Interactively Directing Virtual Crowds in a Virtual Environment

Tsai-Yen Li, Jian-Wen Lin, Yi-Lin Liu, and Chang-Ming Hsu
Computer Science Department, National Chengchi University
64, Sec.2, Chih-Nan Road, Taipei, Taiwan 11605, ROC
{li, s8512, s8506, s8505}@cs.nccu.edu.tw

Abstract

Simulation of emergent group behaviors for creatures such as birds and fishes has been widely used in computer animation. Although the same technique can be adopted to simulate crowds of virtual humans in a shared virtual world, it remains a great challenge to simulate the high-level intelligent behavior of a virtual human with planning capabilities. In this paper, we present a shared virtual environment crowded with real and virtual users. Virtual users, controlled by a world manager, are simulated in groups, each of which is led by an intelligent group leader. At run time a world manager can interactively assign a goal configuration to each group leader, and the system will automatically generate collision-free paths that bring the leaders to the goals. Since we allow multiple groups to move simultaneously in the virtual world, we adopt a decoupled path-planning approach, in which the paths being executed by the other leaders become the motion constraint of the current leader under consideration. The remaining members in a group follow the motion of the leader with emergent behaviors such as flocking. We believe that such an interactive interface will facilitate the simulation of controlled virtual crowds for applications such as 3D virtual shopping malls.

Key words: Virtual Crowd, Shared Virtual Environment, Decoupled Path Planning, Behavioral Animation, and Humanoid Simulation

1. Introduction

As 3D shared virtual environments are becoming prevalent in the cyberspace, the need for better authoring tools to direct groups of avatars also increase. For example, in a 3D virtual shopping mall, a well-controlled crowd of people will increase the realism of virtual shopping. The owners of virtual shops might want to hire crowds of virtual avatars to attract real users to their stores. However, most shared virtual environments today only accept real-user logins. Most of them do not have a flexible interface to adapt both virtual and real users. In addition, there are no good tools to quickly populate the world with virtual avatars that can be directed in an interactive manner.

In this paper we present a shared virtual environment (VE) system that allows coexistence of virtual and real users and enables interactive path planning for virtual crowds. The world is populated with groups of virtual users, controlled by a world manager. Each group is led by an intelligent group leader. At run time a world manager can interactively assign a goal configuration to a group leader through a graphical user interface at the VE server. The system will automatically generate collision-free paths that bring the leaders to their goals. Since we allow multiple groups to move simultaneously in a virtual world, the computational complexity of the involved path-planning problem is rather high. Therefore, we adopt a decoupled path-planning approach, in which the paths being executed by the other leaders become the motion constraint of the current leader under consideration. The remaining group members then take a more emergent strategy to follow their leaders.

We organize the remaining of the paper as follows. In the next section, we will review the researches pertaining to our work in artificial life and geometric planning. In the third section, we will give an overview of the architecture used in our VE system. In the fourth section, we will give a more detail description of the planning algorithm adopted in our system. Then we will present some implementation details and give some examples from our experiments. Finally, we will conclude the paper with some discussions on current limitation of our system and some possible future extensions.

2. Related Work

Simulation of emergent behaviors such as flocking has been widely used in creating realistic animations for groups of virtual creatures such as fishes or birds. [13][15] By applying simple emergent rules to each character, one can simulate realistic flocking behavior for animals. However, it is difficult to simulate a crowd of people simply with these principles because human, as an intelligent character, possesses higher degree of intelligence. In recent years, there have been many efforts in incorporating practical artificial intelligence techniques to create real-time animation. For example, a cognition model has been proposed in [7] to use a more complete control loop to simulate an intelligent character. Researches in virtual human also consider the problem of creating realistic humanoid group motions through vari-

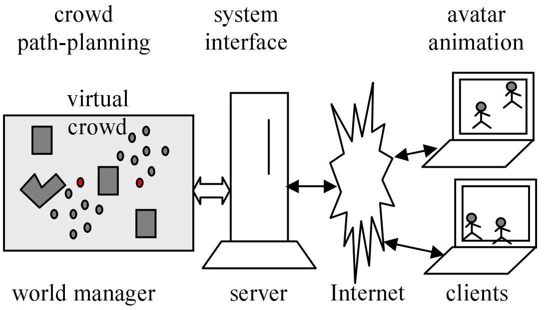


Figure 1. System architecture

ous levels of controls. [4][5][11][12] However, most of these researches do not account for geometric reasoning capability such as path planning. On the other hand, motion-planning techniques have been successfully adopted for automatic movie generation[8] or customized tour guiding[10], although they usually focus on generating dexterous motions for a single human only. In robotics, efficient motion planning algorithms have been proposed to control more than one robot arm in an on-line manner[9]. However, we have not seen similar work been applied to simulate human crowds. Distributed interactive simulation (DIS) and shared virtual environments (SVE) have also been active research fields recently. Most research efforts are focused on system scalability, transmission efficiency, and scene management. In recent years, more and more SVE systems (such as ActiveWorlds[1] and Blaxxun's[3]) include programming interfaces for implementing virtual avatars (or called *bots*). However, they do not have systematic ways to simulate virtual crowds.

3. Overview of System Architecture

Our system is based on an open-source virtual environment (VE) system (VNet)[17]. This VE system adopts a client-server model that uses VRML[16] as its front-end 3D user interface. We augment the system with three major software modules to facilitate the control of group avatars: *system interface* module, *avatar animation* module, and *crowd path-planning* module, as shown in Figure 1.

First, the system interface module creates an interface at the server side between the VE server and the simulator for a world manager to control the motions of virtual crowds. Through this interface, a virtual user is treated in the same way as a real user at the VE server. From the graphical user interface of a client machine, a virtual avatar cannot be distinguished from a real avatar as seen by another real user. Second, the avatar animation module uses a modified messaging protocol used in VNet to send parameterized animations to the clients. With this module, these clients can convert motion-captured data on the fly into humanoid animations conforming to the VRML humanoid version 1.1 standard. Third, the crowd path-planning module is the key component that generates the motions for each group leader directed interactively by a world manager. It also includes a leader-follower steering module with flocking behaviors to generate motions for the remaining group members. The path planning and coordinating methods for multiple group leaders are described in details in the next section.

4. Planning for Crowd Motions

4.1. Problem description

The goal of our system is to provide an interactive interface for directing virtual avatars in a virtual environment. We are given a 2D polygonal description of the obstacles in the virtual world. Each of our avatars has three DOFs (x, y, θ) when they move on a plane. The parameter space for each avatar, called the *Configuration Space* (or *C-space* for short), is denoted by C_i. The overall C-space for the whole system, denoted by C, is the composite space of each individual C-space (C_1xC_2x…xC_n). At any time during the simulation, our system has to make sure that the generated motions for the virtual avatars be realistic and safe. In other words, the motions must be continuous in C and collision-free from other avatars and the obstacles in the environment. A virtual avatar should never make a move that will cause a collision. However, even if a virtual avatar does not make an illegal move, we can not guarantee that a collision will not happen when a real avatar intends to do so. However, except for this kind of situation, it is the job of the planning system to ensure the virtual crowds under its control do not collide.

In order to reduce the complexity of the planning problem, we assume that each avatar can be represented by an enclosing circle of radius r. Due to the geometric symmetry of a circle, we can reduce the degree of freedom for each avatar to two by temporarily ignoring the θ-dimension. The value for θ will be computed in a post-processing step after a path has been generated. For example, we can require that an avatar always face the tangential direction of a path. In order to facilitate collision detections in the planning process, we use a discrete approach by representing the polygonal obstacles with a bitmap and then grow the obstacle boundary by r to form the C-space of each avatar. This computation only needs to be done once when obstacle configurations are determined. The possible collisions between avatars are detected at run-time by checking the distances between the avatars.

4.2. The approach

Although the problem of path planning has been widely studied for the past three decades, one still cannot escape the curse of dimensionality. The planning problem becomes difficult for systems with high degrees of freedom (DOFs) such as coordinating the motions of multiple mobile robots. The scenario of virtual crowds inherently also has such high complexity. For example, the dimension of the composite C-space (C) for the whole virtual avatars is $2m$, where m is the number of virtual avatars. Since the size of a C-space grows exponentially in the number of dimensionalities, a complete planning system deems to be infeasible. However, if we look at the problem of controlling virtual crowds interactively in a more practical manner, we can find several ways to simplify the planning problem and still make the solution interesting.

First, we assume that not every virtual avatar requires high-level planning. Instead, we assume that these simulated virtual avatars are in n groups. Each group, G_i consists of a leader, L_i, and a few followers, F_{ij} (where $i<=n$ is the index of a group and j is the index of a follower in its group). Only the leaders have path-planning capability and the followers will adopt local emergent rules to follow their respective leaders. With this approach, the number of avatars that require planning is greatly reduced, and the flocking behavior of a virtual crowd can also be achieved with traditional artificial life approaches.

4.3. Decoupled planning for group leaders

In robotics, the problem of path planning for multiple robots falls into two categories: *centralized* and *decoupled.* The centralized approaches consider the composite C-space of the whole system, which could be impractical to search exhaustively. On the other hand, the decoupled approaches usually only consider one robot at a time. In one such decoupled approach, each robot is planned independently and then their motions are then coordinated by velocity tuning techniques.[6] Another decoupled approach assumes that robot motions are generated sequentially and each robot is planned under the constraint of the robots whose motions are generated earlier.

In our crowd control system, we take the last decoupled approach by decomposing the overall planning problem for multiple virtual avatars into smaller subproblems. Each of these subproblems considers one virtual avatar at a time under the constraints of other avatars' motions. The same approach has been used in planning the motions of multiple robotic arms in an on-line manner.[9] Although this approach is not complete in nature, this planning scheme fits our application quite well since the needs for path planning happen sequentially. When the world manager directs a group leader by specifying a desired goal, the path planner is called on demand to generate a collision-free path based on the planned/scheduled motions of other group leaders.

At any time when the world manager would like to direct a group leader to a new goal configuration, the planner will try to generate a path that does not cause any collisions with other group leaders as well as the obstacles in the environment. The paths of these group leaders (denoted by τ_i) have been determined as a function of time t. We further extend these paths to infinite time by assuming that an avatar will stay at the last configuration of its path when the path is finished. This extended path is denoted by τ_i^*. In order to account for these constraints, we search for a collision-free path in a so-called *Configuration-Time space* (*CT-space*), which is formed by augmenting the C-space with the time dimension. A legal path τ_k in the CT-space is one that does not overlap with the environmental obstacles (*CB*) and τ_i^* (for i = 1 to n and $i \neq k$) as shown in Figure 2. Each τ_i^* induces a time-dependent obstacle to the virtual avatar under consideration. The objective of the path planner is to find a collision-free path in the CT-space that can connect the current and the goal configurations. For realistic simulation, the velocity of a virtual avatar must be within some reasonable limit; therefore, the slope of any point along a legal path in this CT-space must also be positive (time is not reversible) and less than some user-specified value.

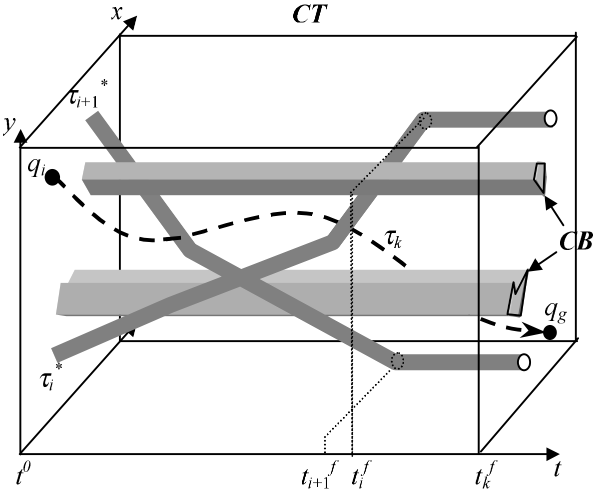


Figure 2. Searching for a feasible path in the CT-space

With the constraints mentioned above, the search in the CT-space is conducted in a best-first fashion based on the value of each configuration in an artificial potential field. This type of potential field is widely accepted as a good heuristic for motion-planning problems[2]. For efficiency consideration, we only construct a 2D potential field accounting for static environmental obstacles. The best-first algorithm returns a legal collision-free path when the search succeeds and gives up when all possible configurations have been visited. Note that a path is legal only if it can remain collision-free for the whole period when all other avatars are active. Therefore, we require that a goal configuration in the CT-space must have a time value that is equal to or greater than the latest finish time of all other virtual avatars. For instance, in Figure 2, t_k^f (the final time for τ_k) must be greater than t_i^f (which is greater than t_{i+1}^f).

4.4. Emergent behavior for followers

Human avatars' crowd motions are similar to other animals'. However, human avatars in a virtual crowd simulation may possess characteristics unique to human be-

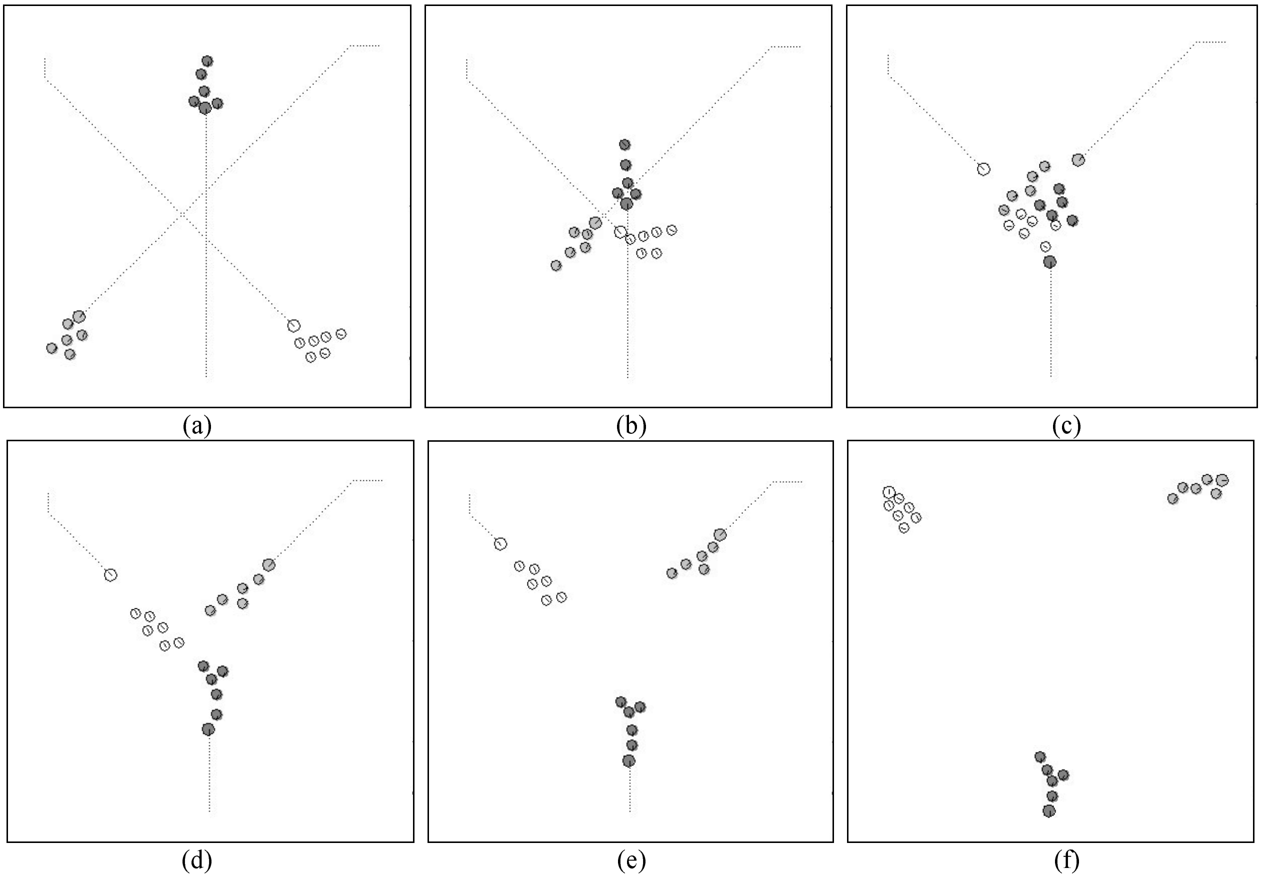

(a) (b) (c)
(d) (e) (f)

Figure 3. Snapshots showing three crossing virtual crowds

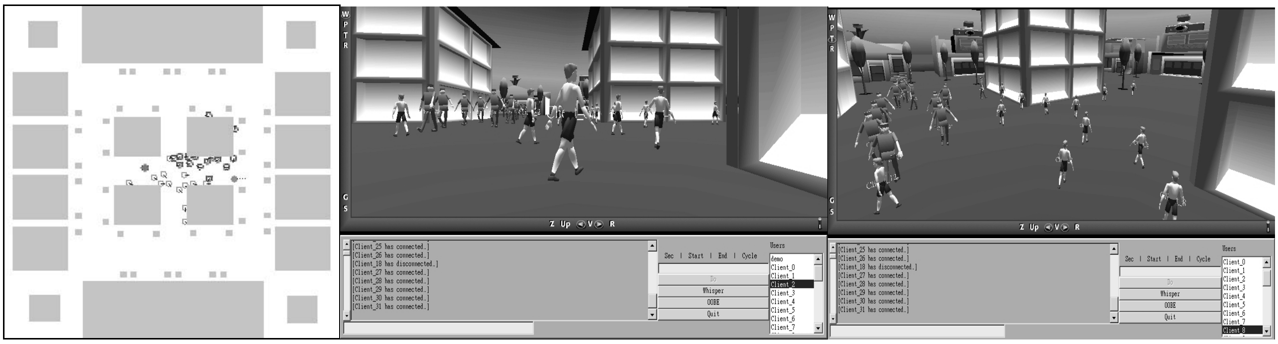
ings. For example, unlike other animals' flocking behavior where the leader-follower relation is formed automatically, a leader acting as a tour guide in a human group is specified explicitly. With the collision-free motions for the leaders generated in the previous subsection, we now describe how we generate the motions for the followers.

To simulate the human grouping behavior, we adopt a strategy similar to the one proposed in [14]. The strategy uses various attractive and repulsive forces to generate steering behaviors. In each control cycle of the simulation, an avatar perceives other avatars in its limited view cone and reacts by adjusting its velocity according to the composite force resulting from various steering and environmental criteria. For example, three steering forces (*separation*, *cohesion*, and *alignment*) were suggested to determine how an avatar reacts to other avatars in its local neighborhood. Separation force is computed according to the repulsive forces exerted by all of its nearby avatars within the view cone. Cohesion is computed by applying an attractive force from the average position of its neighbor avatars in the same group. Alignment is computed by averaging together the velocity of the nearby avatars of the same group. Note that only avatars in the same group exert the cohesion and alignment forces to each other while avatars in different groups can still affect each other with the separation forces. In addition to these three forces, we also apply a repulsive force to an avatar according to its distances from the nearby environmental obstacles. Furthermore, the leader of a group also applies a major attractive force to its followers. This attractive force, proportional to the distance from the leader, drives the followers to the leader even if the leader is not moving.

These five forces altogether are normalized and then reweighted before they are composed. The weight of each force is dynamically adjustable according to the current world status and the past history. For example, if the force causes a follower avatar to collide with an obstacle, the weight of the repulsive force from the obstacle will be increased. When the collision disappears, the weight for this force will incrementally go down to its nominal value. However, a follower still may bump into obstacles because the repulsive and attractive forces cancel each other. In this case, a sliding force along the obstacle boundary is applied to pull the followers toward the leader.

4.5. Examples

In Figure 3, we show an example of three virtual crowds interactively directed by a world manager to their respective goal. Environmental obstacles are not included in this example for clarity. The dashed lines in each figure show the remaining paths to be executed by the leaders. Although the traces appear to be overlapping with each



```
[Client_25 has connected.]
[Client_26 has connected.]
[Client_18 has disconnected.]
[Client_27 has connected.]
[Client_28 has connected.]
[Client_29 has connected.]
[Client_30 has connected.]
[Client_31 has connected.]
```

```
[Client_25 has connected.]
[Client_26 has connected.]
[Client_18 has disconnected.]
[Client_27 has connected.]
[Client_28 has connected.]
[Client_29 has connected.]
[Client_30 has connected.]
[Client_31 has connected.]
```

(a) (b) (c)

Figure 4. Graphical user interface of the shared virtual environment system with virtual crowds

other, the leaders do not collide with each other as the time advances. The virtual crowds start as three separated groups in Figure 3(a). They approach each other in Figure 3(b) when the leaders follow their paths to their goals. These three groups appear to be mixed in Figure 3(c) although the followers in each group are still following their leader. The three groups are separated again in Figure 3(d) although the followers are somewhat behind their leaders due to the delays resulting from the conflicts of performing group crossing. In Figure 3(e), the followers catch up their respective leaders (we allow the followers to move faster than the leaders). The leaders altogether with its followers finally reach their goals in Figure 3(f).

5. Implementation and Experiments

The aforementioned software modules have been fully implemented in Java based on the VNet shared VE system. A virtual world in VRML is created together with its 2D-layout map. This 2D map is an input to the path-planning module for simulating the virtual crowds. In Figure 4 we show sample screen dumps of this system with its 2D and 3D graphical user interface. The 2D interface is presented by a Java program at the server side for interactive crowd controls while the 3D interface is a VRML browser controlled by a Java applet that appears at the client machine.

In the path-planning module, the world is represented by a grid of 128x128. The same resolution is used in the CT-space for searching a feasible path. The time for planning the motion of each group leader is usually only fractions of a second. For example, in the example of Figure 3, the planning time for the path found for each leader in the three groups is 20ms and the computation time for the follower motions during the simulation are negligible. Therefore this kind of performance make the system well suited for our interactive directing purpose.

In our system, we have to ensure that the simulated virtual avatars do not cause any unsafe motions unless they are the intentions of a real avatar. Therefore, during the simulation, we let the leaders, whose motions need to be precisely synchronized, to have higher priorities in each step. The followers will then react according to the leaders' new configurations. However, since the motions of the followers are not planned, there are still no guarantees that they will not be blocking the leaders' ways. Similarly, if a real avatar intends to run into a virtual avatar, collisions are inevitable. Therefore, we perform collision checks for the leaders executing their paths at run time. Whenever a collision will occur in the next step according to the scheduled motion, the path will be cancelled, and the planner will be called again with the original goal and the latest world status.

Although the planner is capable of detecting unexpected collisions and replan accordingly, there exist situations where the path planner may fail to find solutions for the leaders to make moves. These solutions could actually exist but the planner fails to find one because of its incomplete nature of using a decoupled approach. However, in our experiments, this situation rarely happens unless we intend to test it on a pathological case. On the other hand, it is more often that a follower gets stuck at some location minimum of the composite steering force field. We think this situation is similar to the local minimum problem in potential-field-based motion-planning method. Although it is possible to construct local-minimum-free potential fields, it is too consuming to be used for on-line purpose. We are in the process of experimenting with other force fields that account for environmental obstacles in order to improve the situation.

6. Conclusions

In conclusion, we have proposed an interactive system for directing virtual crowds in real-time. The virtual avatars in a shared virtual environment can be controlled with high-level inputs via a graphical user interface. The system is capable of generating collision-free motions with flocking behavior in an avatar group. The planning capability and efficiency have been successfully demonstrated in a public-domain shared virtual environment system. We are also incorporating the planner into the ActiveWorlds VE system in order to simulate autonomous virtual crowds in a 3D shopping mall applications.

Acknowledgement

This work was partially supported by grants from the National Science Council, ROC, under contracts NSC89-2218-E-004-008 and NSC89-2218-E-004-009.

References

[1] ActiveWorlds, URL: http://www.activeworlds.com.

[2] J. Barraquand and J. Latombe, “Robot Motion Planning: A Distributed Representation Approach,” *International Journal of Robotics Research*, 10:628-649 (1991).

[3] Blaxxun Community Server, URL: http://www.blaxxun.com.

[4] T.K. Capin, I.S. Pandzic, N. Magnenat-Thalmann, D. Thalmann, "Integration of Avatars and Autonomous Virtual Humans in: Networked Virtual Environments", *Proceedings of. ISCIS 98, IOS Press*, pp. 326 – 333, Amsterdam, Netherlands, (1998).

[5] T. Capin, I. Pandzic, N. Magnenat Thalmann, D. Thalmann, "Avatars in Networked Virtual Environments", John Wiley & Sons, (1999).

[6] M. Erdmann and T. Lozano-Perez, “*On Multiple Moving Objects*,” AI Memo No. 883, Artificial Intelligence Laboratory, MIT, (1986).

[7] J. Funge, X. Tu, and D. Terzopoulos, “Cognitive Modeling: Knowledge, Reasoning, and Planning for Intelligent Characters,” *Proceedings of ACM SIGGRAPH*, pp29-38 (1999).

[8] Y. Koga, K. Kondo, J. Kuffner and J.-C. Latombe, “Planning motions with intentions,”, *Proceedings of ACM SIGGRAPH’94*, pp 395-408 (1994).

[9] T.Y. Li and J.C. Latombe, “Online Manipulation Planning for Two Robot Arms in a Dynamic Environment,” *International Journal of Robotics Research*, 16(2):144-167, (1997).

[10] T.Y. Li, J.M. Lien, S.Y. Chiu, and T.H. Yu, “Automatically Generating Virtual Guided Tours,” in *Proceedings of the Computer Animation '99 Conference*, Geneva, Switzerland, pp99-106, (1999).

[11] S.R. Musse, F. Garat, D.Thalmann, “Guiding and Interacting with Virtual Crowds in Real-time,” *Proceeding of Eurographics Workshop on Animation and Simulation '99 (CAS '99)*, pp.23-34, Milan, Italy, Springer, (1999).

[12] I.S. Pandzic, T.S. Capin, E. Lee, N. Magnenat-Thalamann, D. Thalmann "Autonomous Actors in Networked Collaborative Virtual Environments", *Proceedings of MultiMedia Modeling '98*, pp. 138-145, IEEE Computer Society Press, (1998).

[13] C. Reynolds, “Flocks, Herds, and Schools: A Distributed Behavioral Model, *Proceedings of ACM SIGGRAPH’87*, pp.25-34, (1987).

[14] C. Reynolds, “Steering Behaviors For Autonomous Characters,” *Proceedings of Game Developers Conference*, (1999).

[15] X. Tu and D. Terzopoulos, “Artificial Fishes: Physicas, Locomotion, Perception, Behavior,” *Proceedings of ACM SIGGRAPH’94*, (1994).

[16] VRML97 International Standard, URL: http://www.web3d.org/technicalinfo/specifications/vrml97/index.htm

[17] VNET, URL: http://www.csclub.uwaterloo.ca/u/sfwhite/vnet.