BEHAVIORAL ANIMATION FOR CROWD SIMULATION

PH.D. THESIS PROPOSAL

DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

ANKARA/TURKEY

by

Funda Durupınar

April, 2007

ABSTRACT

BEHAVIORAL ANIMATION FOR CROWD SIMULATION

Funda Durupınar

Ph.D. Thesis Proposal in Computer Engineering

Supervisor: Assoc. Prof. Dr. Uğur Güdükbay

April, 2007

Autonomous agents appear in many areas of computer animation, from the simulation of biological systems to computer games. Behavioral animation systems model and simulate autonomous agents that act according to some predefined rules and environmental influences. Behavioral animation covers a wide range of problems that are common with robotics, artificial intelligence and artificial life. However, the main application area of behavioral animation is the situations involving a large number of entities, i.e., crowd simulation. Thus, the tedious and labor-intensive work of motion design process is automated, leaving merely the behavior design to the animator.

The main purpose of this thesis study is to develop a behavioral animation system for crowd simulation. The system will include components like perception, memory, learning, motor control, etc. Existing systems in the literature are specific to certain situations only. We aim to develop a generic platform that enables the animator to author certain scenarios with a large number of autonomous agents that perform given tasks. Another issue about the existing systems is their time complexity. They are computationally inefficient ($O(n^2)$) as they require the comparison of each agent with every other agent. We also plan to develop algorithms to decrease the computational cost.

Keywords: Behavioral animation, autonomous agents, virtual crowds, planning, artificial life, learning, perception.

ÖZET

TOPLULUK SIMULASYONU ICIN DAVRANIŞSAL ANIMASYON

Funda Durupınar
Bilgisayar Mühendisliği, Doktora Tez Önerisi
Tez Yöneticisi: Doç. Dr. Uğur Güdükbay
Nisan, 2007

Özerk etmenler, biyolojik sistemlerden bilgisayar oyunlarına, bilgisayar animasyonunun çesitli alanlarında yer alırlar. Davranışsal animasyon sistemleri, çevresel etmenlere bağlı olarak önceden belirlenmiş birtakım kurallara göre hareket eden özerk etmeri modeller ve canlandırır. Davranışsal animasyon, robotbilim, yapay zeka ve yapay yaşam gibi alanlarla ortak problemleri içerir. Bununla birlikte, davranışsal animasyonun esas kullanım alanı, topluluk simülasyonu gibi, çok sayıda varlıktan oluşan sistemlerin simülasyonudur. Böylece, animatöre sadece davranış tasarımı işlemi bırakılarak, sıkıcı ve yorucu olan hareket tasarımı işlemi otomatikleştirilmiş olur.

Bu tez çalısmasının temel amacı, topluluk simülasyonu için bir davranışsal animasyon sistemi geliştirmektir. Bu sistem, algı, hafıza, öğrenme, hareket kontrolü gibi unsurlardan oluşacaktır. Halihazırdaki sistemler sadece belli durumlara özgü calışmaktadır. Biz, animatörün, çok sayıda özerk etmenin verilen görevleri yerine getirdiği belli senaryoları yaratabileceği genel bir sistem geliştirmeyi amaçlıyoruz. Varolan sistemlerle ilgili bir başka konu da her etmeni bir diğer etmenle karşılaştırmak gerekmesi nedeniyle hesaplama bakımından verimsiz oluşudur ($O(n^2)$). Aynı zamanda, hesaplama maliyetini düşürecek algoritmalar geliştirmeyi de planlamaktayız.

Anahtar Sözcükler:

Davranışsal animasyon, özerk etmenler, sanal topluluklar, planlama, yapay yaşam, öğrenme, algı.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Behavioral animation refers to the systems that are made up of embodied autonomous agents that act by some predefined rules in accordance with environmental influences. An autonomous agent can be any entity that responds to the environmental stimuli and has internal motivation. The rules determine the agent's decision making process by using the information acquired from the environment. Behavioral animation spans a large area, from path planning to complex emotional interactions between characters. However, behavioral animation methods are typically applied to situations involving a large number of entities, where each entity must act both independently and in harmony with the rest of the group. Some of the examples where behavioral animation is suitable for are the simulation of flocks, armies, human crowds, pedestrians or vehicles in traffic.

One of the advantages of behavioral animation is the efficient representation of realistic and complex behavior of a large number of agents, automating the motion design process. Another advantage is the utilization of reusable characters, which is especially beneficial to the movie industry and computer games. To be more specific, some of the movies that utilize behavioral animation include "Batman Returns", "The Lion King", "From Dusk Till Dawn", "The Hunchback of Notre Dame", "Hercules", "Spawn", "AntZ", "A Bug's Life", "Star Wars: Episode I – The Phantom Menace" and the video games include “Creatures” and “The Sims”. Some of the other application areas include biological modeling, sociological modeling and military simulations. In this way, experiments and observations can be done more efficiently and less costly compared to their real-world counterparts.

1.1 Our Approach

The main purpose of this thesis study is to develop a behavioral animation system for crowd simulation. The proposed system offers the functionality to create detailed crowds composed of autonomous agents with different behavioral, cognitive, perceptive and physical characteristics. In addition, it will be possible to create and edit 3D environments which these characters will reside in. Existing systems in the literature are specific to certain situations only. We aim to develop a generic platform that enables the animator to author certain scenarios with a large number of autonomous agents that perform the given tasks.

During our thesis study, we will investigate different algorithms and try to enhance them. For instance, existing behavioral techniques are computationally inefficient ($O(n^2)$) as they require the comparison of each agent with every other agent. The cost of these methods can be reduced by some simplification methods without sacrificing realism.

1.2 Outline of Thesis Proposal

The organization of the proposal is as follows: Chapter 2 gives a literature survey on behavioral animation and crowd simulation techniques. Chapter 3 discusses the preliminary research. Finally, Chapter 4 describes the proposal for our Ph.D. thesis, giving the time schedule for this work.

Chapter 2

Related Work

Computational models are categorized into a hierarchy in the order of their appearance in computer graphics [17, 19, 18]. The earliest models were the *geometric models*. Then, forward and inverse kinematics became widely used, and thus *kinematic models* emerged. The next step was the *physical models*. They are used for animating the physical properties of particles, fluids, solids, gases and deformable solids. However, as a result of the desire to further automate the animation process, *behavioral models* emerged. Behavioral modeling involves self-animating characters that perceive environmental stimuli and give appropriate responses. The highest step in the hierarchy is *cognitive models*, through which autonomous characters can be given goals and react deliberatively as well as reactively. The modeling hierarchy can be seen in Figure 2.1. In this chapter, we explain the current state-of-the-art in behavioral models and cognitive models after giving some definitions about behavioral animation systems.

2.1 General Definitions About Behavioral Animation

Behavioral animation techniques can be classified in four ways as [49]:

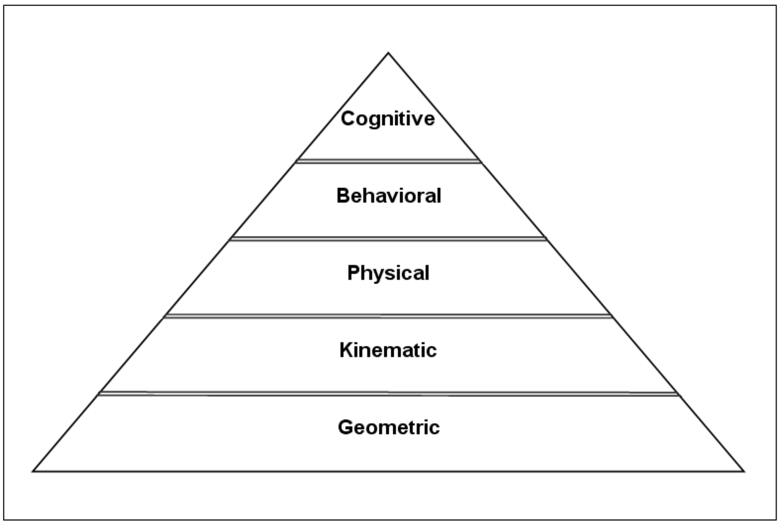

Figure 2.1: Computer graphics modeling hierarchy

1. *Specification and control methods*: Specification can be performed either declaratively or procedurally. Control can be performed either by scripting or sensing the environment.

2. *Generality of the method*: This refers to the type of animations the technique can generate. For instance, some animation techniques are specific to certain types of behaviors such as flocking.

3. *Directability*: Directabilty is the degree to which an autonomous character can be externally controlled, which can also be considered the level of autonomy. Considering directability, crowd behavior can be classified as [57]:

   - *Guided crowds*: Behaviors are explicitly defined by the users
   - *Programmed crowds*: Behaviors are programmed in a script language
   - *Autonomous crowds*: Behaviors are specified using rules or complex models

4. *Ease of authoring*: This refers to the types of primitives provided by the system, the user interface and extensibility mechanisms.

In order to realistically simulate virtual characters, we must first understand the basic properties that comprise the characteristics of these agents. A full behavioral animation system should address these issues. These properties can be summarized as follows [57]:

- *Behavior:* Response of an individual, group or species to the environment.
- *Intelligence:* The ability to learn and understand new situations.
- *Autonomy:* The quality or state of self governing.
- *Adaptation:* The ability to survive in unpredictable or dangerous environments.
- *Perception:* Awareness of the elements of the environment through physical sensation.
- *Memory:* The power or process of reproducing or recalling what has been learned and retained especially through associative mechanisms.
- *Emotion:* An affective aspect of consciousness; state of feeling.
- *Consciousness:* The quality or state of being aware especially of something within oneself or the state of being characterized by sensation, emotion, volition, and thought.
- *Freedom:* The extent that the virtual character's future behavior is unpredictable.

Autonomous agents in behavioral animation systems are classified as *situated*, *reactive*, *embodied* and *virtual* [47]. *Situated* agents are located in a virtual world shared by other entities as opposed to *isolated* agents. An agent is *reactive* if it is driven by stimulus and instinctive. On the other hand, an agent is *deliberative* if it is intellectual in the classical artificial intelligence (AI) sense. *Embodied* agents are animated in a physical manifestation such as an autonomous vehicle or a bird. Finally, the term *virtual* is used to discriminate the agents from mechanical robots, which can also be defined as situated, embodied autonomous agents.

Millar et al. classifies the components of a behavioral animation system in a generic framework as perception system, behavioral system and motor movement system [35]:

- Perception System: Perception techniques determine how an agent perceives its environment and can be classified into three as:

    1. *Zonal approach:* This approach involves surrounding the character with perception regions so that any object in this zone can be perceived by the character. The size of the detection zone is important because too small a zone will weaken the collision avoidance and path planning abilities whereas too large a zone will increase the computation time.
    2. *Sensory approach:* This approach involves placing synthetic sensors on the character. Different types of sensors for smelling, hearing, seeing etc. can be implemented. The type, location and orientation of each sensor is important for perceiving stimuli from the environment.
    3. *Synthetic vision approach:* This approach gives the character a vision of its virtual world. This approach is only useful for vision, no other stimuli will be detected. The advantage of using this method is to learn from research on human vision.

- Behavioral System: This system comprises the behavioral basis of animation and it is responsible for the decision making process. Behavior can be either solely reactive as a reflexive response to a stimulus or it can be an intelligent response driven by internal desires and experience of the character. The form of the response is also various. It can be a movement vector as well as a change in the internal attributes. In a fully-implemented system, the behavioral component includes four important modules:

    1. State variables including perception variables and mental state
    2. The rule base
    3. The memory module

4. The movement module that performs collision handling and path planning

The different approaches used in behavioral techniques can be classified as:

1. *Behavioral (Rule-based) approach:* This approach gives each character a set of rules defining how to react to the environment. It can provide reasonable behaviors in a dynamic environment and it is relatively easy to modify the rules to produce different behaviors. On the other hand, it results in less freedom, i.e., more predictability, it is specific to a particular environment and the number of rules can increase in complex environments.
2. *Network-based approach:* This approach involves creating a series of interconnecting nodes each of which describe the type of behavioral response and these nodes are created as mathematical-based procedures.
3. *Cognitive approach (Artificial intelligence):* This method uses artificial intelligence techniques such as reasoning engines and neural networks to the definition of the behavioral aspects of the animated character. These techniques provide more freedom; however, more difficult to control by the animator.
4. *Mathematical approach:* This approach defines the behavior of the characters in mathematical terms. It provides a means of specifying behavioral responses in a precise manner; however, it is not very intuitive for animators.

Our main emphasis in this proposal will be on the behavioral and cognitive approaches.

- Motor Movement System: The main functionality of this system is to propel the animated character through its virtual world. Motor movement techniques handle only the movement of the character; path planning is handled by the behavioral component. These techniques actually comprise the animation module of the behavioral animation system. The animated character will generate a movement request from its behavioral component and execute this request by using a specific motor movement approach that will be based on some sort of motion description.

2.2 Behavioral Models

Behavioral animation techniques can be categorized into three by considering the possible number of individuals to be simulated, their intelligence level, control mechanisms and collision handling methods. These approaches are particle systems, flocking systems and behavioral systems [41]. Musse et al. extend these categories by adding hierarchical systems [39], which is actually a hybrid of particle, flocking and reactive behaviors. We also include chaos systems, which is a relatively recent approach in behavioral animation techniques.

2.2.1 Particle Systems

Agent-based approaches offer several advantages such as capturing the variability of different individual characteristics and providing heterogeneity to the motion. However, agent-based methods are costly in that each agent must be handled separately, comparing its state with every other agent, thus resulting in $O(n^2)$ time complexity. Several simplifications on agent-based methods have been offered such as local methods, precomputed static plans, global planning on coarse environments and leader-follower models. However, an alternative to agent-based approaches has emerged from the fluid dynamics studies by making an analogy between the crowds and natural phenomena such as the behavior of fluids and gases. Particle systems are composed of many participants with significant dynamics. These systems are physically-based and the control is handled by force fields and global tendency [8, 9, 10]. Although these systems are used to present group and crowd simulations, the individuals in the groups do not have autonomy and heterogeneity.

In [23], Hughes introduces a model representing pedestrians as a continuous density field. The model includes an evolving potential function that guides the density field optimally towards its goal. Chenney [12] presents a technique called flow tiles, for representing and designing velocity fields, and gives application examples of crowd simulation on city streets. The most recent work, "continuum crowds", is proposed by Treuille et al. [59], introducing a real-time crowd model based on continuum dynamics. The system is only applicable to large groups with common goals, so individual differences in each group are not handled. The study of continuum crowds is inspired by Hughes, extending it from pure analytical derivations to simulation of crowds. They use a similar potential function to guide pedestrians towards their goal. In addition, it is possible to combine pedestrians into groups and introduce dynamic discomfort fields to handle geographic preferences and obstacles. The continuous equations in the mathematical model are converted into discretizations in time and space. For this purpose, the space is discretized into a regular grid and the physical variables are defined at various locations within each grid cell. The simulation examples demonstrate smooth flow under different conditions and run at interactive rates.

2.2.2 Flocking Systems

Flocking systems specify animation as distributed global motion with a local tendency. Individuals in flocking systems can seek a goal, move together and avoid collisions. The intelligence level of the individuals of flocks are higher compared to the members of particle systems. Some examples of flocking systems are given in [34, 38].

The principles of behavioral animation are based on the seminal work of Craig Reynolds, who did research on the animation of flocks of birds and schools of fish [45]. Reynolds introduces the term "boid" to refer to bird-like entities, i.e., bird-oids. These entities represent creatures like birds and fish that have flocking or schooling behavior. Each boid acts as an independent actor that maintains proper position and orientation by perceiving the local dynamic environment. The motion of each actor is defined by the laws of simulated physics and a set of programmed behaviors. The main aspect of the system is that the boids have only local information, without knowing the global environment, thus simulating the real-world perception. Each boid perceives its nearby flockmates and the obstacles within its view. The behavior of each individual in the flock is controlled by three simple rules as:

- *Collision avoidance*: Avoiding collisions with neighbors
- *Velocity matching*: Tendency to match velocity with neighbors
- *Flock centering*: Tendency to stay close to neighbors and to be near the center of the flock

These rules are sorted in the order of decreasing precedence, i.e., collision avoidance has the highest precedence and flock centering has the lowest precedence. Thus, conflicting behaviors are resolved by defining static priorities.

Reynolds extended the technique for flocking to include autonomous reactive behavior. He presents steering behaviors for obstacle avoidance [46] and path determination [47] by introducing constraints. The modeling of autonomous agents is performed in a hierarchical manner and specific emphasis is put on the middle layer of steering in [47]. The layers are:

- *Action selection*: Strategy, goals and planning
- *Steering*: Path determination
- *Locomotion*: Animation and articulation

2.2.3 Behavioral Systems

Agents in behavioral systems are more clever compared to the agents in flocking systems. The virtual agents are equipped with synthetic vision and perception of the environment and they are controlled by rules rather than local or global tendencies.

One important study in this field is the simulation of artificial fishes by Terzopoulos et al. [56]. An artificial fish is an autonomous agent that has a three-dimensional, deformable and muscle-based body that conforms with biomechanic and hydrodynamic principles. A fish also has sensors and a brain with motor perception, behavior and learning centers. There are two types of sensors, a temperature sensor that measures the water temperature and a vision sensor that has access to the geometry, material property and illumination information in the rendering pipeline and can identify nearby objects.

The behavior system of an artificial fish is based on intentions. The system runs continuously in a simulation loop and at each timestep, the intention generator issues an intention based on the habits, mental state and incoming sensory information. The habits are associated with the preferences of the fish on brightness, darkness, cold, warmth, schooling and the gender of the fish. The mental state depends on three variables, which are hunger, libido and fear. The range of each variable determines the urge to eat, mate or avoid danger. The intention generator first checks whether there is immediate collision and then these state variables in the order of fear, hunger and libido at each timestep and generates a suitable intention. If all the state variables are below a certain threshold, the generated intention will be to wander about. The intentions generated influence the behavior routines. There are eight behavior routines: *avoiding-static-obstacle*, *avoiding-fish*, *eating-food*, *mating*, *leaving*, *wandering*, *escaping*, and *schooling*. Dithering is avoided by modeling a short-term memory and persistence is ensured in order to ensure robustness in long duration behaviors such as mating or schooling. Three types of fish are modeled: *predators*, *preys* and *pacifists*.

Blumberg and Galyean [7] combine autonomy with directability. Sometimes it might be necessary to control the animated creature to some extent. In that sense, the study makes three contributions:

1. An approach to control which allows an external entity to direct a virtual character at a number of different levels.
2. A general behavioral model for perception and action selection in autonomous animated creatures which also supports external control.
3. A layered architecture which supports extensibility, reusability and multiple levels of direction.

The modeling of autonomous creatures is performed in a hierarchical manner. The levels in the hierarchy are similar to those of Reynold's [47] and organized top-down as follows:

- Behavior system
- Motor system
  - Controller
  - Motor skill
  - Degrees of freedom
- Geometric system

Geometric layer portrays the physical attributes of the character, giving its form and appearance. The more complex this layer is, the more sophisticated and expressive characters we can obtain. The second layer, motor system, executes the actions necessary to perform the goals without any knowledge from the environment. This layer acts as an interface between the geometric layer and the behavior layer, supports and provides imperative commands and minimizes the burden on the behavior layer or an external user. Degrees of freedom are used to modify the underlying geometry. Motor skills are used to produce more complicated motion such as "walking". Finally, the controller is used as an abstraction barrier between the behavior system and the underlying motor skills. It maps commands such as "forward", "turn" or "halt" into calls to turn on or turn off the appropriate motor skill. For instance, "forward" may result in the "walk" motor skill in a dog, whereas the "move" motor skill in a car. The top level is the behavior layer, which performs the decision making process given the goals and environmental information. It senses the environmental stimuli, chooses the best set of actions for the current state and sends out the necessary signals to the motor control layer.

Behaviors may range from very general to very specific and are organized into groups. External control can be added to the system by changing the motivation or sensor variables of the character or by directly scheduling tasks for execution. All constituent parts of a behavior are accessible during run-time; thus any part can be modified.

External control, i.e., directability, is a feature that has been accepted by many other researchers as well [37, 39, 2, 54]. For instance, Anderson et al. introduce constraints on the individual agents and the entire group [2]. They introduce three types of constraints as: specific agents constrained to pass through a location, the center of mass of the group constrained to a point and the members of the flock constrained to lie within a given shape at a given time. Moreover, Sung et al. define a system where users can dynamically specify the group behaviors at a certain part of the environment by attaching information to the environment [54]. They adopt a two-level scalable approach for the crowd simulation. The higher level uses a situation-based distributed control mechanism that gives each agent the rules about how to react at a specific condition based on the local environment. The lower level uses a probability scheme that computes probabilities over state transitions and then samples to move the simulation forward.

Perlin and Goldberg define a system, "Improv", based on scripts, which are sets of author-defined rules [44]. The difference of Improv from other systems is that it focuses on author's view; it provides tools to create actors that respond to users and other actors in real-time. Improv consists of two subsystems: an animation engine and a behavior engine. The animation engine uses procedural techniques to create layered, continuous, non-repetitive motions and smooth transitions between them. The behavior engine, on the other hand, enables authors to create sophisticated rules to govern the way actors communicate, change and make decisions. The animation engine represents the body of the actor whereas the behavior engine represents the mind. The behavior model of Improv is similar to that of [7] as it consists of a layered architecture.

Information about an actor and his relationship to the environment are stored in actor properties, which describe the aspects of an actor's personality. These properties are specified either when the actor is created or within a clause or script whenever a change is necessary.

2.2.4 Hybrid Systems

Hybrid systems mix particle, flocking and reactive behaviors [57]. The intelligence levels of the agents can vary from none to high in these systems. Musse and Thalmann describe a system called "ViCrowd" that is composed of a hierarchy of virtual crowds, groups and individuals, which constitute the *entities* of the simulation [39]. Individuals are virtual human agents that mimic the behaviors of real humans. Groups refer to a group of agents and crowds refer to a set of groups. Some important concepts about the simulation are *intentions*, *beliefs* and *knowledge*, which are the goals, internal status and the information about the virtual environment of the entities respectively. The intentions, beliefs and knowledge and perception determine the *crowd behavior*. The system addresses three specific problems:

1. Modeling of crowd information and hierarchical structure, also concerning its distribution among groups
2. Different levels of realism, in order to provide simple crowd behaviors, as well as complex ones
3. The required structure to provide interaction with groups of agents during the simulation in real-time

These problems are solved by considering *crowd structure* and *crowd behavior*. Crowd structure is a hierarchy composed of crowd, groups and agents, where the groups' information is distributed among the individuals. Crowd behavior deals with different levels of autonomy for the individuals. The agents can either act according to specific rules, react to specific events or can be guided by an interactive process during simulation. The different levels of autonomy has been addressed in [57] as well. This control mechanism also distinguishes hierarchical models from behavioral models.

2.2.5 Chaos Models

Modeling virtual crowds by making use of their chaotic behavior is another method in behavioral approach [22, 53, 48]. As crowds include independently moving individuals, yet exhibit general motion patterns, they can be represented by chaos models. Although these models have only a few parameters, due to the sensitivity of the system to initial conditions and non-regularity, various behaviors can be observed. These methods are superior to using random numbers to achieve variation as they are deterministic and it is difficult to create and control general patterns with random numbers. The representation of crowds is at the macro level, contrary to other micro-level approaches where the focus is on the individuals. The main argument of Saiwaki et al. [48] is that, there are few studies on the behavior of virtual humans with few parameters in contrast to the studies on the behavior of animal groups as humans demonstrate more complex behaviors.

2.3 Cognitive Models

The techniques introduced up to now are limited in the sense that they do not present any learning ability and confined to pre-specified behaviors. Moreover, they have only behavioral control, which is restricted to decision making. However, cognitive control, which involves reasoning and planning to accomplish long-term tasks is also required in order to achieve full autonomy. Behavioral learning and cognitive models have begun to be explored in computer graphics only recently [19, 36, 11, 6, 58, 13, 14, 15].

Funge introduces cognitive modeling as a further step to behavioral modeling [17, 19, 18]. He defines cognitive modeling language, CML, to specify domain knowledge with terms of actions, their preconditions and their effects and to direct the character's behavior in terms of goals. Then, the animator only specifies the sketch plan of the animation and the characters take deliberate actions through reasoning to satisfy the plan. Cognitive modeling is decomposed into two subtasks of domain knowledge specification and character direction. Domain knowledge specification is about informing the character about the environment and character direction is about instructing the character to behave in a certain way in order to achieve specific goals. CML provides a high-level interface for description of the desired goals. On the other hand, it can also serve as a traditional programming language, allowing the precise specification of how the character should act. In order to provide simple and powerful semantics for cognitive modeling, situation calculus is used. CML's syntax employs descriptive keywords with precise mappings to the underlying formal semantics of the situation calculus.

2.3.1 Models with Emotions and Personality

Some studies integrate emotions and psychological models and roles into crowd simulation systems and autonomous agents [1, 58, 43, 50, 51, 51]. For instance, Silverman et al. describe the PMFServ system [51, 52] that makes use of the psychological elements that affect human behavior. PMFServ is a highly flexible software system that can be utilized in various simulation domains. Although it provides an interface for other cognitive architectures, it is as well a fully functional standalone system to simulate human decision making based on emotions.

Allbeck and Badler give a representational basis for character believability, personality and affect [1]. For this purpose, they describe a Parameterized Action Representation (PAR) that is a representation for the actions as instructions for an agent. PAR allows an agent to act, plan and reason about its behaviors and enables the control of the agent's personality, mood and affect. PAR parameterizes the agent, relevant objects, information about paths, locations, manners and purposes. In order to perform an action, the conditions that specify the action must be satisfied. The agents that execute the action are treated as special objects with their properties stored in a hierarchical database.

Pelechano et al. incorporate psychological models into crowd simulation [43]. Their crowd simulation system deals with the wayfinding process that allows the individuals to explore and learn the internal structure of a building as well as the low-level local motion based on social forces. Thus, the agents can generate a cognitive map for navigation and find their way around an environment that they are not familiar with beforehand. The psychological component is included by using PMFServ. Communication and roles are added to achieve individualistic behaviors and spread information about the environment. Individuals have different roles and thus show heterogeneous behavior. The roles depend on two attributes of leadership and training in the existing crowd simulation system. There are trained leaders that have complete knowledge about the environment, untrained leaders and untrained non-leaders, i.e., followers. The agents are thus restricted to only three distinct roles. At this point, the psychological model provides variation through physiology, stress, perception and emotion.

Another system that involves emotions of virtual agents is presented by Tomlinson and Blumberg [58]. The study is based on social learning for interactive virtual characters, which are wild wolves. Wolves are preferred because of their social similarity to humans and their clear yet complex behaviors in a social group. The system provides a computational model that provides models of learning, emotion and development. Social learning involves the ability to have emotions, to express these emotional states and to remember an association between environmental stimuli and emotional states.

2.3.2 Learning

Learning abilities allow the virtual agents to make decisions according to their experiences by creating a cognitive map of the environment. Most of the systems in the literature use reinforcement learning; thus we will briefly overview the terms and definitions related with this type of learning.

Reinforcement learning is an unsupervised learning technique which can be defined as learning from experience in the absence of a teacher [6]. In this learning technique, the world is taken to be in one of a set of perceivable states. The goal of reinforcement learning is to learn an optimal sequence of actions to take the agent from an arbitrary state to the goal state. The main approach is to probabilistically explore states, actions and their outcomes to learn how to act in a given situation. *State* refers to a specific configuration of the world. The set of all represented configurations of the world is called the *state space*. An agent can change the state of the world by performing an *action*. Each agent is assumed to have a finite set of actions and it can perform only one at a time. A *state-action* pair, $< S/A >$, is a relationship between a state S and an action A. It is typically related with a numerical value like *future expected reward*, which gives the value of performing an action A in a given state S. A *policy* represents the probability with which the agent selects an action at a specific state. When the agent reaches a goal state, it receives a *reward* or *reinforcement*.

The most popular reinforcement learning technique is *Q-Learning* [60]. In Q-Learning, state-action space is stored in a lookup table. Each row represents a state and each column represents an action in the table. An entry in the table represents the *Q-value* of a given state-action pair with respect to getting a reward. The optimal value for each state-action pair can be learned by exhaustive search of the state-action pairs and by a local update rule to reflect the consequences of taking a given action in a given state with respect to achieving the goal state.

An important learning example is given by Blumberg et al. [6], where an autonomous virtual dog is interactively taught to perform a desired behavior. The system employs of reinforcement learning along with learning inspired from animal training, i.e., clicker training. The virtual dog mimics the behavior of a real dog by performing the best action in a given context, assessing the relative reliability of its actions in producing a reward and altering its choice of action accordingly.

Another system that uses reinforcement learning is described by Conde et. al [13]. The system is interesting as it does not use reinforcement learning in its classical approach but as a behavioral engine for exploration, learning and visiting the virtual environment. Thus, the interest is in the learning process itself rather than the optimization of learning. The system makes use of situated AI, which involves adaptive artificial systems evolving in an environment that is not entirely predictable. The autonomous and intelligent agents react to their environment by making decisions based on their perception, memory and logic. Intelligence accounts for the ability to make plans and carry out tasks based on the actual state of the virtual environment. Autonomy refers to the agent's capacity to visit and memorize the given virtual environment without any external intervention.

Conde and Thalmann introduce a new low-level learning technique as an alternative to classical Q-learning [15]. The proposed method uses a tree search algorithm with inverse reinforcement learning. The system's objective is to allow the virtual agent to explore an unknown virtual environment and to build structures in the form of cognitive models or maps. Then, the virtual agent can dissipate this information to other agents. Learning through observation of an expert agent is similar to imitation and called apprenticeship learning. The steps of the learning process are as follows:

1. First, a tree search algorithm A* is used to observe the state sequences generated by the user (expert).
2. Q-decomposition approach that uses all pseudo value function components (vision, avoidance and navigation) is integrated.
3. Apprenticeship learning via inverse reinforcement learning is adapted to the behavioral animation.

2.3.3 Motion and Path Planning for Crowds

In artificial intelligence, planning is related with searching for a sequence of logical operators or actions that transform an initial world state into a desired goal state [31]. Motion planning and path planning problems arise in fields such as robotics, assembly analysis, virtual prototyping, manufacturing and computer animation, but the origin of the problem is in robotics. The main purpose for the object is to plan its own motion. In order to plan a motion, the object must have some knowledge about the environment and find a collision-free path among the obstacles in the environment [16]. The path should be preferably short. A classical motion planning problem is known as the *Piano Mover's Problem*, which is about moving a piano one room of a house to another without hitting the static obstacles [31].

A detailed survey on motion planning is given by Latombe [30], Overmars [40] and Baños et al. [20]. Motion planning for crowd simulation has been studied by many researchers [9, 3, 4, 5, 25, 29, 26, 28, 42].

Motion planning approaches can be classified in three groups as [40, 42] *potential fields*, *cell decomposition methods* and *roadmap methods.*

2.3.3.1 Potential Fields

Potential fields put repulsive powers on the obstacles in the environment and attractive powers on the agent's destination. Thus, the object tries to move in the direction of the goal while being pushed away by obstacles. Due to the use of local properties only, the object may move in the wrong direction, resulting in a deadlock situation; getting trapped in local minima. This approach was first introduced by Khatib [27].

2.3.3.2 Cell Decomposition Methods

Cell decomposition methods divide the free space into a number of discrete cells. These methods either use approximate decomposition [24], using grids or quadtrees or exact decomposition, using convex cells to cover the entire free space. Convex cells provide constant time to compute a path between any two configurations within a cell.

These algorithms are easy to implement; however, they are ineffective if the resolution is low. Moreover, when the dimension of the configuration space gets higher or when the complexity of the scene is very large, the number of cells required increases too much to be practical.

2.3.3.3 Roadmap Methods

Roadmaps discretize the navigation space in a network of paths made up of lines and curves along which the object can move free of collisions [55]. The roadmap can be considered a graph and thus the problem is reduced to graph searching. The difficulty of these methods is to compute an effective roadmap.

Chapter 3

Preliminary Work

So far, we have investigated several aspects of crowd simulation and behavioral animation. As an initial step towards creating a full system, we have developed two different crowd simulation applications each with a different behavioral animation technique. The first one is based on Reynolds' behavioral flock model. We have extended the model to include predator and prey flocks. The second example is the continuum model for crowds based on the study of Treuille et al. [59]. We propose to extend this model to non-uniform grids in order to increase the time performance.

3.1 Flock Simulation

Reynolds' boid model has provided new directions for the behavioral animation research. Various extensions of this model such as constrained flocks [2], leader-follower behavior [21] and fuzzy models [32] have been proposed. We extend these studies by proposing a predator/prey model. The characteristics of predator and prey birds have been investigated by the biologists [33].

In this section, we describe the implementation of the prey and predator boids, which are based on [45, 21] and [33]. The flock of preys is a set of boids b_i for $i = 1 \ldots n$. The boids are distributed in the environment according to the chaos rules, i.e., the initial position of each boid is affected by the position of the others. In this way, it is possible to obtain different distributions in the space just by modifying the initial boid's position. The position distribution is as follows:

$$\overrightarrow{x_{i+1}} = R\overrightarrow{x_i} + (1 - R\overrightarrow{x_i}), i = 1 \ldots n$$

where $\overrightarrow{x_i}$ is the position of b_i and R is a constant in the range $2.3 < R \leq 3$. The positions are later normalized so as to confine the boids to the environmental borders. Each boid has a visibility sphere V_i of radius r, which is defined as:

$$V_i = \{b_j, |b_i - b_j| < r, j = 1 \ldots m\} \tag{3.1}$$

The steering forces acting on each prey b_i are:

$$m_i \overrightarrow{v}_i d_t = \overrightarrow{f_{sep_i}} + \overrightarrow{f_{coh_i}} + \overrightarrow{f_{al_i}} + \overrightarrow{f_{av_i}} \tag{3.2}$$

$$\overrightarrow{v}_i = \overrightarrow{x}_i d_t \tag{3.3}$$

$$\overrightarrow{f_{sep_i}} = c_{sep} \sum_{\forall b_j \in V_i} (\overrightarrow{x_i} - \overrightarrow{x_j}) \tag{3.4}$$

$$\overrightarrow{f_{coh_i}} = c_{coh} \sum_{\forall b_j \in V_i} \frac{\overrightarrow{x_j}}{|V_i|} \tag{3.5}$$

$$\overrightarrow{f_{al_i}} = c_{al} \sum_{\forall b_j \in V_i} \frac{\overrightarrow{v_j}}{|V_i|} \tag{3.6}$$

$$\tag{3.7}$$

where $\overrightarrow{v_i}$ is the velocity and $\overrightarrow{x_i}$ is the position of b_i. The forces acting on b_i are, $\overrightarrow{f_{sep_i}}$, $\overrightarrow{f_{coh_i}}$ and $\overrightarrow{f_{al_i}}$, which are the separation, cohesion and alignment forces respectively. Given that a predator p of scaled size ω and a position of $\overrightarrow{x_p}$, the avoidance force $\overrightarrow{f_{al_i}}$ of prey b_i from the predator p is computed as follows:

$$\overrightarrow{f_{av_i}} = c_{av} \frac{\overrightarrow{x_{p,i}}}{1 + \exp(\omega(|\overrightarrow{x_{p,i}}| - r))} \tag{3.8}$$

$$\tag{3.9}$$

where $\overrightarrow{x_{p,i}}$ is the vector from p to b_i and r is the visibility radius of the prey. The model ensures that the prey individuals run away from the predator when the predator is in their visibility sphere. The scaled predator size ω determines the degree fallof for the avoidance force.

The behavior of predators is handled differently. Predators do not tend to form flocks; their only tendency is to catch preys. The governing equations for the control of the movement of a predator p are:

$$m_p \overrightarrow{v}_p d_t = \overrightarrow{f_{at_i}} \tag{3.10}$$

$$\overrightarrow{x_{p_{tar}}} = argmin(\overrightarrow{x_p} - \overrightarrow{x_i}), i = 1 \ldots |V_p| \tag{3.11}$$

$$\overrightarrow{v_{p_{des}}} = \frac{\overrightarrow{x_{p_{tar}}} - \overrightarrow{x_p}}{|\overrightarrow{x_{p_{tar}}} - \overrightarrow{x_p}|} \tag{3.12}$$

$$\overrightarrow{f_{at_p}} = c_{at} \frac{\overrightarrow{v_{p_{des}}} - \overrightarrow{v_p}}{|\overrightarrow{v_{p_{des}}} - \overrightarrow{v_p}|} \tag{3.13}$$

$$\tag{3.14}$$

where $\overrightarrow{x_{p_{tar}}}$ is the target position, which is the closest prey in the visibility sphere of the predator, $\overrightarrow{v_{p_{des}}}$ is the desired velocity and $\overrightarrow{f_{at_p}}$ is the attack force of the predator.

In addition to the boids, the environment can contain solid obstacles. Thus, collision handling by position and velocity correction is performed. The direction of the velocity is inverted, which results in the bouncing of the boids from solid objects in the environment. As a future work, a more intelligent collision avoidance scheme could be adopted. The system design for the predator/prey model is given in Figure 3.1.

3.2 Continuum Crowds

As a second application, we have implemented a crowd simulation system based on continuum dynamics. The crowd is represented as a particle system, which moves by per-particle energy minimization. The crowd is composed of different groups of individuals, where each group has a common goal shared by all

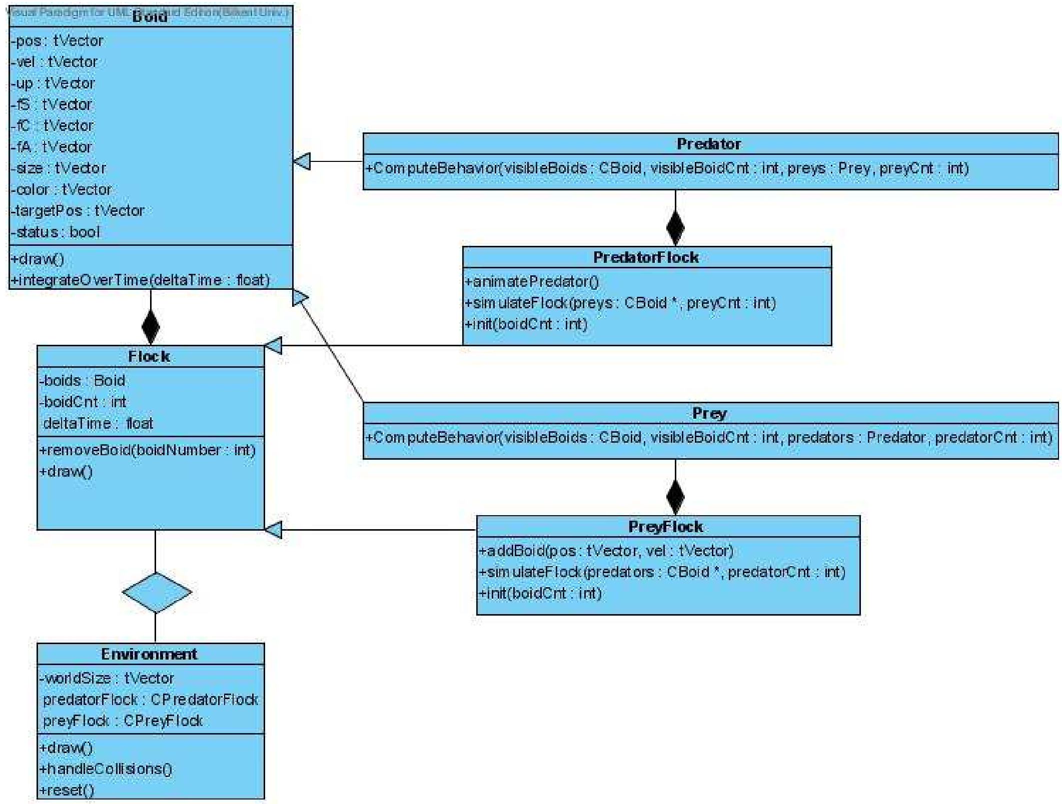

Figure 3.1: System design for flocks of predators and preys.

its members. Time and space discretization is performed in order to convert the continuous equations into simulations. At each time step, a global potential field that describes the optimal paths around obstacles to goal is created. Then, the positions of individuals are updated accordingly. The mathematical model is explained comprehensively in [59], so we will not give the details here. However the implementation is as follows:

For each timestep:

* Convert the crowd to a density field.

  For each group:

  * Construct the unit cost field.
  * Construct the potential and its gradient.
  * Update people's locations.

A crowd of 100 individuals avoiding obstacles can be seen in Figure 3.2. The target position is the red dot behind the obstacle trees. Figure 3.3 depicts two groups crossing through each other in order to reach their goals. The goal position for the group on the left is on the righthand side and the goal position for the group on the right is on the lefthand side. Our system enables the distribution of individuals to selected cells, defining the discomfort areas and goal positions and modifying the grid resolution.

Currently, we are working on the extension of this model to adaptive non-uniform grids. In this way, instead of creating a regular grid, we create a quadtree structure. The leaves of the quadtree are the cells of the grid and the structure is updated at each time step. The space is recursively decomposed into blocks. Thus, the cells are either splitted or merged depending on the number of humans in each cell. One of the problems associated with quadtrees is finding the neighbors of a cell. In the regular grid structures, neighbor finding is a constant time operation. However, in quadtrees, the operation takes $O(d+1)$ time, where d is the depth of the tree. However, this drawback is overcome by the overall efficiency of the method. Suppose that the regular grid structure is a square of N x N cells. The time complexity for the regular structure is then $O(N^2)$. On the other hand, when the grid is represented as a quadtree, the time complexity would be $O(log(N))$. This can be derived from the theorem, which states that the depth of a quadtree for a set P of points in the plane is at most $log(s/c) + 3/2$, where c is the smallest distance between any two points in P and s is the side length of the initial square that contains P. In our case, $s = N$ and $c = 1$. Therefore, the speedup will be substantial. In addition, the smoothness of the simulation depends on the resolution of the grid. The adaptive scheme will enable sparse data points in areas with few people and finer discretizations in more crowded areas.

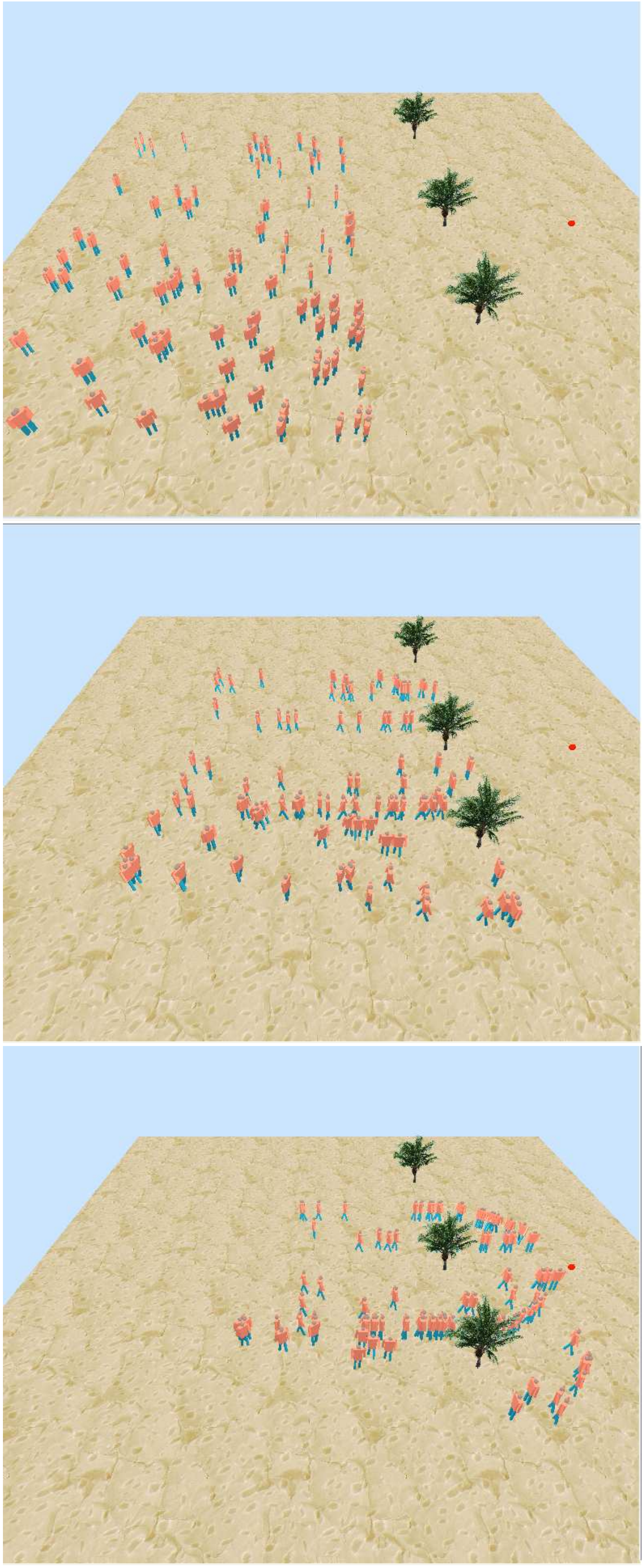

Figure 3.2: Crowd of 100 humans avoiding obstacles. The red dot is the goal position.

Figure 3.3: Two intersecting groups while approaching their goals.

Chapter 4

Proposal

In this chapter, we give our proposal for the thesis, implying the open points in the previous work, and outline a framework for the behavioral animation system for crowd simulation.

4.1 Contributions

The main purpose of this study is to develop a framework to represent a multi-agent behavioral animation system. The generic framework is inspired by the studies of Millar et al. [35]. Existing systems in the literature are specific to certain situations only. Our aim is to develop a generic system that handles different aspects of crowd simulation and provides the animator with the ability to author different scenarios. The system framework is given in the next section.

The problems associated with behavioral animation can be summarized as follows [47]:

- Deciding on the right action is complicated due to factors such as incomplete or erroneous perception.
- There may be competing and contradicting goals, which can lead to

dithering.

- An important goal may be unattainable and pursuing such a goal might prevent the accomplishment of lower-priority but attainable goals.

Thus, a system that overcomes such difficulties must be created. For instance, a complex perception system that is specific to different types of individuals must be created in order to overcome the difficulties caused by erroneous perception. Furthermore, a rule-base that defines various behaviors for tasks with different priorities must be built to handle goal-specific problems.

In addition to these problems, the existing systems are either computationally inefficient or lacking of capturing the individual differences amongst the individuals of the crowd. Agent-based approaches have ($O(n^2)$) time complexity as they require each agent to be compared against every other agent in order to handle interactions among agents, such as collisions. Simplifications to increase the time performance will be investigated. Methods based on continuum dynamics are computationally efficient; however, they can only represent the crowd behavior at the macro level. A hybrid approach that combines the best aspects of each technique can be explored.

4.2 Framework for the System

A framework for a multi-agent behavioral animation system is composed of the following components:

- *The environment*: This system involves the creation of the virtual world that the characters reside in. The environment includes all the objects except the autonomous agents in the animated scene.
- *The Perception System*: The animated characters use their senses to perceive the environment they are living in. Thus, a perception system needs to be modeled. The perception system will consist of one or more perception techniques. It will include the necessary senses such as vision, hearing, smelling, etc.

- *The Behavioral System*: Information gathered via the perception system is passed onto the behavioral system. This system consists of the following modules:
  - *State variables*: The state variables either store information from the perception system or internal status of the agent. The internal status refers to the state of the character within the environment as well as the current psychological status of the character such as the degree of panic or level of fear.
  - *The rule base*: The rule base consists of the user-defined rules that specify how the character reacts to external stimuli. The rules are defined according to the state variables. Each rule has a priority level and the rule with the highest priority is executed first. The goal of the executed rule is placed in the memory module.
  - *The memory module*: This module gives the character the ability to recall the activities in its goal queue and the objects in its environment.
  - *The movement module*: This module handles collisions and planning for the animated character and the output is sent to the motor control system. Efficient path planning techniques will be developed in the scope of this module.
- *The Motor Control System*: This system involves the movement of the characters; all the animation is performed here. The operation of this system depends on the type of the autonomous characters, which are virtual humans in our case. The realism of the animation depends on the complexity of this system.

4.3 Work Plan

The proposed time schedule can be seen in Table 4.1. If everything goes as planned, the Ph.D. study will be finished in June, 2008.

Table 4.1: Proposed time schedule for the Ph.D. thesis.

Number	Start–End Dates	Steps to Perform
1	Sep, 2006–April, 2007	Literature survey, examining various crowd simulation techniques and implementation of different algorithms.
2	May, 2007–Jul, 2007	Creation of the scene database, the environmental system, simple behavior and motor control systems and the user interface.
3	Aug, 2007–Dec, 2007	Creation of the behavioral system.
4	Jan, 2008–Jun, 2008	Creation of a more complex motor control system and character model.

Bibliography

[1] J. Allbeck and N. Badler. Toward representing agent behaviors modified by personality and emotion. In *Proceedings of Embodied Conversational Agents at AAMAS'02*, Bologna, Italy, July 2002.

[2] M. Anderson, E. McDaniel, and S. Chenney. Constrained animation of flocks. In D. Breen and M. Lin, editors, *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pages 286–297, 2003.

[3] O. Arikan, S. Chenney, and D. A. Forsyth. Efficient multi-agent path planning. In *Proceedings of the Eurographic workshop on Computer animation and simulation*, pages 151–162, 2001.

[4] O. Bayazit, J. Lien, and N. Amato. Better group behaviors in complex environments with global roadmaps. In *Proceedings of Int. Conf. on the Sim. and Syn. of Living Sys. (Alife)*, pages 362–370, 2002.

[5] O. Bayazit, J. Lien, and N. Amato. Better group behaviors using rule-based roadmaps. In *Proceedings of International Workshop on Algorithmic Foundations of Robotics (WAFR)*, Nice, France, 2002.

[6] B. M. Blumberg, M. Downie, Y. Ivanov, M. Berlin, M. Johnson, and B. Tomlinson. Integrated learning for interactive synthetic characters. *ACM Computer Graphics (Proceedings of SIGGRAPH'02)*, pages 417–426, 2002.

[7] B. M. Blumberg and T. A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. *ACM Computer Graphics (Proceedings of SIGGRAPH'95)*, pages 47–54, 1995.

[8] E. Bouvier, E. Cohen, , and L. Najman. From crowd simulation to airbag deployment: particle systems, a new paradigm of simulation. *Journal of Electronic Imaging*, 6(1):94–107, 1997.

[9] D. C. Brogan and J. K. Hodgins. Group behaviors for systems with significant dynamics. *Autonomous Robots*, 4:137–153, 1997.

[10] D. C. Brogan, R. A. Metoyer, and J. K. Hodgins. Dynamically simulated characters in virtual environments. *IEEE Comput. Graph. Appl.*, 18(5):58–69, 1998.

[11] R. Burke, D. Isla, M. Downie, Y. Ivanov, and B. Blumberg. Creature smarts: The art and architecture of a virtual brain. In *Proceedings of the Computer Game Developers Conference*, pages 147–166, 2001.

[12] S. Chenney. Flow tiles. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pages 233–242, 2004.

[13] T. Conde, W. Tambellini, and D. Thalmann. Behavioral animation for autonomous virtual agents helped by reinforcement learning. In T. Rist, editor, *Proceedings of IVA'03*, pages 175–180, Berlin, Heidelberg, 2003.

[14] T. Conde and D. Thalmann. Autonomous virtual agents learning a cognitive model and evolving. In *Lecture Notes in Computer Science*, volume 3631, pages 88–98. Springer-Verlag, Berlin, 2005.

[15] T. Conde and D. Thalmann. Learnable behavioral model for autonomous virtual agents: Low-level learning. In *Proceedings of Embodied Conversational Agents at AAMAS'06*, pages 89–96, Hakodate, Hokkaido, Japan, May 8-12 2006.

[16] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin Heidelberg New York, second edition, 1998.

[17] J. Funge. *Making Them Behave: Cognitive Models for Computer Animation*. PhD thesis, Graduate Department of Computer Science, University of Toronto, 1998.

[18] J. Funge. Cognitive modeling for games and animation. *Communications of the ACM*, 43(7):41–48, 2000.

[19] J. Funge, X. Tu, and D. Terzopoulos. Cognitive modeling: Knowledge, reasoning and planning for intelligent characters. *ACM Computer Graphics (Proceedings of SIGGRAPH'99)*, pages 29–38, 1999.

[20] H. H. González-Baños, D. Hsu, and J. Latombe. Motion planning: recent developments. In S. Ge and F. Lewis, editors, *Autonomous Mobile Robots: Sensing, Control, Decision-Making and Applications*. CRC Press, 2006.

[21] C. Hartman and B. Beneš. Autonomous boids. *Computer Animation and Virtual Worlds*, 17(3):199–206, 2006.

[22] Y. Hijikata, T. Komatsu, N. Saiwaki, and S. Nishida. Automatic generation of moving crowd using chaos and electric charge model. In *Proceedings of the IEEE International Conference on Systems,Man and Cybernetics*, volume 2, pages 342–347, 2002.

[23] R. L. Hughes. The flow of human crowds. *Annual Review of Fluid Mechanics*, 35:169–182, 2003.

[24] J. James J. Kuffner. Goal-directed navigation for animated characters using real-time path planning and control. In *Proceedings of CAPTECH'98*, November 1998.

[25] A. Kamphuis and M. H. Overmars. Finding paths for coherent groups using clearance. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, 2004.

[26] A. Kamphuis, M. Rook, and M. H. Overmars. Tactical path finding in urban environments. In *Proceedings of First International Workshop on Crowd Simulation (V-CROWDS'05)*, Lausanne, Switzerland, 2005.

[27] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

[28] Y.-C. Lai, S. Chenney, and S. Fan. Group motion graphs. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, volume 23, 2005.

[29] M. S. A. Latif and S. Widyarto. The crowd simulation for interactive virtual environments. In *ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Appications in Industry (VRCAI'04)*, pages 278–281, Singapore, 2004.

[30] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[31] S. M. LaValle. *Planning algorithms*. Cambridge University Press, 2006.

[32] I. Lebar Bajec, N. Zimic, and M. Mraz. Fuzzifying the thoughts of animats. In T. Bilgiç, B. D. Baets, and O. Kaynak, editors, *Proceedings of of IFSA 2003*, volume 2715 of *Lecture Notes in Artificial Intelligence*, pages 195–202, Berlin, 2003. Springer-Verlag.

[33] S.-H. Lee, H. Pak, and T.-S. Chon. Dynamics of prey-flock escaping behavior in response to predator's attack. *Journal of Theoretical Biology*, 240(2):250–259, 2006.

[34] M. J. Mataric. Learning to behave socially. In D. Cliff, P. Husbands, J. Meyer, and S. Wilson, editors, *From Animals to Animats: International Conference on Simulation of Adaptive Behavior*, pages 453–462, 1994.

[35] R. Millar, J. Hanna, and S. Kealy. A review of behavioural animation. *Computers and Graphics*, 23(1):127–143, 1999.

[36] J.-S. Monzani, A. Caicedo, and D. Thalmann. Integrating behavioural animation techniques. In A. Chalmers and T. Rhyne, editors, *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, volume 20, 2001.

[37] S. R. Musse, C. Babski, T. Capin, and D. Thalmann. Crowd modelling in collaborative virtual environments. In *Proceedings of VRST'98*, pages 115–124, Taipei, Taiwan, 1998.

[38] S. R. Musse and D. Thalmann. Model of human behavior: Group interrelationship and collision detection analysis. In *Proceedings of Workshop of Computer Animation and Simulation of Eurographics'97*, Budapest, Hungary, 1997.

[39] S. R. Musse and D. Thalmann. Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics*, pages 152–164, 2001.

[40] M. H. Overmars. Recent developments in motion planning. In *International Conference on Computational Science*, volume 3, pages 3–13, 2002.

[41] R. Parent. *Computer Animation: Algorithms and Techniques.* Morgan Kaufmann, 2001.

[42] S. Paris, S. Donikian, and N. Bonvalet. Environmental abstraction and path planning techniques for realistic crowd simulation. *Computer Animation and Virtual Worlds*, 17:325–335, 2006.

[43] N. Pelechano, K. O'Brien, B. Silverman, and N. Badler. Crowd simulation incorporating agent psychological models, roles and communication. In *Proceedings of First International Workshop on Crowd Simulation (V-CROWDS'05)*, Lausanne, Switzerland, 2005.

[44] K. Perlin and A. Goldberg. Improv: A system for scripting interactive actors in virtual worlds. *Computer Graphics*, 29(3), 1996.

[45] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *Computer Graphics*, 21(4), 1987.

[46] C. W. Reynolds. Not bumping into things. *ACM SIGGRAPH'88 Course Notes, #27, Developments in Physically-based Modeling*, pages G1–G13, 1988.

[47] C. W. Reynolds. Steering behaviors for autonomos characters. In *Proceedings of Game Developers Conference*, pages 763–782, San Jose, California, 1999.

[48] N. Saiwaki, T. Komatsu, T. Yoshida, S., and Nishida. Automatic generation of moving crowd using chaos model. In *Proceedings of the IEEE International Conference on System, Man and Cybernetics*, volume 4, pages 3715–3721, 1997.

[49] S. Shanbhag. Behavioral animation : A report. In *Proceedings of the Inter Research Institute Student Seminar in Computer Science (IRISS'02)*, 2002.

[50] B. G. Silverman. More realistic human behavior models for agents in virtual worlds: Emotion, stress and value ontologies. Technical report, Systems Engineering Department, University of Pennsylvania, 2001.

[51] B. G. Silverman, G. Bharathy, K. O'Brien, and J. Cornwell. Human behavior models for agents in simulators and games: Part I-enabling science with PMFserv. *Presence: Teleoperators. and Virtual Environments*, 15(2), 2006.

[52] B. G. Silverman, G. Bharathy, K. O'Brien, and J. Cornwell. Human behavior models for agents in simulators and games: Part II-gamebot engineering with PMFserv. *Presence: Teleoperators. and Virtual Environments*, 15(2):163–185, 2006.

[53] C. Soh, P. Raveendran, and Z. Taha. Automatic generation of self-organized virtual crowd using chaotic perturbation. In *Proceedings of TENCON 2004, 2004 IEEE Region 10 Conference*, volume 2, pages 124–127, 2004.

[54] M. Sung, M. Gleicher, and S. Chenney. Scalable behaviors for crowd simulation. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, volume 23, pages 519–528, 2004.

[55] M. Sung, L. Kovar, and M. Gleicher. Fast and accurate goal-directed motion synthesis for crowds. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pages 291–300, 2005.

[56] D. Terzopoulos, X. Tu, and R. Grzeszczuk. Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artificial Life*, 1(4):327–351, 1994.

[57] D. Thalmann, S. R. Musse, and M. Kallmann. Virtual humans' behaviour: Individuals, groups, and crowds. In *Proceedings of Digital Media Futures*, 1999.

[58] B. Tomlinson and B. Blumberg. Alphawolf: Social learning, emotion and development in autonomous virtual agents. In *Proceedings of First GSFC/JPL Workshop on Radical Agent Concepts*, pages 35–45, 2002.

[59] A. Treuille, S. Cooper, and Z. Popovic. Continuum crowds. *ACM Transactions on Graphics (SIGGRAPH '06)*, 25(3):1160 – 1168, 2006.

[60] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.