Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2008)
M. Gross and D. James (Editors)

Composite Agents

H. Yeh, S. Curtis, S. Patil, J. van den Berg, D. Manocha and M. Lin

University of North Carolina at Chapel Hill, NC 27599-3175 USA
{hyeh,seanc,sachin,berg,dm,lin}@cs.unc.edu
http://gamma.cs.unc.edu/CompAgent

Abstract

We introduce the concept of composite agents *to effectively model complex agent interactions for agent-based crowd simulation. Each composite agent consists of a basic agent that is associated with one or more proxy agents. This formulation allows an agent to exercise influence over other agents greater than that implied by its physical properties. Composite agents can be added to most agent-based simulation systems and used to model emergent behaviors among individuals. In practice, there is negligible overhead of introducing composite agents in the simulation. We highlight their application to modeling aggression, social priority, authority, protection and guidance in complex scenes.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

1. Introduction

Over the last few decades, advances in AI techniques, cognitive modeling, and agent-based systems have made modeling of autonomous agents and virtual crowds feasible for offline animations and feature films. Recently, there is growing interest in developing real-time crowd systems for video games [Rey06] and virtual environments. In addition, multi-agent simulation systems are also used for for studying human and social behaviors for architectural and urban design, training and emergency evacuations. For example, a computational framework for analyzing human and crowd behaviors can help improve safe egress analysis and design.

The study of behavior of humans in crowded situations has been an important and fascinating topic in various fields. It is well known that humans not only behave differently in crowded scenarios, but may also undergo temporary personality change, as observed by Gustave LeBon in 1895 [Bon95]. When two or more groups of people meet in the same physical space, many outcomes are possible depending on their mental state and the situation. Crowds can be calm or can suddenly become excited or agitated with skirmishes among the crowd members. One of the key challenges is automatically generating such interactions and simulating crowd movement patterns for agent-based simulations. This requires modeling of how the agents react to other nearby agents and the environment.

Main Results: In this paper, we introduce a simple concept, *composite agents*, which can easily model a variety of emergent behaviors for agent-based crowd simulation. The composite agent formulation provides an elegant method for a single agent to extend its influence over other agents. The idea is to inject intangible factors into the simulation by embodying them in "physical" form and relying on the simulator's pre-existing functionality for local collision avoidance.

We show that the composite agent framework is capable of modeling commonly observed emergent crowd behaviors that arise when humans respond to various social and psychological factors. These include aggression, social priority, authority, protection, guidance, etc. In order to model each of these factors, we present simple algorithms to compute the state of proxy agents that are associated with the crowd behaviors. We have implemented our algorithm in an agent-based simulation system that uses a global road map for navigation and *velocity obstacles* [FS98, vdBLM08] for collision-avoidance. We demonstrate its effect in many complex scenarios such as an emergency evacuation of a building, modeling interactions at a subway station, and modeling authority in a mob. The runtime overhead of adding composite agents to these scenarios with hundreds of agents is negligible.

Organization: The rest of the paper is organized in the following manner. We briefly survey related work in agent-based simulation in Section 2. We introduce the notion of composite agents in Section 3 and use them to model different emergent behaviors in Section 4. We describe our implementation in Section 5 and highlight many applications in Section 6.

2. Related Work

Modeling behaviors of individual agents and virtual crowds has been extensively studied in several fields including computer graphics, robotics, traffic engineering, social sciences, etc. We refer the readers to many excellent surveys [SS01, TOCD06].

Many efficient algorithms have been developed for navigating agents in virtual environments [LD04, SAC*07, BLA02, PLT05, KO04]. Moreover, different methods have been proposed for collision avoidance, including geometric-based [Feu00, FS98, SAC*07, vdBLM08], grid-based [LMM03], force-based [HLTC03, LKF05, SNH01], and divergence-free flow tiles [Che04].

There is considerable work on modeling the local dynamics and generating emergent crowd behaviors. The seminal work of Reynolds demonstrated that simple local rules can generate emergent flocking [Rey87] and other behaviors [Rey99]. Other authors take into account sociological factors [MT97], psychological effects [POSB05], situation-guided control [SGC04], cognitive and behavioral models [FTT99, ST05, YT07], etc. Among these local methods, the social forces model [HM95] has been actively studied and many extensions have also been proposed [CBS*05, BMdOB03, LKF05, SGA*07]. Cellular automata models [BMS02, BKSZ01] and hierarchical approaches [MT01] are also used for modeling different behaviors. Recently, a continuum theory for the flow of crowds was proposed [Hug02] and applied to crowd simulation [TCP06]. Our approach is complementary to most of these methods and can be combined with them to model many emergent behaviors, as described in Section 4.

3. Composite Agents

In this section, we first introduce our terminology and describe a basic framework for agent-based simulation. Next, we present an algorithm to incorporate composite agents into such a framework.

3.1. Definitions and Background

We assume a general agent-based simulation system called SIMULATOR. The set of agents being simulated are denoted as $Agents = \{A_1, A_2, \ldots A_n\}$. Each agent A_i has its own *state*, denoted as ϕ_i. This state can be categorized into an *external state* ε_i and an *internal state* ι_i. ε_i represents properties of A_i that affect the motion of other agents in the system in computing collision-free paths, such as position $\mathbf{p}_i$, velocity $\mathbf{v}_i$ and geometric representation $\mathcal{G}_i$.

The internal state ι_i include properties that are relevant to the agent itself but are not considered by other agents. These may include the goal position of the agent or the memory [LKF05], mental state [ST05], etc. We denote the environment using Φ_{Env}, which consists of the state necessary to navigate a collision-free path through the environment.

We assume that during each time step, the SIMULATOR performs the following functions for each agent:

- Generates a neighbor set using a function called GATHERNEIGHBORS().
- Updates the agent's state using UPDATE().

The GATHERNEIGHBORS function computes the subset of *Agents* that A_i considers when planning its motion. This can be defined in many ways, for example based on the field of view [LKF05], or computing nearest-k neighbors [Rey87, Rey99]. Let $E_{Nbr} = \{\varepsilon_k | A_k \in \text{GATHERNEIGHBORS}(A_i)\}$ denote the collection of all external states of neighbors of A_i.

The UPDATE function can be expressed as $\phi_i \leftarrow \text{UPDATE}(\phi_i, E_{Nbr}, \Phi_{Env})$. Different agent-based simulation systems use different algorithms within UPDATE to evolve the state of agents. For example, a force-based system calculates the repulsive, attractive and frictional forces among agents from their relative positions and velocities [HFV00, LKF05]. Other methods explicitly compute the velocities and positions from the geometric configurations of the agents [vdBLM08, Feu00, PAB07], or use a set of rules to update the state of the agents [Rey99, ST05], with a combination of geographical directions [SGC04]. No matter what mechanism the UPDATE function is based on, the formulation of composite agents exploits its functionality.

3.2. Composite Agents Formulation

We classify the agents into basic agents and composite agents. A *basic* agent is the agent representation native to the SIMULATOR. A *composite agent* is a basic agent A_i that is associated with a set of *proxy agents* $P_{i,j}$'s. The behaviors of these proxies are coordinated with that of the basic agent to achieve particular effects. For example, in one case, the proxies could be thought of as hands extended from the basic agent which get extended towards other agents, encouraging those agents to step away to avoid collision.

This relationship is represented as:

$$\begin{aligned} proxy(A_i) &= \begin{cases} \emptyset & \text{for basic agents} \\ \{P_{i,1}, P_{i,2}, \ldots, P_{i,m}\} & \text{for composite agents} \end{cases} \\ parent(P_{i,j}) &= A_i. \end{aligned}$$

A proxy agent $P_{i,j}$'s state includes an external state $\varepsilon_{i,j}$, which consists of the same properties as in the basic agent's external state, and a unique internal state $\iota_{i,j}$. We require that $P_{i,j}$ has access to the internal state, ι_i, of its parent A_i. We denote the set of all proxy agents in the simulation as $Proxies = \bigcup_i proxy(A_i)$.

The fact that a proxy agent possesses the same set of external properties as a basic agent, and that UPDATE only considers the external states of the neighboring agents, leads to the central idea behind composite agents: both the basic agents and proxy agents are treated uniformly by the UPDATE function. Therefore other agents react to a proxy agent in exactly the same way as they would to a basic agent. The proxy agent, however, updates itself according to a unique set of rules, defined in the P-UPDATE function. This function includes in its input the full state of the parent agent, not just the external state. Given this formulation of proxy and composite agents, the overall simulation algorithm proceeds as follows:

- for each $A_i \in Agents$
  - $Nbr \leftarrow$ GATHERNEIGHBORS$(Agents_i^*)$
  - $\phi_i \leftarrow$ UPDATE$(\phi_i, E_{Nbr}, \Phi_{Env})$
- for each $P_{i,j} \in Proxies$
  - $\phi_{i,j} \leftarrow$ P-UPDATE$(\phi_{i,j}, \phi_i, \Phi_{Env})$

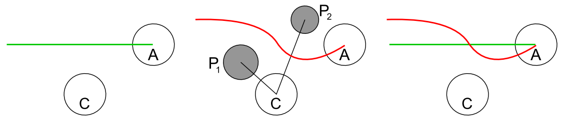
The GATHERNEIGHBORS function now selects the relevant neighbors from a larger pool: $Agents_i^* = Agents \cup Proxies - proxy(A_i)$. Clearly, a composite agent should not consider its own proxies as obstacles. So, those proxies are excluded from its neighbor set. Once the Nbr data structure is computed, the unchanged UPDATE function computes the result according to all of the neighbors, basic and proxy alike.

3.3. Influence of Composite Agents

The fact that an agent reacts to both basic and proxy agents equivalently has a direct consequence. The influence that a composite agent C_i exerts over other agents is extended beyond its own external properties ε_i, to indirectly include all the influences of the $\varepsilon_{i,j}$'s of its proxy agents. Fig. 1 illustrates a basic example. When an agent A encounters a composite agent C, it observes both the latter's basic and proxy agents, and computes a path to avoid collision with all of them. The influence of C over A is different from that of a basic agent, thus enriching the way A interacts with C.

4. Modeling Intangible Factors

In the previous section, we gave an overview of composite agents. In this section, we show that different emergent behaviors can be easily modeled using composite agents. In each case, we describe the phenomenon observed in a real crowd, briefly discuss the social or psychological factor underlying this phenomenon and propose an intuitive mechanism to embody the factor into a proxy agent. Finally, we translate the mechanism into the proxy update function, P-UPDATE(), such that the collective behavior exhibited by a crowd of the resulting composite agents agrees with our observations.



(a) (b) (c)

Figure 1: *Responses of an agent A encountering a composite agent C. (a)The green line shows the original planned path taken by A. (b) In the presence of the proxy agent of C, A takes the red path and avoids collision with P_1 and P_2. (c)Comparison of the paths.*

For the purpose of this discussion, we assume that the external state consists of position, velocity and geometric representation, i.e. $\varepsilon = (\mathbf{p}, \mathbf{v}, \mathcal{G})$, although the agent-based simulation algorithm may also consist of additional terms.

4.1. Aggression

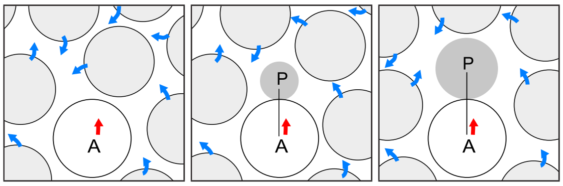
Aggressive behavior can be characterized as follows:

1. A person feels a sense of urgency—the desire to reach a goal more quickly.
2. The urgency is expressed in some manner causing other agents to either yield or steer clear.

In real world scenarios, urgency can be perceived through various media, such as gestures, noises or social protocols. For example, a person communicates urgency through stance, stride and manner. Similarly, a police officer can show his urgency by using his car's sirens. Other people accommodate for that urgency, and, as a result, the aggressive agent carves its way through a congested environment. Similar psychological factors have been modeled before, such as the panic situation in HiDAC [PAB07], the hurry factor in social forces [LKF05], etc. Our formulation is different from these models in terms of how urgency is conveyed to other agents and and how the other agents respond.

The first characteristic highlighted above can be captured by introducing an extra property, URGENCY. In order to communicate the urgency to other agents, we associate one proxy agent $P_{i,1}$, called an *aggression proxy*, to the agent A_i. The proxy is placed near A_i in the direction it intends to move, as shown in Fig. 2(b). Intuitively, $P_{i,1}$ serves as a "cowcatcher" on a train—its presence clears the space in front of A_i because other agents avoid colliding with it and take a detour around it. The resulting space affects A_i's motion and makes it possible to move in a desired direction and carve a path through the crowd.

If we assume constant URGENCY, the P-UPDATE function could be formulated as:



(a) (b) (c)

Figure 2: ***Aggression****: Agent A's desired direction is blocked. As A's urgency increases, its aggression proxy, P, grows and the other agents move to avoid it, leaving a space for A to move into.*

- $\mathbf{p}_{i,1}$ is positioned at a distance from $\mathbf{p}_i$ in the direction that A_i intends to move.
- $\mathbf{v}_{i,1}$ is chosen to be identical to $\mathbf{v}_i$.
- $\mathcal{G}_{i,1}$ is a simple shape, such as a circle (as appropriate for the simulator.)

We can also model changing URGENCY. We consider two functions to simulate factors that contribute to urgency. and blend them together to form a single URGENCY value in the range $[0,1]$:

1. *Velocity-based urgency*: An agent becomes more urgent if it is not going in the preferred direction and speed. The greater the deviation of the current velocity from the preferred velocity, the greater this value grows.
2. *Distance-based urgency*: The distance to the goal is compared before and after the time step. If the agent gets closer, the URGENCY value reduces; if the agent gets farther, the URGENCY value increases.

We now relate the size and distance d of the aggression proxy to the URGENCY value. In other words, if A_i has a higher URGENCY, the proxy agent becomes larger and is placed farther from A_i, thereby clearing more space for A_i. The new extended P-UPDATE function that models such urgency-based behavior is given as:

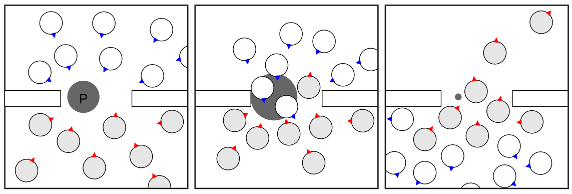
- $\mathbf{p}_{i,1}$ is placed at a distance d, proportional to URGENCY, from $\mathbf{p}_i$;
- $\mathcal{G}_{i,1}$ is scaled to a factor proportional to URGENCY, so that as A_i's urgency increases, so does the size of its proxy agent.

Fig. 2(a) through Fig. 2(c) illustrate this formulation

4.2. Social Priority

When traveling by elevator, it is standard practice for people exiting the elevator to be allowed egress first. If sufficient space is available, then people can enter and exit simultaneously. This is a special case of a more general social protocol: when space is limited and contested, some people are granted higher *priority* to occupy the space.

This social priority acts like a beachhead at the contested site, letting the higher-prioritized people to pass through but not the lower-prioritized people.



(a) (b) (c)

Figure 3: ***Priority****: The white agents should be given preference in passing through the doorway. (a) Each white agent has a priority proxy located at P and identical priority values. (b) As the white agents approach, the proxy grows, reserving the space for all of the white agents. (c) Finally, after the white agents have passed, the proxy shrinks to nothing and the gray agents may pass through unimpeded.*

To model this behavior using composite agents, we introduce a new property: PRIORITY. By definition, we say that a basic agent has lower priority than all composite agents. A proxy agent $P_{i,1}$, called a *priority proxy*, is placed at the contested location and grows as its parent A_i nears it. An agent with a lower priority observes that the space is occupied by the priority proxy and plans around it, thus, implicitly giving preference to higher-prioritized agents to pass through first. The P-UPDATE function is formulated as:

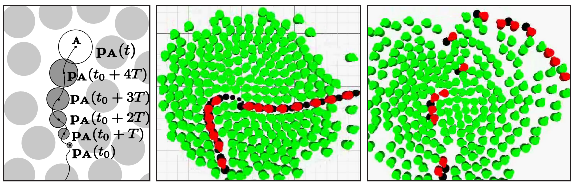
- $\mathbf{p}_{i,1}$ is set right at the contested location;
- $\mathbf{v}_{i,1}$ is set to zero;
- $\mathcal{G}_{i,1}$ grows as A_i approaches the contested location, and shrinks as A_i leaves.

We illustrate our formulation using the doorway example highlighted in Fig. 3. Note that there is no explicit behavior prescribed for agents with lower priority.

4.3. Authority

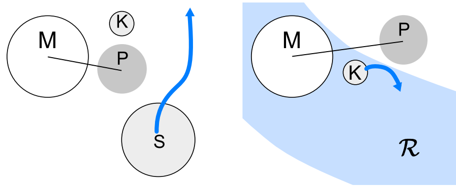
We observe that when a line of soldiers or fire-fighters march into a dense crowd and they are still able to maintain a coherent line. Their authority makes it so that even if there is space between two consecutive members, civilians do not attempt to break the line. We can approximate this manifestation of authority with a *trailblazer*, who marks space that the members of his group can travel through while others cannot.

The above trailblazer can be modeled using composite agents. We add a TRAIL IDENTIFIER property. This property controls which "trail" a composite agent follows. We assign a set of proxy agents, *trail proxies*. A trail proxy marks the path traveled by the composite agent. After every T seconds of the simulation, an agent places a proxy agent at its current position. The sequence of proxy agents marks the most recent segment of the path that the agent has traveled. These proxies serve as obstacles to other agents, both



(a) (b) (c)

Figure 4: ***Authority**: (a) An agent A and the trail (a sequence of trail proxies.) The trail proxies are placed at positions $\mathbf{p}_i$, at time instants t_0, t_0+T, t_0+2T, t_0+3T and t_0+4T. (b) A line of police maintains a formation while walking in a crowd; the police are associated with trail proxies and aggression proxies. (c) A simulation with the same initial configuration except without trail proxies.*



(a) (b)

Figure 5: *(a) Protection: a mother M protects her child K by placing a proxy agent P in between K and an approaching stranger S. (b) Guidance: when K is about to stray from the correct pathway, indicated as the region $\mathcal{R}$, the mother places a proxy agent P just outside $\mathcal{R}$ to alter K's direction.*

the basic and other composite agents which do not have the same TRAIL IDENTIFIER. Therefore they create an available path for composite agents with the same TRAIL IDENTIFIER (Fig. 4(a)).

We formulate this behavior in the following manner. Consider a trailblazer A_i and its proxy agents $P_{i,1}, P_{i,2}, \ldots, P_{i,m}$. We say that $P_{i,j}$ has a life cycle of period τ that starts at time $start_j$, and an age, represented as age_j, which increases as simulation time passes. When age_j becomes greater than τ, age_j is reset to 0, the starting time $start_j$ is set to the current time t, and the cycle starts again. At the beginning of the cycle, the position of the proxy is set to be that of the parent and its size is set to be the same as the parent. As the proxy agent ages, it shrinks.

The P-UPDATE function for $P_{i,j}$ is expressed as:

- $\mathbf{p}_{i,j}$ is equal to $\mathbf{p}_i(start_j)$, i.e. where A_i was when the cycle started;
- $\mathbf{v}_{i,j}$ is zero;
- $\mathcal{G}_{i,j}$ is similar to $\mathcal{G}_i$ and scaled to the factor $1 - \frac{age_j}{\tau}$;
- internal state: age_j is increased by Δt; if $age_j \geq \tau$, age_j is set to 0 and $start_j$ is set equal to the current time t.

Initially we let $start_1, start_2, \ldots, start_m$ to be equal to $0, T, \ldots, (m-1)T$, respectively. Fig. 4(a) also marks the starting time for each proxy agent. Fig. 4(b) shows a working example of trailblazers. In contrast, Fig. 4(c) shows the same scenario without trailblazers. The red agents still try to in a line move to the same goal, but fail to maintain the formation and are scattered by the crowd.

4.4. Protection and Guidance Behavior

Composite agents can also be used to facilitate interactions involving protection or guidance. Examples of such interactions arise, when a child walks with a mother in a dense crowd. The child has a limited field of view and cannot detect all possible collisions with the other agents, and may not have the information about a global path or goal in terms of global navigation. The mother protects the child from possible collisions and guides the child to stay on the current path.

Modeling such behavior involves very specialized individual behaviors for the mother M. These include:

1. Maintaining extra information that the mother needs to know where the child K is, predicts collisions for the child, and determines whether the child's moving direction is in a certain range;
2. Reacting to the situation, i.e. offering protection and guidance.

These behaviors can be easily modeled using composite agents. We associate a proxy agent P_1 with the mother M. For protection behavior, suppose the mother detects that a stranger S is approaching, then

- $\mathbf{p}_{i,1}$ is set to be in between K and S, say $\mathbf{p}_{i,1} = \frac{1}{2}(\mathbf{p}_k + \mathbf{p}_s)$;
- $\mathbf{v}_{i,1}$ is set to be equal to $\mathbf{v}_M$;
- $\mathcal{G}_{i,1}$: any shape that obstructs the trajectory for S to hit K.

It is possible that S will eventually avoid K without the protection, but it may come very close to K and barely pass by. The mother may dislike the situation and prevents this from happening. The presence of $P_{i,1}$ forces S to maneuver earlier. Fig. 5(a) demonstrates this formulation.

In terms of guidance behavior, suppose the mother detects that K is about to head outside of a region $\mathcal{R}$, which she thinks is an acceptable pathway, then

- $\mathbf{p}_{i,1}$ is set to be slightly outside of $\mathcal{R}$, along the line defined by $\mathbf{v}_k$;
- $\mathbf{v}_{i,1} = \mathbf{v}_M$;
- $\mathcal{G}_{i,1}$: any shape that is sufficient to block K.

See Fig. 5(b) for an illustration. In this case it is K who detects the presence of P_1 and steers away from it.

5. Implementation

Simulator: Our approach can be incorporated into most agent-based simulation systems. Our current implementation is based on *Reciprocal Velocity Obstacles*

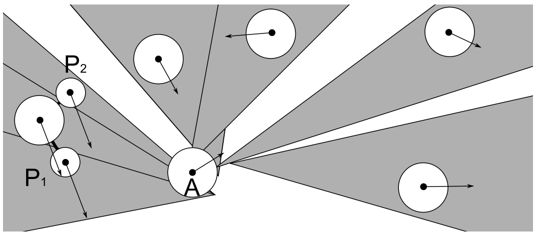


Figure 6: *The Reciprocal Velocity Obstacles (RVO) induced by multiple agents on agent A. Agent A, not aware of the fact that some contributors of RVO's are proxy agents (P_1 and P_2), chooses the least penalized velocity to be its next velocity.*

(RVO) [vdBLM08]. Each agent in the simulation, A_i has position $\mathbf{p}_i$, velocity $\mathbf{v}_i$, and geometric shape $\mathcal{G}_i$ associated with it. The specific UPDATE function for the RVO algorithm takes into account the position and velocity of nearby agents of A_i to compute a new velocity and direction of motion for A_i in the following manner.

Given two agents, A_i and A_j, let $\mathbf{p}_i$ and $\mathbf{p}_j$ be their current positions, and $\mathbf{v}_i$ and $\mathbf{v}_j$ be the current velocities, respectively. Let $\lambda(\mathbf{p},\mathbf{v})$ define the ray shot from point $\mathbf{p}$ in the direction along $\mathbf{v}$ (i.e. $\lambda(\mathbf{p},\mathbf{v}) = \mathbf{p}+t\mathbf{v}$). Moreover, $\mathcal{G}_i \oplus \mathcal{G}_j$ denotes the *Minkowski* sum of two geometric primitives $\mathcal{G}_i$ and $\mathcal{G}_j$, i.e. $\mathcal{G}_i \oplus \mathcal{G}_j = \{\mathbf{x}_i+\mathbf{x}_j | \mathbf{x}_i \in \mathcal{G}_i, \mathbf{x}_j \in \mathcal{G}_j\}$. Let $-\mathcal{G}_i$ denote the shape $\mathcal{G}_i$ reflected in its reference point, i.e. $-\mathcal{G}_i = \{-\mathbf{x}_i | \mathbf{x}_i \in \mathcal{G}_i\}$. The reciprocal velocity obstacle RVO^i_j that agent A_j induces on agent A_i is defined as follows:

$$RVO^i_j = \{\mathbf{v}'_i | \lambda(\mathbf{p}_i, 2\mathbf{v}'_i - \mathbf{v}_i - \mathbf{v}_j) \cap \mathcal{G}_j \oplus -\mathcal{G}_i \neq \emptyset\}. \quad (1)$$

If agent A_i chooses a new velocity outside RVO^i_j and agent A_j chooses a new velocity outside RVO^j_i, the agents are *guaranteed* to have chosen a collision-free and oscillation-free trajectory [vdBLM08].

In terms of multi-agent navigation, the RVO formulation is applied as follows to each agent independently, as shown in Fig. 6. Among its admissible velocities, the RVO algorithm selects the one with a minimal penalty. This penalty is defined among other things in terms of the distance between the chosen velocity and the *preferred* velocity (the lower the better), and the expected *time to collision* computed based on the chosen velocity (the higher the better). Notice that from the perspective of agent A_i, it does not know whether each reciprocal velocity obstacle is from a proxy agent or a basic agent. Overall, we can assume that given a set of agents $Agents = \{A_1, A_2, \ldots\}$, the function $\mathrm{RVO}(A_i, Agents - A_i)$ returns the optimal velocity for agent A_i for the next simulation cycle. For details on this function, we refer the readers to [vdBLM08].

We chose to implement our approach with RVO because it produces collision-free and oscillation-free trajectories even in highly dense scenarios. The concept of composite agents, however, can be naturally mapped in other frameworks too. In the social forces model, a proxy agent exerts forces (e.g. repulsion, attraction and friction) on basic agents, and affects the trajectories of the latter. In Reynolds' steering model [Rey99], a proxy agent plays the same role as a basic agent when others perform cohesion, alignment, separation and collision avoidance. In cellular automata framework [BMS02, BKSZ01] a proxy agent also occupies a cell, and affects other cells' state transition.

Proxy Update: In our implementation, proxy agents are fully responsible for updating their own state. All additional information of the parent that is relevant for a specific behavior, (e.g. URGENCY and PRIORITY), are maintained in the proxies. A basic agent then does not need to know that there are proxies associated with it, thereby eliminates the need to re-define or inherent the basic agent class.

This also allows behaviors associated with each kind of proxy agents to be arbitrarily *composed*, because an agent can have a heterogeneous set of proxy agents. For example, it is reasonable to assign both a priority as well as an aggression proxy to a composite agent. The resultant agent would have the priority to pass through a narrow passage and a better capacity to push through the other agents surrounding the doorway.

Dynamic State: We also let the proxy agents respond differently to queries about their properties ($\mathbf{v}$, $\mathbf{p}$, and $\mathcal{G}$) depending on who is querying. Because velocity plays a fundamental role an RVO-based simulator, the priority proxy always reports velocity towards the querying agent, with the speed based on its growth rate. This satisfies the planning algorithm in RVO better than a constant (or zero) velocity would.

Conditional Neighbors: Recall that the function GATHERNEIGHBORS collects an agent's relevant neighbors. An agent enters another agent's neighbor set *Nbr* if it fulfills certain criteria (spatial proximity, group relationship, etc.) Besides the normal criteria for belonging to the neighbor set, proxies may require additional criteria. Besides the fact that an agent should not react to its own proxy agents, priority proxies, for example, do not belong in the neighbor set of agents with greater than or equal priorities. Likewise, trail proxies do not belong in the neighbor set of agents with the same trail id. In our implementation, the proxy agent has the power to reject being included in another agent's *Nbr*. By doing so, this keeps proxy logic out of the GATHERNEIGHBORS functionality.

Visualization: We use a simple method to retrofit human locomotion onto the simulated paths. The idea of composite agents is orthogonal to the computational model of human locomotion.

6. Results

We demonstrate some of the benefits of using composite agents in different scenarios.

Office Evacuation: This scenario depicts an emergency evacuation from an office building (Fig. 7). As part of the evacuation procedure, all the agents move towards the exits. A fraction of these agents have aggression proxies associated with them. These agents are able to carve their way through the dense crowd and evacuate the building more quickly than the other agents. This fact is also highlighted in the accompanying video. We also observed that if multiple aggressive agents tried to make their way through an exit at the same time, they interfere with each other creating congestion at the doorway and slowing down the overall evacuation flow—which is in agreement with what happens in real life.

Subway Station: In this scenario, we simulate the behavior of pedestrians in a crowded subway station when a train has just arrived. The priority proxies are set up at each of the train's exits, and the exiting agents have a higher priority associated with them than the boarding agents. The proxies behave much like a soft constraint; boarding agents defer to exiting agents, but may board simultaneously if there is space. The outcome is highlighted in the supplementary video and in Fig. 8.

Embassy: In the scenario shown in Fig. 9, we simulate a crowd protesting in front of the gates of an embassy. The objective of the policemen is to clear the mob and make way for the ambassador's car. The task is accomplished in two stages:

1. Two ranks of policemen make their way through the mob and separate the protesters into two halves.
2. The policemen march forward, thereby clearing the path in front of the gate and allowing the car to depart

The police agents have aggression proxies to help carve their way through the mob and have trail proxies to help maintain the integrity of the police line.

Analysis: Table 1 summarizes the performance of our system on the three demo scenarios. The third column indicates the additional number of proxy agents added to the simulation setup to emulate the desired behaviors. The additional overhead of using the composite agent framework with an existing multi-agent simulation system is measured by comparing the simulation time (in frames per second) and memory usage of the demo scenario with and without the proxy agents.

Limitations: Our method enriches the set of agent interactions that can be modeled with a basic agent-based simulation system. But there are some difficulties inherent in this approach. First, behaviors may not necessarily admit intuitive physical incarnation, e.g. behaviors complicated communication or group coordination. Second, composite agents rely on the mechanism provided by the underlying planning system (e.g. collision avoidance), this level of indirection disallows precise control over the exact nature of the agent interactions. Unpredictable results could *possibly* be obtained, though we have not encountered them in our simulations.

Scene	#Basic agents	#Proxy agents	% Overhead simulation time	% Overhead memory usage	Type of proxy agents
Office	1000	47	1.9%	0.6%	aggression
Subway	340	100	0.3%	0.12%	priority
Embassy	240	200	10.75%	1.9%	trail, aggression

Table 1: *Performance of our approach on the three demo scenarios. The results indicate that the composite agent framework adds very little overhead to an existing multi-agent simulation system in terms of both simulation time and memory usage.*

7. Conclusions and Future Work

We introduce a novel concept, *composite agents*, for modeling various crowd behaviors with little computational overhead to the overall simulations. We have successfully demonstrated their application by modeling various intangible factors, such as aggression, social priority, authority, protection, guidance, etc. In the near future, we would like to model other types of agent behaviors using composite agents and apply them to different scenarios. Secondly, we would like to validate the human-like behaviors generated by composite agents. Furthermore, we would like to explore the different emergent behaviors when our model is incorporated with different agent-based simulation systems. Finally, we would like to extend the idea to model group behaviors.

Acknowledgment

This research was supported in part by ARO Contracts DAAD19- 02-1-0390 and W911NF-04-1-0088, NSF awards 0400134, 0429583 and 0404088, DARPA/RDECOM Contract N61339-04- C-0043, Intel, Carolina Development, and Disney.

References

[BKSZ01] BURSTEDDE C., KLAUCK K., SCHADSCHNEIDER A., ZITTARTZ J.: Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications 295* (2001), 507–525.

[BLA02] BAYAZIT O. B., LIEN J.-M., AMATO N. M.: Better group behaviors in complex environments with global roadmaps. *Int. Conf. on the Sim. and Syn. of Living Sys. (Alife)* (2002), 362–370.

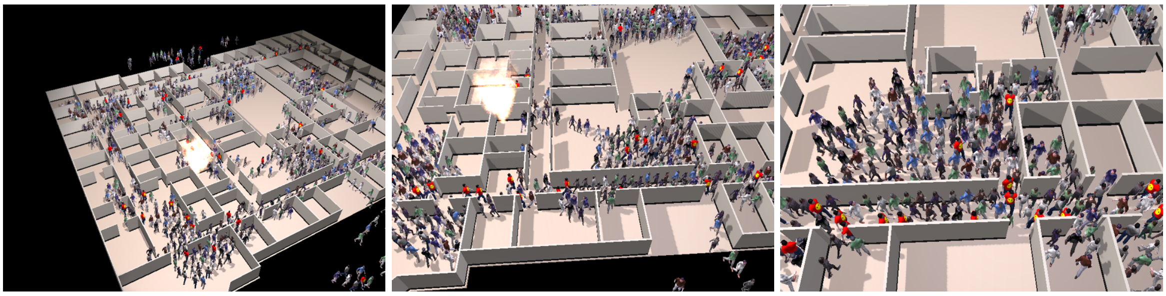


Figure 7: *Emergency evacuation in an office building: The agents in red are aggressive agents. They are able to carve their own way through the crowd and exit more quickly than the others.*

Figure 8: *A crowded subway station: The exiting agents have a higher priority and are given preference to pass through the doorway first. The priority proxy formulation eliminates the need for any kind of explicit coordination between the exiting and boarding agents.*

[BMdOB03] BRAUN A., MUSSE S. R., DE OLIVEIRA L. P. L., BODMANN B. E. J.: Modeling individual behaviors in crowd simulation. *casa* (2003), 143.

[BMS02] BANDINI S., MANZONI S., SIMONE C.: Dealing with space in multi–agent systems: a model for situated mas. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3* (2002), 1183–1190.

[Bon95] BON G. L.: *The Crowd: A Study of the Popular Mind*. 1895. Reprint available from Dover Publications.

[CBS*05] CORDEIRO O. C., BRAUN A., SILVERIA C. B., MUSSE S. R., CAVALHEIRO G. G.: Concurrency on social forces simulation model. *First International Workshop on Crowd Simulation* (2005).

[Che04] CHENNEY S.: Flow tiles. *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), 233–242.

[Feu00] FEURTEY F.: *Simulating the Collision Avoidance Behavior of Pedestrians*. Master's thesis, Univ. of Tokyo, 2000.

[FS98] FIORINI P., SHILLER Z.: Motion planning in dynamic environments using velocity obstacles. *International Journal on Robotics Research 17*, 7 (1998), 760–772.

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

[HFV00] HELBING D., FARKAS I., VICSEK T.: Simulating dynamical features of escape panic. *Nature 407* (2000), 487–490.

[HLTC03] HEÏGEAS L., LUCIANI A., THOLLOT J., CASTAGNÉ N.: A physically-based particle model of emergent crowd behaviors. *Graphikon '03* (2003).

[HM95] HELBING D., MOLNÁR P.: Social force model for pedestrian dynamics. *Phys. Rev. E 51*, 5 (May 1995), 4282–4286.

[Hug02] HUGHES R. L.: A continuum theory for the flow of pedestrians. *Transportation Research Part B: Methodological 36* (July 2002), 507–535.

[KO04] KAMPHUIS A., OVERMARS M.: Finding paths for coherent groups using clearance. *Proc. of ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (2004), 19–28.

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

[LKF05] LAKOBA T. I., KAUP D. J., FINKELSTEIN

Figure 9: *A crowd of protesters outside an embassy: Two ranks of policemen clear the protesters off the road. Notice that when forcing their way into the crowd, even if the actual gap between the individual policemen is enough for a protest or to pass through, the perceived continuity of authority prevents the protesters from breaking the police line.*

N. M.: Modifications of the helbing-molnar-farkas-vicsek social force model for pedestrian evolution. *SIMULATION 81* (2005), 339.

[LMM03] LOSCOS C., MARCHAL D., MEYER A.: Intuitive crowd behaviour in dense urban environments using local laws. In *Theory and Practice of Computer Graphics (TPCG'03)* (2003), pp. 122–129.

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

[MT01] MUSSE S. R., THALMANN D.: Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics 7* (2001), 152–164.

[PAB07] PELECHANO N., ALLBECK J. M., BADLER N. I.: Controlling individual agents in high-density crowd simulation. *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), 99–108.

[PLT05] PETTRE J., LAUMOND J.-P., THALMANN D.: A navigation graph for real-time crowd animation on multilayered and uneven terrain. *First International Workshop on Crowd Simulation* (2005).

[POSB05] PELECHANO N., O'BRIEN K., SILVERMAN B., BADLER N.: Crowd simulation incorporating agent psychological models, roles and communication. *First International Workshop on Crowd Simulation* (2005).

[Rey87] REYNOLDS C. W.: Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics 21* (1987), 25–34.

[Rey99] REYNOLDS C. W.: Steering behaviors for autonomous characters. *Game Developers Conference 1999* (1999).

[Rey06] REYNOLDS C.: Big fast crowds on ps3. In *sandbox '06: Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames* (2006), ACM Press, pp. 113–121.

[SAC*07] SUD A., ANDERSEN E., CURTIS S., LIN M., MANOCHA D.: Realtime path planning for virtual agents in dynamic environments. *Proc. of IEEE VR* (2007).

[SGA*07] SUD A., GAYLE R., ANDERSEN E., GUY S., LIN M., MANOCHA D.: Real-time navigation of independent agents using adaptive roadmaps. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology* (2007), ACM, pp. 99–106.

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

[SNH01] SUGIYAMA Y., NAKAYAMA A., HASEBE K.: 2-dimensional optimal velocity models for granular flows. In *Pedestrian and Evacuation Dynamics* (2001), pp. 155–160.

[SS01] SCHRECKENBERG M., SHARMA S. D.: *Pedestrian and Evacuation Dynamics*. Springer, 2001.

[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.

[TCP06] TREUILLE A., COOPER S., POPOVIC Z.: Continuum crowds. *Proc. of ACM SIGGRAPH* (2006), 1160 – 1168.

[TOCD06] THALMANN D., O'SULLIVAN C., CIECHOMSKI P., DOBBYN S.: *Populating Virtual Environments with Crowds*. Eurographics 2006 Tutorial Notes, 2006.

[vdBLM08] VAN DEN BERG J., LIN M., MANOCHA D.: Reciprocal velocity obstacles for realtime multi-agent navigation. *Proc. of IEEE Conference on Robotics and Automation* (2008).

[YT07] YU Q., TERZOPOULOS D.: A decision network framework for the behavioral animation of virtual humans. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), pp. 119–128.