in virtual worlds. In *Computer Graphics and Interactive Techniques*, 1996.
[21] E. Menou. Real-time character animation using multi-layered scripts and spacetime optimization. 2001.
[22] A. Normoyle, M. Likhachev, and A. Safonova. Stochastic activity authoring with direct user control. In *Interactive 3D Graphics and Games*. ACM, 2014.
[23] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. In *International Joint Conference on Artificial Intelligence*, 1971.
[24] P. Tozour. *Influence Mapping*. Game Programming Gems 2, 2001.
[25] J. Svensson and S.J. Johansson. Influence map-based controllers for ms. pacman and the ghosts. In *Computational Intelligence and Games (CIG),*, 2012.
[26] I. Karamouzas, P. Heil, P. Beek, and M. H. Overmars. A predictive collision avoidance model for pedestrian simulation. In *International Workshop on Motion in Games*, 2009.
[27] M. Schuerman, S. Singh, M. Kapadia, and P. Faloutsos. Situation agents: agent-based externalized steering logic. *Comput. Animat. Virtual Worlds*, 2010.
[28] H. Yeh, S. Curtis, S. Patil, J. van den Berg, D. Manocha, and M. Lin. Composite agents. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2008.
[29] P. Sweetser and J. Wiles. Combining influence maps and cellular automata for reactive game agents. In *Intelligent Data Engineering and Automated Learning-IDEAL 2005*, pages 524–531. Springer, 2005.
[30] P. Avery, S. Louis, and Benjamin Avery. Evolving coordinated spatial tactics for autonomous entities using influence maps. In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, pages 341–348. IEEE, 2009.
[31] Unity3d, 2015.