Volume 21 (2002), number 4 pp. 767–775 COMPUTER GRAPHICS *forum*

Towards Interactive Real-Time Crowd Behavior Simulation

Branislav Ulicny and Daniel Thalmann

Virtual Reality Lab, EPFL, CH-1015 Lausanne, Switzerland
Branislav.Ulicny@epfl.ch, Daniel.Thalmann@epfl.ch

Abstract

While virtual crowds are becoming common in non-real-time applications, the real-time domain is still relatively unexplored. In this paper we discuss the challenges involved in creating such simulations, especially the need to efficiently manage variety. We introduce the concept of levels of variety. Then we present our work on crowd behaviour simulation aimed at interactive real-time applications such as computer games or virtual environments. We define a modular behavioural architecture of a multi-agent system allowing autonomous and scripted behaviour of agents supporting variety. Finally we show applications of our system in a virtual reality training system and a virtual heritage reconstruction.

Keywords: autonomous agents, crowd simulations, levels of variety, multi-agent systems, virtual environments, virtual heritage, virtual reality training systems

ACM CSS: I.3.7 Three-Dimensional Graphics and Realism—*Animation,* I.2.11 Distributed Artificial Intelligence—*Multi-agent systems*

1. Introduction

In recent years, virtual crowds have become a more and more common element of our cinematic experience. Whether it was a photo-realistic crowd of digital passengers in Titanic, a legion of animated ants in AntZ, or a rendered army of droids in Star Wars, computer graphics (CG) generated crowds of characters added significantly to the impact of the movies employing them, allowing the visualization of scenes not possible only some years ago. Crowds of CG human and non-human characters help to overcome the prohibitive costs and complexities of working with large numbers of extras, substitute as virtual stunt-men in dangerous shots and integrate well with virtual scenes.

The situation is different, however, in the realm of real-time applications: CG crowds are still rare in computer games, virtual reality educational or training systems. Out of several thousands of titles produced every year, only a few employ larger numbers of characters. Several sport games include spectator crowds with very simple behaviours. In Rockstar Games's State of Emergency, a virtual mob provides background to this action game and Elixir Studios's Republic The Revolution features crowds of virtual citizens.

Because of different requirements and constraints in almost every aspect (such as character generation and rendering, or motion and behaviour control), different approaches are needed compared to the relatively simpler domain of non-real-time crowds used in motion pictures. In addition to the common tasks of managing the variety and easing control of multiple characters, real-time applications need to handle interactivity and have to deal with the limited computational resources available.

Several works in different fields have been exploring issues connected to the domain of crowd simulations. In his pioneering work, Reynolds [1] described distributed behavioural model for simulating the aggregate motion of a flock of birds. Bouvier and Guilloteau [2] used a combination of particle systems and transition networks to model human crowds in the visualization of urban spaces. Brogan and Hodgins [3] simulated group behaviours for systems with significant dynamics. Aubel and Thalmann [4] introduced dynamically generated impostors to render virtual humans. Tecchia *et al.* [5] proposed image-based methods for real-time rendering of animated crowds in virtual cities. O'Sullivan *et al.* [6] described crowd and

group simulation with levels of detail for geometry, motion and behaviour. McPhail *et al.* [7] studied individual and collective actions in temporary gatherings. Still [8] used mobile cellular automata for simulation and analysis of crowd evacuations. However, only a few works [9,10] tried to explore more general crowd models, integrating several sub-components such as collision avoidance, path-planning, higher-level behaviours, interaction or rendering.

In this paper we present our current work building on the real-time crowd simulation system described by Ulicny and Thalmann [10]. We model a crowd as a multi-agent system with emphasis on individuals (in contrast to groups [9]). We define a layered modular behavioural architecture allowing autonomous and scripted behaviour of the agents supporting variety, based on the combination of rules and finite state machines for higher level behaviour computation, and path-finder and collision-avoidance for lower-level motion control.

The structure of the paper is as follows. First we explore assumptions and challenges for real-time interactive crowd simulations, especially compared to non-real-time crowd and real-time single agent systems, followed by the analysis of the levels of variety. Then we present the overview of the model of the world, with a more detailed description of the behaviour model. We briefly discuss the implementation of the system and finally, before concluding, we present two case studies: a virtual reality training system and a virtual heritage reconstruction, each employing our crowd simulation system.

2. Assumptions and Challenges

With the increasing speed of computers, more complex simulations are becoming possible: nowadays it is feasible to display scenes containing hundreds of thousands of polygons at interactive rates. The main challenge is now becoming how to bring life to these scenes, how to animate virtual objects in a persuasive way.

Simulating human beings is complex task at every level. Ideally we would like to have a complete model of the world, indistinguishable from reality, running on the computer in real-time. With the current state of the art it is obviously not possible (and it is questionable if it ever will be [11]), so we need to focus on such aspects of reality that allow us to sufficiently model a subset of the world for the application to be useful. We need to select the target space and time scale and resolution of our simulations: for example if the modelled scenario lasts for five minutes, it will probably use a much more detailed model than a scenario lasting five hours. However, it cannot then be expected to give reasonable results when forced to run for a longer time.

In our work we focus on multi-agent virtual human simulations able to run in real-time with 3D visualization allowing user interaction such as computer games, shared virtual worlds, or VR training systems. This narrows the area of our interest to situations observable in real-time as they happen, excluding for example studies of crowd behaviour happening over larger time periods not localized in the exact space common in sociology [12].

Our target simulations bring different challenges compared to the systems either involving small number of interacting characters (e.g. the majority of contemporary computer games), or non-real-time applications (e.g. crowds in movies, visualization of crowd evacuation after off-line model computation).

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, move, react and sound different from each other. Even if assuming that the 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 the control of multiple characters are needed, but 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 CPU time or memory space. Fast and scalable methods to compute behaviour, able to take into account inputs not known in advance, are needed.

3. Levels of Variety

One of the important issues that arises with increasing numbers of simulated entities is the question of their variety (Merriam–Webster's Dictionary defines variety as "the quality or state of having different forms or types"). Even subtle variations in the motions or the look of the individual virtual humans can greatly enhance the realism of the virtual crowd as a whole. For analysis of the crowd simulation systems and their components it is useful to be able to define the degree of variety more precisely.

We introduce the notion of levels of variety: we say that a system has level of variety zero (LV0) if, for a given task, it is using only a single solution, level of variety one (LV1) if it is able to make a choice from a finite number of solutions (here it can be useful to distinguish another sub-level LV1+ if the solution is composed of combinations of sub-solutions), and level of variety two (LV2) if it is able to use solutions chosen from an infinite number of possible solutions.

LV0 systems can be relatively easily upgraded to LV1 by adding meta-layers, allowing the selection of one solution out of a defined set; LV2 systems need generative models.

Figure 1: *Crowd in the virtual world.*

In practice LV1 and LV2 systems can be perceptually indistinguishable as it is always possible to inject more pre-defined solutions into an LV1 system, but this is feasible only for a small number of required solutions.

Let us consider the example of crowd visualization: a system where the virtual crowd is composed of only one type of human would be LV0, a system where the crowd is composed of multiple humans selected from a pre-defined set would be LV1 (or alternatively LV1+ if these humans would be composed of sets of exchangeable parts such as heads, bodies, textures) and finally a system would be LV2 if it would be able to display a potentially infinite number of unique humans generated for example by a parameterizable anthropometric model generating humans with different morphologies [13].

The challenge of applying most of the classical artificial intelligence or computer graphics approaches to the domain of crowd simulations is that they were not designed with the aim of achieving variety. For example, the usual goal of path-finding algorithms is to find the best solution (that is LV0) for how to get from one place in some environment to another one. Using such an ideal path without further modification for multiple entities, however, would result in unwanted artefacts of the entities moving in queues.

We would like to note, however, that even while the real world exhibits large variety, systems with a lower level of variety are justifiable, sometimes even preferable over systems with higher variety. For example, when simulating the emergency egress of a crowd from a burning building, perfect visualization could be distracting; simpler uniform visualization of the individuals could help to emphasize problems with the flow of fleeing people.

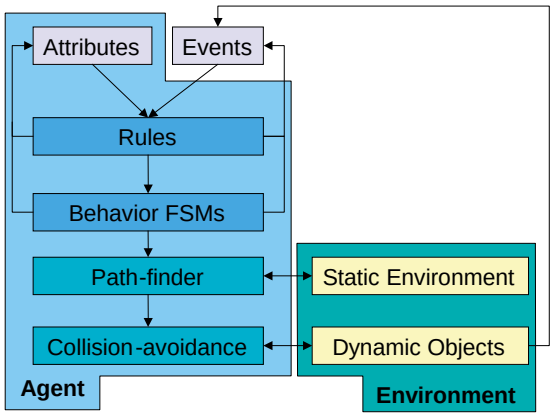

Figure 2: *The model of the world.*

4. The Model of the World

Our simulations consist of autonomous virtual human agents existing in a dynamic virtual 3D environment (see Figure 1). The model (see Figure 2) is composed of the agents, dynamic objects and a static environment. We distinguish between the static part of the environment, such as the layout of the streets and buildings, and the dynamic part consisting of objects that can change their position or state during the scenario, such as fire or gas clouds.

Parts of the model can, but do not necessarily have to, have visualization counterparts. For example, invisible objects such as exits can convey the semantics of the environment. We discern between the behaviour-displaying part of the virtual human agent (hereto referred to as a virtual human) and the behaviour controlling part (referred

Figure 3: *Crowd using simple humanoid visualization.*

Figure 4: *Crowd using complex humanoid visualization.*

to as an agent). The same simulation can be run using different visualizations: Figures 3 and 4 show the same scenario (crowd in the park) using LV0 simple humanoids and LV1 complex humanoids with deformable bodies as visualizations.

Agents with the most complex visualization have 3D graphics body representations and are able to perform certain low-level actions, such as the playing of pre-recorded body animation sequences (such as gestures or changes of postures), walking to a specified location with different gaits [14], displaying facial animation sequences [15], looking at specified places or playing 3D localized sounds. Higher-level behaviours are then composed of particular combinations of these low-level actions.

Agents contain a set of internal attributes that can correspond to various psychological or physiological states needed to model particular scenarios (such as memory, fear, mobility, or level of injuries), a set of higher-level complex behaviours (such as wander, flee or follow path) and a set of rules determining selection of these behaviours. Agents can interact with both static and dynamic parts of the environment.

Interaction between the agents and the static environment is done by a shared path-finder module which allows agents to move around the scene in the correct way. Waypoints on the path are not defined as exact locations, but as random places from some epsilon surrounding of a path node, thus ensuring LV2 variety of the individual trajectories of more agents following the same path.

Interaction between the agents and the dynamic objects is done via exchanges of events, simulating either physical interactions (e.g. effects of a fire on the agent, or the agent on the fire) or perceptual interactions (e.g. agent perceiving danger at a specified distance from the threat). Further events serve also as a means of inter-agent communication (e.g. an agent which is injured sends a message requesting for help) and user-agent interaction where the user can send events (such as an order to stop) to the virtual humans via a user interface.

5. Behaviour Model

In order to behave in a believable way, agents must act in accordance with their surrounding environment, be able to react to its changes, to the other agents and also to the actions of the real humans interacting with the virtual world. We need a model connecting the perception of the agents with their actions.

Our aim is to have a behaviour model that is simple enough to allow for real-time animation of many agents, yet still sufficiently complex to provide interesting behaviours. Considering the requirements mentioned in Section 2, we propose the following model (see Figure 2) based on the combination of rules and finite state machines (FSM) for determining agents' behaviours using a layered approach.

At the highest level, rules select high-level behaviours according to the state of the agent constituted by attributes and the state of the virtual environment conveyed by events. The rules consist of three parts:

(1) **Selection part** — for *who* (e.g. a particular agent, or agents in a particular group).
(2) **Condition part** — *when* the rule is applicable (e.g. at a defined time, after receiving an event, when some attribute reaches a specified value or any boolean combination of such conditions).
(3) **Consequent part** — *what* is the consequence of rule firing (e.g. a change of an agent's behaviour or attribute, or sending the event).

The reason for splitting the usually single antecedent part into two sections is the optimization of the rule-base use, where the condition part is evaluated only for relevant agents. According to our experience it is more practical to store all the rules for all the agents in a single rule-base (as opposed to each agent keeping their own set of rules). In such a way it is easier to maintain consistency of the rules.

An example of a rule from the actual simulation (see Section 7) is:

```
FOR  ALL
WHEN EVENT = in_danger_area
     AND ATTRIBUTE fear > 50%
THEN BEHAVIOR FLEE
```

Variety of the reactions to the same situation is achieved by different agents having different values of the attributes (at the beginning through different initializations, later because of their different histories), which consequently leads to different rules being triggered.

At the middle level, high-level behaviours are implemented using hierarchical finite state machines. Each behaviour is realized by one FSM which drives the selection of the low-level actions for a virtual human (like 'move to location, play short animation sequence'), manages connections with the environment (like path queries or event sending) and also can call other FSMs to delegate sub-tasks such as path following.

There are two types of high-level behaviours. First we can specify a scripted behaviour by using explicit sequences of low-level actions, which is more precise, but less autonomous and with less environment coupling. Alternatively, we can let agents perform autonomously complex behaviours with feedback from the environment. Examples of such autonomous behaviours are wandering, fleeing, neutralizing the threat, or requesting and providing help. Both types can be mixed as needed.

At the lowest level, motion control is also organized into layers (see Figure 2). As a result of higher-level behaviour, an agent's behaviour FSM decides (or is told directly by the rule) that the agent wants to move to particular location. This request is forwarded to the path-finding layer, which constructs a sequence of waypoints that need to be passed to get to the location. Finally, it is the responsibility of the collision-avoidance layer to move the agent between waypoints, correcting its trajectory in order to avoid collisions.

6. System Implementation

In addition to the requirements posed by the nature of a targeted simulation, the software development of a complex system is by itself becoming a complex task that generates additional requirements for a highly modular and flexible architecture with the ability to exchange sub-components and test them separately.

Our system is designed with a clear separation of the model part (where the behaviour is computed) from the visualization part (where the behaviour is displayed). Every component of the model of the world (agents, objects, environment) is independent of its visualization: the model can be run without any graphical output. Such an organization allows the use of different representations of the virtual humans, objects and environments according to the needs of particular applications or in the different stages of the software life-cycle. For example, simplified visualization (see Figure 3) proved to be very helpful in shortening the development cycles. In extreme cases there doesn't have to be any visualization at all: for example, the crowd module can be used to control external entities in a distributed simulation (see Section 7).

Another advantage of such a design is that it allows the use of different update rates for different components: higher-level behaviour computation can be run with much lower frequency (and consequently a much smaller CPU time slice) than lower-level motion control, which again needs a lower frequency update than the refresh rate of the screen image.

A desirable side effect of this organization is that our crowd system is platform independent — it started as a monolithic application on SGI, continued as a module of the distributed system based on a standard PC, and currently it forms a part of a multi-platform virtual reality development framework. In all the applications, not just the conceptual architecture but most of the actual implementation code of the model is the same.

This architecture also addresses the animation variety issue (see Section 2), by separating the action selecting part (behaviour FSM) from the action executing part (virtual human controller). A behaviour FSM orders the controller to do a certain type of animation (such as waving a hand) and the controller then randomly chooses a particular one from the set of such animations with random variation of the speed of the playback, so that even if more agents are executing the same behaviour they don't necessarily act in exactly the same way.

7. Case Study: Crowds in Virtual Reality Training System

In this section we will present an application of our crowd simulation in a virtual reality training system called CROSSES (CROwd Simulation System for Emergency Situations). CROSSES is a training system for emergency situations. Its aim is to train people to efficiently react to

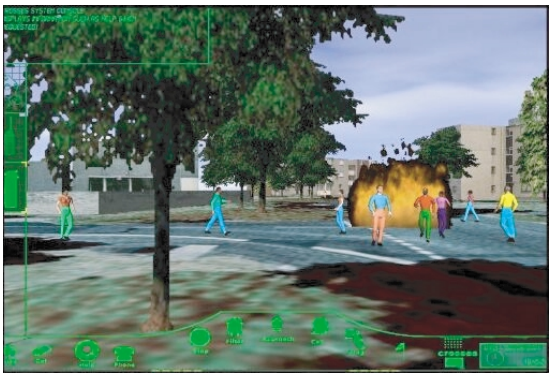

Figure 5: *A crowd reacting to an emergency event.*

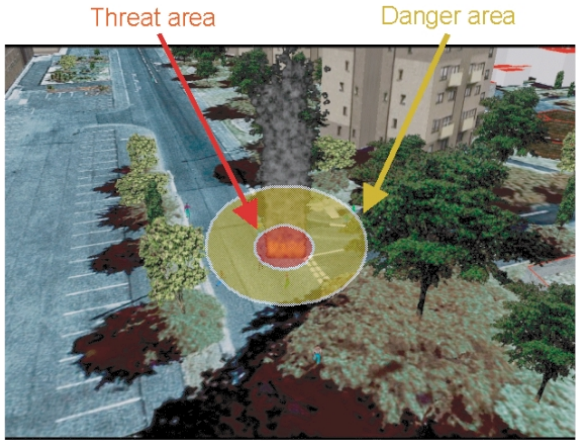

Figure 6: *Threat, a semantic environmental object.*

emergency situations such as the occurrence of a fire or leakage of a poisonous gas in a town with proximity to a chemical factory.

Training sessions take place in a virtual world populated by virtual humans that have complex behaviours dependent on the events in the simulation (see Figure 5). The virtual 3D urban environment, including buildings and trees, is reconstructed from aerial images [16] of the actual town. Areas accessible for walking have to be specified, allowing a path-finder to construct correct paths [17]. A realistic 3D virtual population of people with different professions (such as workers, firemen, policemen or paramedics) is constructed by automatic low-cost modelling [13].

The system is designed as a distributed application with components able to run on different computers and communicating by exchanging messages over the network.

The CROSSES project aims to reproduce scenarios depicting urban emergency situations involving crowds of virtual human agents with behaviours based on those of real people in such situations. The goal of the crowd module is to provide real-time approximation of the given behaviours before, during and after an emergency situation happens. It should also be possible for real human participants of the simulation (such as trainees or trainers) to affect the outcome of the scenario by their actions. The objective is to confront human participants with a real-time 3D reconstruction of the given scenario which is plausible enough to be useful in the training process, thus providing a trainee with enough information to be able to assess the situation and to make decisions influencing the scenario in a desired way.

In modelling the behaviour, we focus on reactivity to the scenario events, such as the occurrence of a fire or a gas leak, and interactivity towards the users, for example reacting to the “Stop” command given by the user interface.

We can illustrate autonomous behaviours with an example of an agent fleeing from a threat. When the threat (an environmental semantic object) becomes active by scenario or through the user interface, it sends events to the affected agents. Perception of the danger is simulated by the threat emitting events for the agents inside its danger area (see Figure 6). For a particular agent, the rule eventually becomes activated and “Flee” behaviour is triggered. The agent selects the way out of danger by using the environmental object called “Exit” and starts running toward the exit area. After passing through the exit, the agent returns to normal behaviour.

8. Case Study: Crowds in a Virtual Heritage Reconstruction

Another application of our crowd system is a reconstruction of a virtual heritage site. The aim of CAHRISMA (Conservation of the Acoustical Heritage by the Revival and Identification of the Sinan’s Mosques’ Acoustics) project is to create an integrated 3D audio-visual system to conserve the architectural heritage, including both acoustical and visual characteristics, of the Sinan’s mosques and Byzantine churches. The realism of the reconstructed mosques can be increased by recreating life inside the architectural models.

The goal of the crowd module is to simulate a crowd of virtual humans able to move and interact within a real-time, photo-realistic simulation containing complex buildings [18]. In this case our focus is on the ability to control the scenario and the quality of the animation. The crowd module allows the construction of different scenarios involving scripted behaviours of the virtual humans, such as letting a group of worshippers enter the mosque, walk towards the area designated for praying and then start a praying sequence (see Figures 7, 8).

One of the challenges arising in this application was the orchestration of convincing animation sequences. From

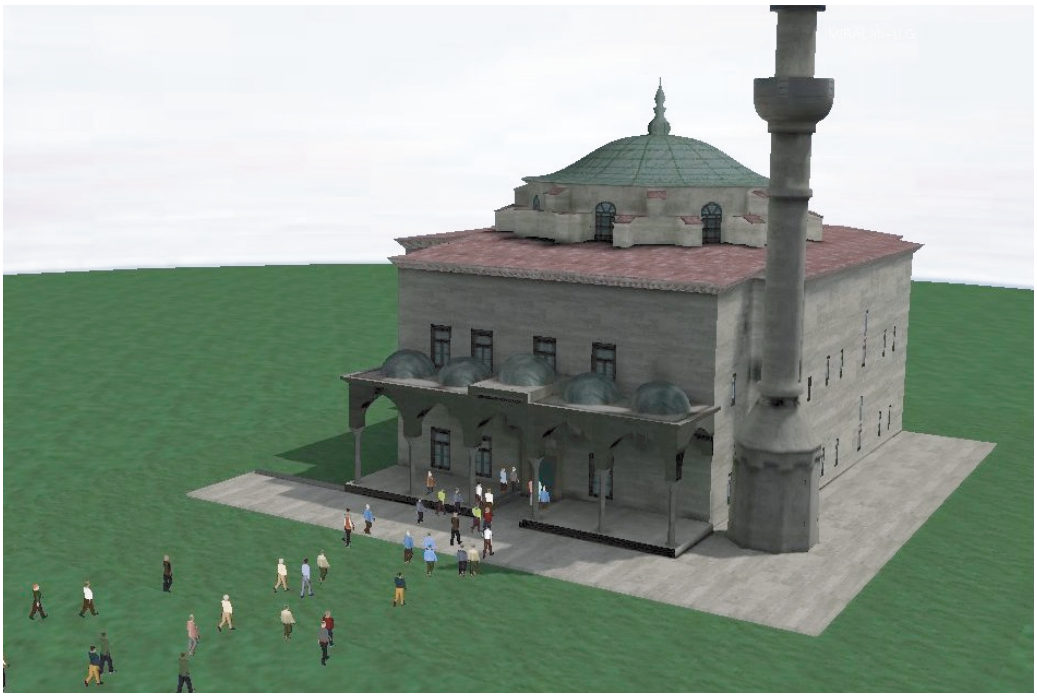

Figure 7: *Crowd entering the mosque.*

Figure 8: *Crowd performing praying sequence inside the mosque.*

the praying ceremony scenario arose the requirement for people to perform synchonous steps of the praying sequence. Observations of the real world praying sequence indicate that synchronicity is not perfect: there are small variations in the timing. Because it was not feasible to record each animation for each member of the crowd individually, a single sequence of motion-captured animation had to be reused. It was the task of the crowd module to create the illusion of variety, thus avoiding the mechanical appearance that would occur if each member of the crowd performed exactly the same action at exactly the same time.

We used the events and rules of the crowd system in conjunction with the ability to change the speed of the motion sequence for synchronization and desynchronization of the animation. After arriving at the area designated for praying, agents receive events telling them to start with the first part of the praying.

```
FOR  GROUP worshipers
WHEN EVENT = start_pray_1
THEN SCRIPT
      WAIT RANGE 0.0 2.0
      PERFORM_ACTION Pray1
      SEND_EVENT ready_to_pray_2
                 TO AGENT leader
```

The rule (see above) becomes triggered at the same time for every agent, but because they take different times to react, they start the animation action asynchronously. Varying the speed of the animation clip for individual members of the crowd gives an additional increase in the realism. After finishing the animation, agents send an event announcing the end of the step to one agent having the function of synchronization. After all agents have finished this step, the leader emits a command to proceed to the next one. In such a way the next step of the praying sequence does not start until everyone has finished the previous one.

9. Conclusions and future work

This paper presented our work on a crowd simulation system aimed at real-time applications. First we discussed challenges of real-time crowd simulation, especially the need to efficiently manage variety. We introduced the notion of levels of variety and defined a modular behavioural architecture for multi-agent simulations allowing the management of variety. We demonstrated the validity of our approach with two case studies, where our crowd system was used to manage crowd behaviour in a virtual reality training system and a virtual heritage site reconstruction.

For future work we plan to continue enhancing the levels of variety: for the animations we want to employ motion models able to synthesize variations of a given motion [19]. Another possible extension is to improve the behaviour model by incorporating new behaviours based on sociological observations from the real world gatherings.

Acknowledgements

We are grateful to Mireille Clavien and Alessandro Foni for the design of the virtual humans and the mosque; Jan Ciger and Marcello Kallmann for their work on the path-finding and collision-avoidance code; Michal Ponder, George Papagiannakis and Tom Molet for their support with the VHD++ framework. This work has been supported by the Swiss National Research Foundation and the Federal Office for Education and Science in the framework of the European projects CROSSES and CAHRISMA.

References

1. C.W. Reynolds. Flocks, herds, and schools: a distributed behavioral model. In *Proc. SIGGRAPH '87*, pages 25–34. 1987.

2. E. Bouvier and P. Guilloteau. Crowd simulation in immersive space management. In *Proc. Eurographics Workshop on Virtual Environments and Scientific Visualization '96*, Springer-Verlag, pages 104–110. 1996.

3. D. Brogan and J. Hodgins. Group behaviors for systems with significant dynamics. *Autonomous Robots*, 4:137–153, 1997.

4. A. Aubel and D. Thalmann. Real-time display of virtual humans: Level of details and impostors. *IEEE Transactions on Circuits and Systems for Video Technology*, 10:207–217, 2000.

5. F. Tecchia, C. Loscos and Y. Chrysanthou. Image-based crowd rendering. *IEEE Computer Graphics and Applications*, 22(2):36–43, 2002.

6. C. O'Sullivan, J. Cassell, H. Vilhjálmsson, S. Dobbyn, C. Peters, W. Leeson, T. Giang and J. Dingliana. Crowd and group simulation with levels of detail for geometry, motion and conversational behaviour. In *Proc. Third Irish Workshop on Computer Graphics*, Eurographics Ireland, Dublin, Ireland, pages 15–20. 2002.

7. C. McPhail, W.T. Powers and C.W. Tucker. Simulating individual and collective actions in temporary gatherings. *Social Science Computer Review*, 10(1):1–28, 1992.

8. G.K. Still. Crowd Dynamics, PhD thesis, Warwick University, 2000.

9. S.R. Musse and D. Thalmann. A hierarchical model for real time simulation of virtual human crowds. *IEEE*

Transactions on Visualization and Computer Graphics, 7(2):152–164, 2001.

10. B. Ulicny and D. Thalmann. Crowd simulation for interactive virtual environments and vr training systems. In *Proc. Eurographics Workshop on Animation and Simulation*, Springer-Verlag, pages 163–170. 2001.

11. S. Lloyd. Ultimate physical limits to computation. *Nature*, 406:1047–1054, 2000.

12. J.S. McClelland. *The crowd and the mob: from Plato to Canetti*. Unwin Hyman, 1989.

13. H. Seo, L. Yahia-Cherif, T. Goto and N. Magnenat-Thalmann. Genesis: Generation of e-population based on statistical information. In *Proc. Computer Animation 2002*. IEEE Press, 2002.

14. R. Boulic, P. Becheiraz, L. Emering and D. Thalmann. Integration of motion control techniques for virtual human and avatar real-time animation. In *Proc. VRST '97*, ACM Press, pages 111–118. 1997.

15. Taro Goto, S. Kshirsagar and N. Magnenat-Thalmann. Automatic face cloning and animation. *IEEE Signal Processing Magazine*, 18(3):2001.

16. B.-M. Straub, M. Gerke and A. Koch. Automatic extraction of trees and buildings from image and height data in an urban environment. In *Proc. International Workshop on Geo-Spatial Knowledge Processing for Natural Resource Management*, pages 59–64. 2001.

17. N. Farenc, R. Boulic and D. Thalmann. An informed environment dedicated to the simulation of virtual humans in urban context. In *Proc. Eurographics'99*, Blackwell, pages 309–318. 1999.

18. G. Papagiannakis, G. L'Hoste, A. Foni and N. Magnenat-Thalmann. Real-time photo realistic simulation of complex heritage edifices. In *Proc. Virtual Systems and Multimedia 2001*, pages 218–227. 2001.

19. I. Lim and D. Thalmann. Construction of animation models out of captured data. In *Proc. IEEE International Conference on Multimedia and Expo '02*. 2002.