Crowd Simulation

Daniel Thalmann and Soraia Raupp Musse

Crowd Simulation

With contributions from: Adriana Braun, André Tavares da Silva, Barbara Yersin, Branislav Ulicny, Cláudio Rosito Jung, Daniel Paiva, Fernando Santos Osório, Júlio Cesar Jacques Júnior, Jonathan Maïm, Julien Pettré, Leandro Motta Barros, Luiz Gonzaga da Silveira Júnior, Pablo de Heras Ciechomski, Renata Vieira, Ronan Boulic.

Springer

Daniel Thalmann
EPFL VRlab
Station 14
CH 1015 Lausanne, Switzerland
daniel.thalmann@epfl.ch

Soraia Raupp Musse
Graduate Course in Computer Science
PUCRS
6681, Av. Ipiranga, 6681 - Builing 32
Porto Alegre 90619-900, Brazil
soraia.musse@pucrs.br

British Library Cataloguing in Publishing Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2007927755

ISBN 978-1-84628-824-1 e-ISBN 978-1-84628-825-8

Printed on acid-free paper.

9 8 7 6 5 4 3 2 1

Springer Science+Business Media
springer.com

(Daniel)
To my beloved wife Nadia and beloved daughters Melanie,
Vanessa, and Sabrina

(Soraia)
To my daughter Marina, parents Iara (in loving memory) and
Eduardo, who made me understand how simple things can be.

Preface

This book surveys algorithms and techniques of crowd simulation, and is intended for graduate students, researchers, and professionals. In particular, computer animation researchers, developers, designers, and urban planners will greatly benefit from this book.

In 1996, when the authors started researching into crowd simulation, there was very little material available on crowd simulation in the Computer Science literature. Daniel Thalmann supervised Soraia Raupp Musse PhD thesis in 1997 and since then they have jointly published more than 40 papers in the domain. There has since been significant research in this area and many techniques have been developed, with the entertainment industry in particular realising the potential of crowd animation. As a result, many other research groups have also started working in the area. As early pioneers in this research, the authors organized the first workshop on Crowd Simulation (V-Crowds) in 2005 in Lausanne. But why is this subject so fascinating?

Aggregated motion is both *beautiful* and *complex* to contemplate. *Beautiful* due to the synchronisation, homogeneity and unity described in this type of motion, and *complex* because there are many parameters to be handled in order to provide these characteristics. History shows that there has always been interest in understanding and controlling the motion and behaviour of crowds of people. Psychologists and sociologists have studied the behaviours of groups of people for several years, primarily to study the effects that occur when people with the same goal become one entity – a crowd or a mass. When this happens, people can lose their individuality and adopt the behaviour of the crowd entity, behaving in a different way than if they were alone.

Certain problems arise only when studying crowds. For instance, collision avoidance problems related to a large number of individuals in the same place require different strategies in comparison with the methods used to avoid collision between individuals. Also, motion planning used in a group that walks together requires more information than that needed to implement individual motion planning. The trajectories computed for agents who are part of the same group and walk together with similar speeds have to be different even when they share the same environment and goals.

Moreover, a crowd is not only a large group of individuals, but can also be formed by groups which in turn are related to individuals. In addition other levels of behaviour can exist when treating crowds in this hierarchical structure. The group behaviours can be used to specify the way a group moves, behaves and acts in order to fit different group structures (flocking, following, repulsion, attraction, etc). Individual abilities can also be required in order to improve the autonomy and intelligence of crowds, for instance perception, emotional status, memory, communication, etc. However, when we consider thousands of individuals, these complex behaviours cannot be provided individually due to the hardware constraints and to computational time rates. A further problem relates to how to improve the intelligence and provide autonomy to scalable crowds, in real-time systems.

The simulation of large crowds in real time requires many instances of similar characters. We need algorithms to allow for each individual in the crowd to be unique. In this book we explain two methods: first, a simple and efficient way of attaching accessories to individuals in order to modify their look; and secondly, a new and generic technique based on segmentation maps to add detailed color variety and patterns to human meshes as well as to accessories. Both methods are scalable to suit all human levels of detail exploited in crowd simulations.

Depending on the application of crowds, other requirements may be needed. For instance real time simulations can be required in order to populate virtual environments in virtual reality systems. In order to provide a tool to simulate behavioural aspects of a crowd, social conventions of interrelationships are needed. Yet, accurate validations should be provided to simulate safety systems.

Some crowd requirements along with strategies and techniques that can be adopted to deal with these, are described in this book. Some of the topics presented are related to population modelling, virtual human animation, computer vision techniques focusing on crowd control and crowd rendering, and some applications are analysed.

Contents

1

Introduction

Although collective behavior has been studied since as early as the end of the nineteenth century [LeB95], attempts to simulate it by computer models are quite recent, with most of the works done only in the mid and late nineties. In the past two decades researchers from a broad range of fields such as architecture [SOHTG99,PT01,TP02], computer graphics [Rey87,HB94,BG96,MT01, TLC02b, UT02, BMB03], physics [HM95, HFV00, FHV02], robotics [MS01], safety science [TM95a,Sti00,Sim04], training systems [Bot95,Wil95,VSMA98], and sociology [MPT92, TSM99, JPvdS01] have been creating simulations involving collections of individuals. Nevertheless, despite the apparent breadth of the crowd simulation research basis, interdisciplinary exchange of ideas is rare; researchers in one field are usually not very aware of works done in other fields.

Most approaches were application-specific, focusing on different aspects of the collective behavior, using different modeling techniques. Employed techniques range from those that do not distinguish individuals such as flow and network models in some of the evacuation simulations [TS92], to those that represent each individual as being controlled by more or less complex rules based on physical laws [HIK96,HFV00], chaos equations [SKN98], behavioral models in training systems [Wil95], or sociological simulations [JPvdS01].

We can distinguish two broader areas of crowd simulations. The first one is focusing on a **realism of behavioral aspects** with usually simple 2D visualizations like evacuation simulators, sociological crowd models, or crowd dynamics models. In this area, a simulated behavior is usually from a very narrow, controlled range (for example, people just flying to exit or people forming ring crowd structures) with efforts to quantitatively validate correspondence of results to real-world observations of particular situations [TM95b]. Ideally, a simulation's results would then be consistent with data sets collected from field observations or video footage of real crowds either by human observers [SM99] or by some automated image processing method [MVCL98, CYC99]. Visualization is used to help understand simulation results, but it is not crucial. In most cases, a schematic representation, with crowd members represented by colored dots, or sticky figures, is enough, sometimes even preferable as it allows highlighting important information.

In the second area, a main goal is **high-quality visualization** (for example, in movie productions and computer games), but usually the realism of the behavior model is not the priority. What is important is a convincing visual result, which is achieved partly by behavior models, partly by human intervention in the production process. A virtual crowd should both look good and be animated in a believable manner, the emphasis of the research being mostly on rendering and animation methods. Crowd members are visualized as fully animated three-dimensional figures that are textured and lit to fit into the environment [DHOO05a]. Here, behavior models do not necessarily aim to match quantitatively the real world; their purpose is more to alleviate the work of human animators, and to be able to respond to inputs in case of interactive applications.

Nevertheless, a recent trend seems to be a **convergence of both areas**, where visualization-oriented systems are trying to incorporate better behavior models to ease creation of convincing animations and behavior-oriented models are trying to achieve better visualization, especially in the domain of evacuation simulators. We can expect that the most demanding applications would be training systems, where both valid replication of the behaviors and high-quality visualization are necessary for a training to be effective.

1.1 Requirements and Constraints for Crowd Modeling

Real-time crowds bring different challenges compared with the systems either involving a small number of interacting characters (for example, the majority of contemporary computer games), or non-real-time applications (as crowds in movies, or visualizations of crowd evacuations after off-line model computations). In comparison with single-agent simulations, the main conceptual difference is the **need for efficient variety management** at every level, whether it is visualization, motion control, animation, or sound rendering. As everyday experiences hint, virtual humans composing a crowd should look different, move different, react different, sound different, and so forth. Even if assuming perfect simulation of a single virtual human would be possible, creating a simulation involving multiple such humans would still be a difficult and tedious task. Methods easing control of many characters are needed; however, such methods should still preserve the ability to control individual agents.

In comparison with non-real-time simulations, the main technical challenge is **increased demand on computational resources** whether it is general processing power, graphics performance, or memory space. One of the foremost constraining factors for real-time crowd simulations is crowd rendering. Fast and scalable methods both to compute behavior, able to take into account inputs not known in advance, and to render large and varied crowds, are needed. While non-real-time simulations are able to take advantage of knowing a full run of the simulated scenario (and therefore, for example, can run iteratively over several possible options selecting the globally best solution), real-time simulations have to react to the situation as it unfolds in the moment.

1.2 Crowd Simulation Areas

In order to create a full simulation of the crowd in the virtual environment, many issues have to be solved. The areas of relevance for crowd simulation and some associated questions include:

Generation of virtual individuals: How to generate a heterogeneous crowd? How to create a population with desired distribution of features [GKMT01, SYCGMT02, BBOM03]? Chapter 3 discusses some of these aspects.

Crowd animation: How should virtual entities move around and avoid collisions with both a static environment and dynamic objects How can a group move in a coordinated manner? [ALA*01a, GKM*01, AMC03, LD04, BBM05]? Chapter 4 presents some techniques to solve these problems.

Crowd behavior generation: How should a virtual crowd respond to changes in their surroundings? How should agents respond to behaviors of other agents? What is an appropriate way of modeling perception for many agents [Rey87, TT94, HB94, BCN97, BH97, Rey99, Mus00, UT02, NG03]? Chapter 5 describes some methods used for solving these questions.

Interaction with virtual crowds: How and which information should be exchanged from real people to control virtual humans? What is the most efficient metaphor to direct crowds of virtual extras [FRMS*99, UdHCT04]? Chapter 6 presents some discussion concerning these aspects.

Virtual crowd rendering: How to display many animated characters, quickly? How to display a wide variety of appearances [ABT00, LCT01, TLC02a, WS02, dHSMT05, CM05]? Chapter 7 explains some details concerning crowd rendering.

Integration of crowds in virtual environments: Which aspects of the environment need to be modeled? Which representation of environmental objects is best suited for fast behavior computation [FBT99, BLA02, KBT03, LMM03, PVM05]? Chapter 8 presents some discussion about these aspects.

Many of these aspects are to a greater or lesser extent intertwined. For example, efficiency of rendering constrains the possible variety of behaviors and appearances; higher-level behavior generation controls lower-level motion systems, but the behavior should also respond appropriately to collisions encountered while moving; the behavior model affects interaction possibilities; the environment representation affects possible behaviors; relating real and virtual humans allows handling of more complex behavior and environment representations and so on.

This book aims to discuss some of these aspects, organized in eight chapters, also including a state-of-the-art and presentation of some relevant applications developed by the authors.

2

State-of-the-Art

One of the largest areas where crowd behaviors have been modeled is the domain of safety science and architecture with the dominant application of crowd evacuation simulators. Such systems model movements of a large number of people in usually closed and well-defined spaces like inner areas of buildings [TM95a, BBM05], subways [Har00], ships [KMKWS00], or airplanes [OGLF98]. Their goal is to help designers to understand the **relation between the organization of space and human behavior** [OM93].

The most common use of evacuation simulators is the modeling of crowd behavior in case of forced evacuation from a confined environment due to some threat like fire or smoke. In such a situation, a number of people have to evacuate the given area, usually through a relatively small number of fixed exits. Simulations are trying to help answer questions like: Can the area be evacuated within a prescribed time? Where do the holdups in the flow of people occur? Where are the likely areas for a crowd surge to produce unacceptable crushing pressure [Rob99]? The most common modeling approach in this area is the use of cellular automata serving both as a representation of individuals and as a representation of the environment.

Simulex [TM95a,TM95b] is a computer model simulating the escape movement of persons through large, geometrically complex building spaces defined by 2D floor plans and connecting staircases. Each individual has attributes such as position, body size, angle of orientation, and walking speed. Various algorithms as distance mapping, way finding, overtaking, route deviation, and adjustment of individual speeds due to proximity of crowd members are used to compute egress simulation, where individual building occupants walk toward and through the exits.

G. Still developed a collection of programs named *Legion* for simulation and analysis of the crowd dynamics in evacuation from constrained and complex environments like stadiums [Sti00]. Dynamics of crowd motion is modeled by mobile cellular automata. Every person in the crowd is treated as an individual, calculating its position by scanning its local environment and choosing an appropriate action.

Helbing et al. [HM95, HFV00, WH03] proposed a model based on physics and sociopsychological forces in order to describe the human crowd behavior in panic situations. The model is set up by a particle system where each particle i of mass m_i has a predefined speed v_i^0, i.e., the desired velocity, in a certain direction $\mathbf{e}_i^0$ to which it tends to adapt its instantaneous velocity $\mathbf{v}_i$ within a certain time interval τ (for 1st term of Equation 2.1). Simultaneously, the particles try to keep a velocity-dependent distance from other entities j and walls w controlled by interaction forces $\mathbf{f}_{ij}$ and $\mathbf{f}_{iw}$ (second and third terms of Equation 2.1), respectively. The change of velocity with time t is given by the dynamical equation:

$$m_i \frac{dv_i}{dt} = F_i^{(H)} = m_i \frac{v_i^0 \mathbf{e}_i^0 - \mathbf{v}_i(t)}{\tau_i} + \sum_{j \neq i} \mathbf{f}_{ij} + \sum_{w} \mathbf{f}_{iw} \tag{2.1}$$

Braun et al. [BMB03, BBM05] extended the Helbing Model ($F_i^{(H)}$) in order to deal with different individuals and group behaviors, and also with complex environments. In this work, the agents' population can be composed heterogeneously by individuals with different attributes.

2.1 Crowd Management Training Systems

The modeling of crowds has also been essential in police and military simulator systems used for training in how to deal with mass gatherings of people.

CACTUS [Wil95] is a system developed to assist in planning and training for public order incidents such as large demonstrations and marches. The software designs are based on a world model in which crowd groups and police units are placed on a digitized map and have probabilistic rules for their interactive behavior. The simulation model represents small groups of people as discrete objects. The behavioral descriptions are in the form of a directed graph where the nodes describe behavioral states (to which correspond actions and exhibited emotions) and transitions represent plausible changes between these states. The transitions depend on environmental conditions and probability weightings. The simulation runs as a decision making exercise that can include pre-event logistic planning, incident management, and debriefing evaluation.

Small Unit Leader Non-Lethal Training System [VSMA98] is a simulator for training U.S. Marines Corps in decision making with respect to the use of nonlethal munitions in peacekeeping and crowd control operations. Trainees learn rules of engagement, the procedures for dealing with crowds and mobs, and the ability to make decisions about the appropriate level of force needed to control, contain, or disperse crowds and mobs. Crowds move within a simulated urban environment along instructor-predefined pathways and respond both to actions of a trainee and to actions of other simulated crowds. Each crowd is characterized by a crowd profile—series of attributes like fanaticism, arousal state, prior experience with nonlethal munitions, or attitude toward Marines. During an exercise, the crowd behavior computer model operates in real time and responds to trainee actions (and inactions) with appropriate simulated behaviors such as loitering, celebrating, demonstrating, rioting, and dispersing according to a set of Boolean relationships defined by experts.

2.2 Sociological Models of Crowds

Despite being a field primarily interested in studying collective behavior, only a relatively small number of works on crowd simulations have been done in sociology.

McPhail et al. [MPT92] studied individual and collective actions in temporary gatherings. Their model of the crowd is based on perception control theory [Pow73] where each separate individual is trying to control his or her experience in order to maintain a particular relationship to others: in this case it is a spatial relationship with others in a group. The simulation program *GATHERING* graphically shows movement, milling, and structural emergence in crowds. The same simulation system was later used by Schweingruber [Sch95] to study the effects of reference signals common to coordination of collective behavior and by Tucker et al. [TSM99] to study formation of arcs and rings in temporary gatherings.

Jager et al. [JPvdS01] modeled clustering and fighting in two-party crowds. A crowd is modeled by a multi agent simulation using cellular automata with rules defining approach–avoidance conflict. The simulation consists of two groups of agents of three different kinds: hardcore, hangers-on, and bystanders, the difference between them consisting in the frequency with which they scan their surroundings. The goal of the simulation was to study effects of group size, size symmetry, and group composition on clustering, and “fights”.

2.3 Group Behavior in Robotics and Artificial Life

Researchers working in the field of artificial life are interested in exploring how group behavior emerges from local behavioral rules [Gil95]. Software models and groups of robots were designed and experimented with in order to understand how complex behaviors can arise in systems guided by simple rules. The main source of inspiration is nature, where, for example, social insects efficiently solve problems such as finding food, building nests, or division of labor among nestmates by simple interacting individuals without an overseeing global controller. One of the important mechanisms contributing to a distributed control of the behavior is **stigmergy**, indirect interactions among individuals through modifications of the environment [BDT99].

Dorigo introduced **ant systems** inspired by behaviors of real ant colonies [Dor92]. Ant algorithms have been successfully used to solve a variety of discrete optimization problems including the traveling salesman problem, sequential ordering, graph coloring, or network routing [BDT00]. Besides insects, groups of more complex organisms such as flocks of birds, herds of animals, and schools of fish have been studied in order to understand principles of their organization. Recently, Couzin et al. presented a model of how animals that forage or travel in groups can make decisions even with a small number of informed individuals [CKFL05].

Principles from biological systems were also used to design behavior controllers for autonomous groups of robots. Mataric studied behavior-based control for a group of robots, experimenting with a herd of 20 robots whose behavioral repertoire included safe wandering, following, aggregation, dispersion, and homing [Mat97]. Molnar and Starke have been working on assignment of robotic units to targets in a manufacturing environment using a pattern formation inspired by pedestrian behavior [MS01]. Martinoli applied swarm intelligence principles to autonomous collective robotics, performing experiments with robots that were gathering scattered objects and cooperating to pull sticks out of the ground [Mar99]. Holland and Melhuish experimented with a group of robots doing sorting of objects based on ant behaviors where ants sort larvae and cocoons [HM99]. In an interesting work using a robot to control animal behavior, Vaughan et al. developed a mobile robot that gathers a flock of real ducks and maneuvers them safely to a specified goal position [VSH*00].

2.4 Crowds in Virtual Worlds

In order to have a persuasive application using crowds in virtual environments, various aspects of the simulation have to be addressed, including behavioral animation, environment modeling, and crowd rendering. If there is no satisfactory rendering, even the best behavior model will not be very convincing. If there is no good model of a behavior, even a simulation using the best rendering method will look dumb after only a few seconds. If there is no appropriate model of the environment, characters will not behave believably, as they will perform actions in the wrong places, or not perform at all.

The goal of behavioral animation is to **ease the work of designers** by letting virtual characters perform autonomously or semiautonomously complicated motions which otherwise would require large amounts of human animators' work; or, in case of interactive applications, the behavioral models allow characters to **respond to user initiated actions**.

In order for a behavior to make sense, besides characters, their surrounding environment has to be modeled, not just graphically but also semantically. Indeed, a repertoire of possible behaviors is very dependent on what is and what is not included in a model of the environment. It happens very often that the environment is visually rich, but the interaction of characters with it is minimal.

Finally, for interactive applications, it is necessary to display a varied ensemble of virtual characters in an efficient manner. Rendered characters should visually "fit" into the environment—they should be affected by light and other effects in the same manner as their surroundings.

Next, we will present representative works for each of these topics grouped according to their main focus.

2.5 Behavioral Animation of Groups and Crowds

Human beings are arguably the most complex known creatures, therefore they are also the most complex creatures to simulate. A behavioral animation of human (and humanoid) crowds is based on foundations of group simulations of much more simple entities, notably flocks of birds [Rey87, GA90] and schools of fish [TT94]. The first procedural animation of flocks of virtual birds was shown in the movie by Amkraut, Girard, and Karl called Eurhythmy, for which the first concept [AGK85] was presented at The Electronic Theater at SIGGRAPH in 1985 (final version was presented at Ars Electronica in 1989). The flock motion was achieved by a global vector force field guiding a flow of flocks [GA90].

In his pioneering work, Reynolds [Rey87] described a distributed behavioral model for simulating aggregate motion of a flock of birds. The technical paper was accompanied by an animated short movie called "Stanley and Stella in: Breaking the Ice" shown at the Electronic Theater at SIGGRAPH '87. The revolutionary idea was that a **complex behavior** of a group of actors can be obtained by **simple local rules** for members of the group instead of some enforced global condition. The flock is simulated as a complex particle system, with the simulated birds (called **boids**) being the particles. Each boid is implemented as an independent agent that navigates according to its local perception of the environment, the laws of simulated physics, and the set of behaviors. The boids try to avoid collisions with one another and with other objects in their environment, match velocities with nearby flock mates, and move toward a center of the flock. The aggregate motion of the simulated flock is the result of the interaction of these relatively simple behaviors of the individual simulated birds. Reynolds later extended his work by including various steering behaviors as goal seeking, obstacle avoidance, path following, or fleeing [Rey99], and introduced a simple finite-state machines behavior controller and spatial queries optimizations for real-time interaction with groups of characters [Rey00].

Tu and Terzopoulos proposed a framework for animation of artificial fishes [TT94]. Besides complex individual behaviors based on perception of the environment, virtual fishes have been exhibiting unscripted collective motions as schooling and predator evading behaviors analogous to flocking of boids.

An approach similar to boids was used by Bouvier et al. [BG96, BCN97] to simulate human crowds. They used a combination of particle systems and transition networks to model crowds for the visualization of urban spaces. At the lower level, attractive and repulsive forces, analogous to physical electric ones, enable people to move around the environment. Goals generate attractive forces, obstacles generate repulsive force fields. Higher level behavior is modeled by transition networks with transitions depending on time, visiting of certain points, changes of local population densities, and global events.

Brogan and Hodgins [HB94, BH97] simulated group behaviors for systems with **significant dynamics**. Compared to boids, a more realistic motion is achieved by taking into account physical properties of motion, such as momentum or balance. Their algorithm for controlling the movements of creatures proceeds in two steps: first, a perception model determines the creatures and obstacles visible to each individual, and then a placement algorithm determines the desired position for each individual given the locations and velocities of perceived creatures and obstacles. Simulated systems included groups of one-legged robots, bicycle riders, and point-mass systems.

Musse and Thalmann [Mus00, MT01] presented a **hierarchical model** for real-time simulation of virtual human crowds. Their model is based on groups, instead of individuals: groups' are more intelligent structures, where individuals follow the groups specification. Groups can be controlled with different levels of autonomy: guided crowds follow orders (as go to a certain place or play a particular animation) given by the user in run-time; programmed crowds follow a scripted behavior; and autonomous crowds use events and reactions to create more complex behaviors. The environment comprises a set of interest points, which signify goals and way points; and a set of action points, which are goals that have some actions associated. Agents move between way points following Bezier curves.

Recently, another work was exploring group modeling based on hierarchies. Niederberger and Gross [NG03] proposed an architecture of hierarchical and heterogeneous agents for real-time applications. Behaviors are defined through specialization of existing behavior types and weighted multiple inheritance for creation of new types. Groups are defined through recursive and modulo based patterns. The behavior engine allows for the specification of a maximal amount of time per run in order to guarantee a minimal and constant frame rate.

Ulicny and Thalmann [UT01, UT02] presented a crowd behavior simulation with a modular architecture for multiagent system allowing autonomous and scripted behavior of agents supporting variety. In their system, the behavior is computed in layers, where decisions are made by behavioral rules and execution is handled by hierarchical finite-state machines.

Most recently, a real-time crowd model based on continuum dynamics has been proposed by [TCP06b]. In their model, a dynamic potential field integrates global navigation with moving obstacles, efficiently solving for the motion of large crowds without the need for explicit collision avoidance.

Perceived complexity of the crowd simulation can be increased by using **levels of detail** (LOD). O'Sullivan et al. [OCV*02] described a simulation of crowds and groups with level of details for geometry, motion, and behavior. At the geometrical level, subdivision techniques are used to achieve smooth rendering LOD changes. At the motion level, the movements are simulated using adaptive levels of detail. Animation subsystems with different complexities, as a keyframe player or a real-time reaching module, are activated and deactivated based on heuristics. For the behavior, LOD is employed to reduce the computational costs of updating the behavior of characters that are less important. More complex characters behave according to their motivations and roles, less complex ones just play random keyframes.

2.6 Environment Modeling for Crowds

Environment modeling is closely related to behavioral animation. The purpose of the models of the environment is to facilitate simulation of entities dwelling in their surrounding environments. Believability of virtual creatures can be greatly enhanced if they behave in accordance with their surroundings. On the contrary, the suspense of disbelief can be immediately destroyed if they perform something not expected or not permitted in the real world, such as passing through the wall or walking on water. The greatest efforts have therefore been directed to representations and algorithms preventing "forbidden" behaviors from occuring; until quite recently the two major artificial intelligence issues concerning game development industry were collision avoidance and path-planning [Woo99, DeL00].

The majority of the population in the developed world lives in cities; it is there that most human activities take place nowadays. Accordingly, most of the research has been done for **modeling of virtual cities**. Farenc et al. [FBT99] introduced an **informed environment** dedicated to the simulation of virtual humans in the urban context. The informed environment is a database integrating semantic and geometrical information about a virtual city. It is based on a hierarchical decomposition of an urban scene into environment entities, like quarters, blocks, junctions, streets, and so on. Entities can contain a description of the behaviors that are appropriate for agents located on them; for example, a sidewalk tells that it should be walked on, or a bench tells that it should be sat on. Furthermore, the environment database can be used for a path-finding that is customized according to the type of client requesting the path, so that, for example, a pedestrian will get paths using sidewalks, but a car will get paths going through roads.

Another model of a virtual city for a behavioral animation was presented by Thomas and Donikian [TD00]. Their model is designed with the main emphasis on traffic simulation of vehicles and pedestrians. The environment database is split into two parts— a hierarchical structure containing a tree of polygonal regions, similar to the informed environment database; and a topological structure with a graph of a road network. Regions contain information on directions of circulation, including possible route changes at intersections. The agents then use the database to navigate through the city.

In a recent work, Sung et al. [SGC04] presented a new approach to control the behavior of a crowd by storing behavioral information into the environment using structures called **situations**. Compared with previous approaches, environmental structures (situations) can overlap; behaviors corresponding to such overlapping situations are then composed using probability distributions. Behavior functions define probabilities of state transitions (triggering motion clips) depending on the state of the environment features or on the past state of the agent.

On the side focused on more generic **path-planning** issues, several works have been done. Kallmann et al. [KBT03] proposed a fast path-planning algorithm based on a fully dynamic constrained Delaunay triangulation. Bayazit et al. [BLA02] used global roadmaps to improve group behaviors in geometrically complex environments. Groups of creatures exhibited behaviors such as homing, goal searching, covering, or shepherding, by using rules embedded both in individual flock members and in roadmaps. Tang et al. [TWP03] used a modified A* algorithm working on a grid overlaid over a height-map generated terrain. Recently, Lamarche and Donikian [LD04] presented a topological structure of the geometric environment for a fast hierarchical path-planning and a reactive navigation algorithm for virtual crowds. Most recently, work presented by Pettré et al. [PLT05] shows how to automatically and robustly compute a multilevel navigation graph using three-dimensional cylinders. This work also shows how to reuse the resulting path-planning computation for a few hundred agents that can react to congestion along the path.

2.7 Crowd Rendering

Real-time rendering of a large number of 3D characters is a considerable challenge; it is able to exhaust system resources quickly even for state-of-the-art systems with extensive memory resources, fast processors, and powerful graphic cards. “Brute-force” approaches that are feasible for a few characters do not scale up for hundreds, thousands, or more of them. Several works have been trying to circumvent such limitations by clever use of graphics accelerator capabilities, and by employing methods profiting from the fact that our perception of the scene as a whole is limited.

We can perceive in full detail only a relatively small part of a large collection of characters. A simple calculation shows that to treat every crowd member as equal is rather wasteful. Modern screens can display around 2 million pixels at the same time, where a fairly complex character can contain approximately 10,000 triangles. Even if assuming that every triangle were be projected to a single pixel, and that there would be no overlap of characters, the screen fully covered by a crowd would contain only around 200 simultaneously visible characters. Of course, in reality the number would be much smaller; a more reasonable estimate is around a few dozen fully visible characters, with the rest of the crowd either being hidden behind these prominent characters or taking significantly less screen space. Therefore, it makes sense to take full care only of the foremost agents, and to replace the others with some less complex approximations. Level of details techniques then switch visualizations according to position and orientation of the observer. In the recent work of Hamill et al. [HMDO05] they pursue psychophysics, a discipline to decide perceptual limitations to the human vision system for example. Doing tests on how motion affects the perception of a human represented by an impostor or by a geometric structure, they were able to define distances of least noticeable switching between models.

Billboarded impostors are one of the methods used to speed up crowd rendering. Impostors are partially transparent textured polygons that contain a snapshot of a full 3D character and are always facing the camera. Aubel et al. [ABT00] introduced dynamically generated impostors to render animated virtual humans. In their approach, an impostor creating process is running in parallel to full 3D simulations, taking snapshots of rendered 3D characters. These cached snapshots are then used over several frames instead of the full geometry until a sufficient movement of either camera or a character will trigger another snapshot, refreshing the impostor texture.

In another major work using impostors, Tecchia et al. [TLC02a] proposed a method for real-time rendering of an animated crowd in a virtual city. Compared with the previous method, impostors are not computed dynamically, but are created in a preprocessing step. Snapshots are sampled from viewpoints distributed in the sphere around the character. This process is repeated for every frame of the animation. In run-time, images taken from viewpoints closest to the actual camera position are then used for texturing of the billboard. Additionally, the silhouettes of the impostors are used as shadows projected to a ground surface. Multitexturing is used to add variety by modulating colors of the impostors. In a later work they added lighting using normal maps [TLC02b]. Their method using precomputed impostors is faster than dynamical impostors, but it is very demanding on texture memory, which leads to lower image quality as size of textures per character and per animation frame have to be kept small.

A different possibility for a fast crowd display is to use **point-based rendering techniques**. Wand and Strasser [WS02] presented a multiresolution rendering approach which unifies image based and polygonal rendering. They create a view-dependent octree representation of every keyframe of animation, where nodes store either a polygon or a point. These representations are also able to interpolate linearly from one tree to another so that in-between frames can be calculated. When the viewer is at a long distance, the human is rendered using point rendering; when zoomed in, using polygonal techniques; and when in between, a mix of the two.

An approach that has been getting new life is that of **geometry baking**. By taking snapshots of vertex positions and normals, complete mesh descriptions are stored for each frame of animation as in the work of Ulicny et al. [UdHCT04]. Since current desktop PCs have large memories, many such frames can be stored and replayed. A hybrid approach of both baked geometry and billboarding was presented at I3d, where only a few actors are fully geometrical while the vast number of actors are made up of billboards [DHOO05a]. A similar approach can be found in [CLM05]. A more recent approach to crowd rendering using geometry is through **dynamic meshes** as presented in the work of de Heras et al. [dHSMT05], where dynamic meshes use systems of caches to reuse skeletal updates which are typically costly. A hybrid of dynamic and baked meshes is found in [YMdHC*05] where the graphics programming unit (GPU) is used to its fullest.

What is common to all approaches is instancing of template humans, by changing the texture or color, size, orientation, animation, animation style, and position. This is carefully taken care of to smoothly transition from one representation to another so as not to create pops in representation styles. In the billboarding scenario this is done by applying different colors on entire zones such as torso, head, legs, and arms. This way the texture memory is used more efficiently as the templates are more flexible. For the geometrical approaches these kinds of differences are usually represented using entirely different textures as the humans are too close just to change basic color for an entire zone [UdHCT04].

2.8 Crowds in Non-Real-Time Productions

One of the domains with the fastest growth of crowd simulations in recent years is special effects. While only 10 years ago, there were no digital crowds at all, nowadays almost every blockbuster has some, with music videos, television series, and advertisements starting to follow. In comparison with crowds of real extras, virtual crowds allow one to significantly reduce costs of production of massively populated scenes and allow for bigger creative freedom because of their flexibility. Different techniques, as replications of real crowd video footage, particle systems, or behavioral animation, have been employed to add crowds of virtual extras to shots in a broad range of movies, from historical dramas,[1–3] through fantasy and science fiction stories,[4–6] to animated cartoons.[7–9]

---

[1] http://www.titanicmovie.com
[2] http://www.dreamworks.com
[3] http://troymovie.warnerbros.com
[4] http://www.starwars.com/
[5] http://www.lordoftherings.net
[6] http://whatisthematrix.warnerbros.com
[7] http://www.pixar.com/featurefilms/abl
[8] http://disney.go.com/disneyvideos/animatedfilms/lionking
[9] http://www.shrek2.com

The main factors determining the choice of techniques are the required visual quality and the production costs allowed for the project [Leh02]. It is common to use different techniques even in a single shot in order to achieve the best visuals; for example, characters in the front plane are usually real actors, with 3D characters taking secondary roles in the background.

Although a considerable amount of work was done on crowds in movies, only relatively little information is available, especially concerning more technical details. Most knowledge comes from disparate sources, for example, from "making-of" documentary features, interviews with special effects crew or industry journalist accounts. For big budget productions, the most common approach is **in-house development** of **custom tools** or suites of tools which are used for a particular movie. As the quality of the animation is paramount, large libraries of motion clips are usually used, produced mainly by motion capture of live performers. All production is centered around shots, most of the times only a few seconds long. In contrast to real-time simulations, there is little need for continuity of the simulation over longer periods of the time. It is common that different teams of people work on parts of the shots which are then composited in postprocessing.

The most advanced crowd animation system for non-real-time productions is *Massive*; used to create battle scenes for *The Lord of the Rings* movie trilogy.[10] In *Massive*, every agent makes decisions about its actions depending on its sensory inputs using a brain composed of thousands of logic nodes [Koe02]. According to the brain's decision, the motion is selected from an extensive library of motion captured clips with precomputed transitions. For example, in the second part of the trilogy over 12 million motion captured frames (equivalent to 55 hours of animation) were used. *Massive* also uses rigid body dynamics, a physics-based approach to facilitating realistic stunt motion such as falling, or animation of accessories. For example, a combination of physics-based simulation and custom motion capture clips was used to create the scene of "The Flooding of Isengard" where orcs are fleeing from a wall of water and falling down the precipice [Sco03].

In comparison with real-time application, the quality of motion and visuals in non real-time productions is far superior, but it comes at a great cost. For example, for *The Lord of the Rings: The Two Towers*, rendering of all digital characters took 10 months of computations on a strong render farm with thousands of computers [Doy03].

2.9 Crowds in Games

In current computer games virtual crowds are still relatively rare. The main reason is that crowds are inherently costly, both in terms of real-time resource requirements and for costs of a production. Nevertheless, the situation is

---

[10] http://www.massivesoftware.com