ARCrowd - A Tangible Interface for Interactive Crowd Simulation

Feng Zheng
University of North Carolina at Chapel Hill
CB #3175, Sitterson Hall
Chapel Hill, NC, USA
zhengf@cs.unc.edu

Hongsong Li
Beijing Institute of Technology
School of Software, Beijing Institute of Technology, Beijing, China
lihongsong@bit.edu.cn

ABSTRACT

Manipulating a large virtual crowd in an interactive virtual reality environment is a challenging task due to the limitations of the traditional user interface. To address this problem, a tangible interface based on augmented reality (AR) technology is introduced. With a novel interaction framework, the users are allowed to manipulate the virtual characters directly, or to control the crowd behaviors with markers and gestures. The marker-gesture pairs are used to adjust the environment factors, the decision-making processes of virtual crowds, and their reactions. The AR interface provides more intuitive means of control for the users, promoting the efficiency of user interface. Several simulation examples are provided to illustrate the various crowd control methods.

Author Keywords

Augmented reality, crowd simulation, crowd authoring, user interface.

ACM Classification Keywords

H.5.2 Information Systems: User Interfaces

General Terms

Design, experimentation, human factors.

INTRODUCTION

Most of the existing crowd simulation systems handle the virtual crowds with traditional interaction means on a 2D interface, i.e. using keyboard and mouse, which is capable of manipulating 3D objects with complex input metaphors. However, to design a realistic crowd scene often requires the users to interactively control the crowd behaviors, which demands even more complex operations. Consequently, it is difficult for non-technical users to fully grasp how to operate the complicated user interface.

A tangible augmented reality (Tangible AR) interface supporting interactive simulation and intuitive control of crowd behavior is presented to address the problem. Markers are used to control the spatial attributes, the behavior characteristics, and the environmental factors of the virtual crowds. With the presented interface, users are allowed to author and control virtual crowds in an interactive and efficient way, which also offers a more intuitive experience of interactions between the virtual and the real.

The demo video showing simulation examples is available at: http://www.cs.unc.edu/~zhengf/iui2011.html.

RELATED WORKS

As crowd behavior models become more and more sophisticated, the problem of creating and authoring crowd scenes become prominent. However, there are few progresses made on how to author crowd scenes efficiently and intuitively.

Anderson et al. controlled crowd behavior through adjusting constrains on their behavior [1]. In his method, virtual groups can move along curves, within a certain range or formations while moving. CrowdBrush, presented by Ulicny et al., employed a brush metaphor in an interactive graphical interface to add crowds and modify attributes of crowds for better user experience [2]. However, it only applies to crowd attributes related to spatial information and has limited control over time and events of simulation process. Sung et al. designed a situation-based control structure to control crowd behaviors. They devised a graphical interface for users to directly specify positions and range of situations, thus to achieve controls over crowd behavior [3]. All these crowd authoring techniques provide only 2D interactions and lack the use of 3D viewing and interaction techniques.

Augmented reality (AR) is a supplementation of the real world, and merges synthetic sensory information into a user's perception of a real world [4]. In recent years, AR techniques were widely adopted to improve the traditional user interfaces. 3DARModeler, proposed by Do and Lee, combined both traditional input method and the tangible input method (markers) for building 3D models [5]. Fujisawa and Kato used AR as a tangible interface for interactive fluid simulation [6].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IUI 2011, February 13–16, 2011, Palo Alto, California, USA.
Copyright 2011 ACM 978-1-4503-0419-1/11/02...$10.00.

To our best knowledge, there is no AR interface designed for interactive crowd simulation applications. Our goal is to take advantage of the intuitiveness of Tangible AR to design an easy-to-use interface for non-technical users.

INTERACTION DESIGN

There is no universal approach to control a virtual crowd. Instead of manipulating each virtual character, the user usually choose to control group's spawn points, destinations, reactions, motion paths, appearances, and environments. Since the AR interface only outputs the coordinate information of a fiducial marker, such as a position and an orientation, it is straightforward to control the spatial properties of the virtual groups and their environments with AR techniques. For the rest of the control factors, we propose an interaction framework that maps the user input on the AR interface to these factors.

The proposed AR interface consists of a large main cardboard as the work place, and fiducial markers. The markers are categorized into three classes:

- *Object-markers:* representations of virtual objects, such as virtual groups and obstacles;
- *Operator-makers:* used to adjust certain properties of a selected virtual group, such as behavior, appearance, path, etc.;
- *Control-marker:* used to control the simulation process only.

Figure 1. Setup of working environment.

Figure 1 shows the setup of our working environment. Through the AR interface, the users can control multiple markers to author a crowd scene and at the same time manipulate a camera to vary the angle of view. The camera captures the scene and passes the video stream to the AR interface. The AR interface then identifies the markers and passes the corresponding input information to a crowd simulation system. The user input information is further interpreted to controls of the virtual crowd. Finally, the crowd behavior is simulated and the 3D crowd scene is rendered to the AR interface. This setup is especially efficient in a cooperative working environment involving more than one user.

For a crowd scene including more than one virtual group, it is necessary to specify which group to receive the user input. Selection, as an important interaction operation, must be implemented. To do that, we implemented a simple *"closest-marker"* rule that the *operator-marker* always operate on the closest *object-marker* (a representation of a virtual group). A white line is drawn between the centers of two markers to demonstrate the selecting relation.

Operator Markers	Gestures	User Operations
"create a group"	pinch open / pinch close	create a virtual group with a given number of virtual characters
"switch behavior"	tap	switch the behavior of the selected virtual group
"switch clothes"	tap	switch the textures of the selected virtual group
"switch low-level behavior"	tap	switch the low-level behavior of the selected virtual group
"follow a path"	tap	the virtual group moves along a given path
"control-marker"	double tap	begin or end the simulation

Table 1. Marker-gesture pairs.

However, the presence of both an *object-marker* and an *operator-marker* are not sufficient for complex user input such as adjusting the number of virtual characters in a group. To solve the problem, we abstract the user inputs as *marker-gesture* pairs, as shown in Table 1. To control a virtual group, the user not only needs to position an operator-marker close to an object-marker, but also needs to make certain gesture on the operator-marker, such as tap or double-tap.

In summary, we propose an interaction framework that includes three stages, as shown in Figure 2. In the first stage, the input processing stage, the makers and the gestures are identified, and the coordinate information of the markers is registered. All input events are then passed to the second stage, the interpretation and simulation stage. For direct manipulation of the *object-markers*, the coordinate information is directly passed to the rendering stage to update the position of corresponding virtual objects. For crowd behavior controls, the *marker-gesture* pairs are identified and further interpreted as input information of a crowd behavior model. With various *marker-gesture* pairs, the users are allowed to control various aspects of crowd behaviors. The simulation result of the crowd behavior model is the reaction of each virtual character in a virtual group to the input information. At last, the simulation results are passed to the rendering stage to generate 3D crowd scenes.

SYSTEM IMPLEMENTATION

Interactive Crowd Simulation System

In this work we first implemented an interactive crowd simulation system to illustrate the proposed user interface. The simulation system is based on a crowd behavior model. There are many available crowd behavior models that simulate various crowd behaviors. Our aim is to have a crowd model that is simple enough to allow real-time simulation of crowd and to allow behavior control through the proposed interface. Our behavior model contains three layers: the perception layer, the behavior layer, and the reaction layer. The perception layer refers to crowd's motivation and knowledge of their environments, includes obstacles, events, and so on. The behavior layer describes the decision-making process by analyzing environment knowledge, and drives behaviors to change according to knowledge and motivation. The reaction layer is at the bottom of the three-layer model, which specifies all the actions of a certain behavior. In short, the model transforms motivations to act in the second layer to body actions of virtual-characters.

The system is implemented on a VR engine (Virtools 4.0). The interface includes both a WIMP interface and the proposed AR interface (based on ARToolkit). The WIMP interface is used for system configuration and data management only. The AR interface receives the input events and simulates reactions of the virtual groups to these events. The motion of the virtual groups for the next frame is then determined. The VR engine handles the simulation process and the scene management, the skeletal animation, and finally the scene rendering.

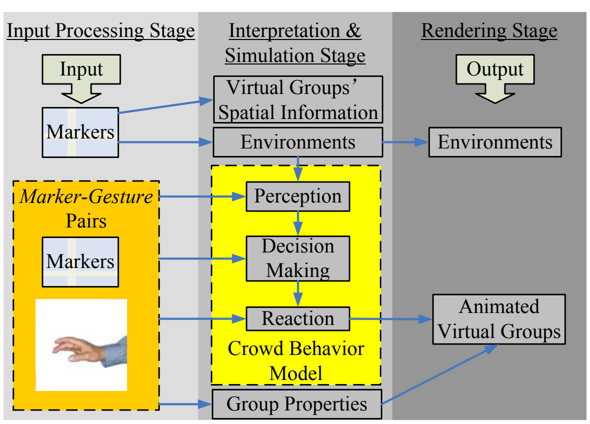

Figure 2. An interaction framework.

Gesture Identification

For each frame of the input video stream, the AR interface identifies the markers and registers their coordinate information. These input events are treated as raw input events without time features. To identify a gesture, an event queue is implemented to store the raw input events. Thus, a series of raw event queues can be analyzed and translated into a high-level input event-"gesture". For example, a marker disappears for several frames can be translated as a *"tap"* gesture, in which the user's finger blocks the marker for a short period.

The raw input events of a marker contain more information than those of a mouse, such as 3D coordinates and orientations. Therefore, it is possible to identify many user gestures such as "*shaking*" and "*rolling*". However, identifying many gestures may easily run out the time budget for each frame. Therefore, we only identify several simple gestures such as "*tap*" and "*double tap*".

User Operations

Group Creation Operations

To create a group of virtual-characters, the users need one *object-marker* that represents a virtual character and one *"create a group"* operator marker. The first one creates a single virtual-character and the second marker adjusts the number of the virtual-characters created for this group by varying the distance between the two markers, which is interpreted as "*pinch open*" or "*pinch close*" gestures, as shown in Figure 3(a). The AR interface determines the target of operation for the *"create a group"* marker with a proximity query, and computes the distance between the two markers for each frame, and creates the virtual-characters in a preset formation. The number of virtual-characters created is proportional to the square of the distance. The formation could be square, rectangle, circle, or random formations.

Attributes Customizing Operations

After created, a group of virtual-characters should be customizable to various textures (attires) and accessories, in order to change its external appearance. For convenience, all the options of an attribute should be setup in advance. To switch textures, the user needs to place a *"switch clothes"* marker near to the selected group, and uses his finger to tap the marker. The AR interface identifies the gesture and switch the attribute value to its next option. Figure 3(b) shows the result of switching the texture of a group with a *"switch clothes"* marker. More attributes can be adjusted in the similar means. One limitation of this attribute customizing operation is that the attribute values have to be discrete and set in advance.

Events and Reactive Behavior Control

For autonomous groups that react to the events without user interference, the users can control their behavior by controlling the events and the reactions to these events. Similar to the situation-based control over crowd behavior [3], the users of AR interface can control the environment factors by directly grabbing and placing an object-marker in real environments, which represents obstacle or hazard. Additionally, controlling the crowd reaction is critical for creating more complex crowd behaviors. For example, in a fire situation, the civilians would flee way from the fire, but the firefighters would run directly to the fire. It is essential to allow the users to adjust the reactive behaviors of the chosen groups interactively. In Figure 3(c-d), we show an example of fire-crowd interactions. Figure 3(c) shows that a burning fire (positioned by an object-marker) triggered a reactive behavior of three groups of virtual-characters (created with three object-markers), *"flee from fire"*; as a result, all groups were trying to flee from the fire. Subsequently, a behavior-marker was to change the behavior of a selected group from *"flee from fire"* to *"head to fire"*, as shown in Figure 3(d).

Guiding Crowds

For less autonomous group that requires guidance from the users, the users need to specify the path of the selected group, as well as their low-level behaviors. To do so, several object-markers placed in the real scene can act as anchor points of a path. With these markers, a smooth 3D curve can be generated by cubic spline interpolation. Then a *"follow path"* marker can be presented to switch the behavior of a selected group from its innate behavior to the new behavior. Figure 3(e) shows the object-markers used as the anchor points and a *"follow path"* operator-marker triggered the path-following behavior. During the group marching along the 3D path, the user interactively adjusted the markers to change the shape of the path, as shown in Figure 3(f).

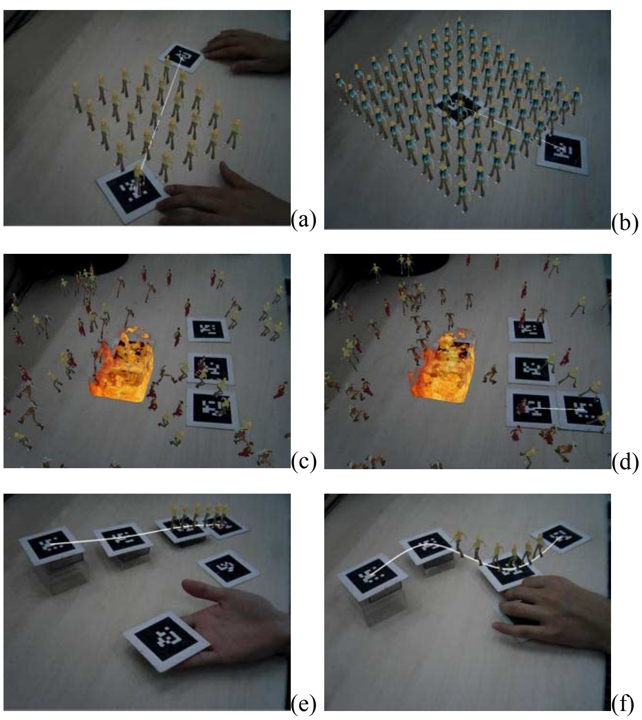

Figure 3. User operations: (a) group creation; (b) attributes customizing; (c-d) events and reactive behavior control; (e-f) group guiding.

SUMMARY AND FUTURE WORK

In this paper, a tangible user interface based on AR techniques is presented. With a novel interaction framework, the interface enables the users to create a crowd scene by manipulating the virtual character directly, or by controlling the crowd behaviors with markers and gestures. Thus, the benefit of our work is to provide an intuitive 3D interface for authoring and interacting with virtual crowd in an interactive environment. We tested the proposed interface on authoring several crowd scenes, including emergency evacuation and fire in a small town, as shown in the demo video. We conclude that it is convenient for non-technical users to design crowd scenes with the proposed interface.

One limitation of the AR-based interfaces is that it only works well for editing the spatial features of crowds, but its ability to adjust temporal aspects is limited. Therefore, one direction of the future works is to extend the AR interface for better timeline management. The other direction is to develop an interaction model of AR interface, especially for crowd control. Furthermore, we plan to conduct a user study to formally evaluate the usability of the proposed the interface, and solicit improvement suggestions.

Although the AR interface is build on top of a crowd simulation system we implemented, the other crowd simulation system could easily adopt this interfacing technique. Thus, the application scope of the AR interface can be much broader. The proposed AR interface after future improvements can be applied to various areas, such as public safety, city planning, and special effects.

REFERENCES

1. Anderson M, Mcdaniel E, Chenney S. Constrained Animation of Flocks, *Eurographics/ACM SIGGRAPH Symposium on Computer Animation, ACM SIGGRAPH*. New York: ACM Press (2003), pp.286-297.
2. Ulicny B, Ciechomski P, Thalmann D. Crowdbrush: Interactive Authoring of Real-time Crowd Scenes, *Eurographics/ACM SIGGRAPH Symposium on Computer Animation, ACM SIGGRAPH*. New York: ACM Press (2004), pp. 243-252.
3. Sung M, Gleicher M, Chenney S. Scalable behaviors for crowd simulation. *Computer Graphics Forum*, 23(3) (2004), pp.519-528.
4. Azuma R. T. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments 6:4* (1997), pp.355-385.
5. Do T. V. and Lee J.-W. 3DARModeler: a 3D Modeling System in Augmented Reality Environment. *International Journal of Electrical, Computer, and Systems Engineering* (2010) 4:2.
6. Fujisawa M. and Kato H. Interactive Fluid Simulation Using Augmented Reality Interface. R. Shumaker (Ed.): Virtual and Mixed Reality, LNCS 5622 (2009), pp. 431–438.