REAL-TIME CHARACTER ANIMATION USING PUPPET METAPHOR

Yoshihiro Okada
Graduate School of Information Science and Electrical Engineering, Kyushu University
6-1 Kasuga-koen, Kasuga, Fukuoka, 816-8580 Japan
e-mail: okada@i.kyushu-u.ac.jp, Phone: +81-92-583-7632, Fax: +81-92-583-7632

Abstract: For computer animation creation, character design is a very important factor but very hard work. Especially its motion design is very laborious work. So the authors propose new motion design method using a puppet metaphor and this paper introduces its prospective application examples. The puppet show is one of very popular entertainments. Puppets are not real characters like human actors/actresses and they cannot take real actions. However the puppet show has enough entertainment aspects. Moreover, the puppet is also children toy and it is easy to manipulate for even children. Then the authors employ the puppet metaphor for motion design of computer animation creation.

Key words: Character animation, Puppet show, Virtual puppet, Virtual Reality, Interface

1. INTRODUCTION

Advances in recent computer hardware technology have made possible 3D rendering images in real-time. However, it is still difficult for end-users to develop 3D graphics software. For this reason, Okada and Tanaka developed a 3D prototype system called *IntelligentBox* [1]. *IntelligentBox* is a component based construction system. Its application fields include various kinds, e.g., 3D game construction [2, 3], collaborative virtual environment construction [4] and so on. *IntelligentBox* also has aspects as an interactive animation system [1, 5].

For computer animation creation, character design is a very important factor but very hard work. Especially its motion design is very laborious work. So this paper proposes new motion design method using a puppet metaphor and introduces its prospective application examples, e.g., CG puppet show theatre, 3D fighting game etc. The puppet show is one of very

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35660-0_65
R. Nakatsu et al. (eds.), *Entertainment Computing*

popular entertainments. Puppets are not real characters like actual human actors/actresses and they cannot take real actions. However the puppet show has enough entertainment aspects. Moreover, the puppet is also children toy and it is easy to manipulate for even children. For this reason, we employ the puppet metaphor for motion design of computer animation creation. We have developed fundamental mechanisms for such a virtual puppet using *IntelligentBox.* As mentioned above, *IntelligentBox* is a component based construction system that provides various functional components called *boxes*. The proposed virtual puppet is also developed as a composite *box*. *IntelligentBox* also provides a particular component called *RoomBox* [4]. Using *RoomBox*, our virtual puppet will become available through *Internet.* Strictly speaking, multiple users using a different computer can play his/her CG represented puppet collaboratively. We use a data-glove and one magnetic-based motion sensor as controlling devices for the puppet. Then the user controls his/her puppet dynamically and interactively. Its application fields include interactive game development beside computer animation creation as described in section 4.

There are many researches on motion generation for computer animation. Witkin and Kass proposed concept of *spacetime* constraints [6]. After that, many research papers based on *spacetime* constraints were published [7]. IK (Inverse Kinematics) is one of the other popular methods for efficient motion generation. The motion path functionality is also a popular technique to intuitively define movement of a character's center of mass. Furthermore, the use of motion capture systems has become common to generate motion data for computer animation creation. Noser and Thalmann proposed virtual tennis game environment using a full-body motion capture system as a real-time motion input interface [8]. In this paper, we propose a real-time motion design method for character animation using only one motion sensor and a data-glove. We have never met such approaches.

The remainder of this paper is organized as follows. Section 2 explains essential mechanisms of *IntelligentBox.* Section 3 describes fundamental mechanisms of the puppet. Section 4 shows several prospective application examples. Finally Section 5 concludes this paper.

2. OVERVIEW OF INTELLIGENTBOX

As introduction of *IntelligentBox*, this section explains its essential mechanisms briefly.

2.1 Model-Display Object (MD) Structure

As shown in Figure 1, each *box* consists of two objects, a *model* and a *display object.* This structure is called an *MD* (*Model-Display object*) structure. A *model* holds state values of a *box.* They are stored in variables called *slots.* A display object defines how the *box* appears on a computer screen and defines how the *box* reacts to user operations.

Figure 1 also shows messages between a *display object* and a *model.* This is an example of a *RotationBox.* A *RotationBox* has a *slot* named 'ratio' that holds a rotation angle value. Through direct manipulations on a *box*, its associated *slot* value changes. Furthermore, its visual image simultaneously changes according to the *slot* value change. In this way a *box* reacts to a user's manipulations according to its functionality.

2.2 Message-sending Protocol for Slot Connections

Figure 2 illustrates a data linkage concept among *boxes.* Each *box* has multiple *slots.* Its one *slot* can be connected to one of the *slots* of other *box.* This connection is called a *slot connection.* The *slot connection* is carried out by three messages when a parent-child relationship exists between two *boxes.* They are a *set* message, a *gimme* message and an *update* message. These messages have the following formats:

(1) Parent box *set* <slotname> <value>.
(2) Parent box *gimme* <slotname>.
(3) Child box *update.*

A <value> in a format (1) represents any value, and a <slotname> in formats (1) and (2) represents a user-selected *slot* of the parent *box* that receives these two messages.

A *set* message writes a child *box slot* value into its parent *box slot.* A *gimme* message reads a parent *box slot* value and sets it into its child *box slot.* *Update* messages are issued from a parent *box* to all of its child *boxes* to tell

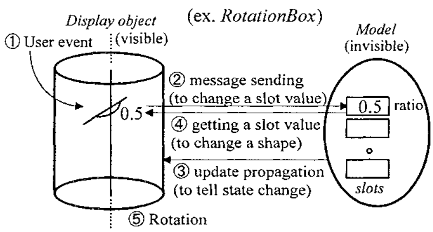

Figure 1. An *MD* structure of a *box* and its internal messages

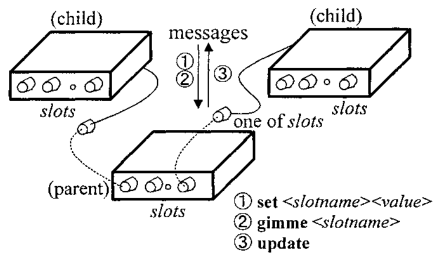

Figure 2. Standard messages between *boxes*

them that the parent *box slot* value has changed. In this way, these three messages connect a child *box slot* and its parent *box slot*, and combine their two functionalities.

3. REAL-TIME PUPPET CONTROL

Our proposed puppet is controlled using a data-glove and one magnetic-based motion sensor device. Then only one-hand motion manipulates the puppet.

3.1 Component Structure of Puppet Model

Figure 3 (left) shows components of a typical puppet model. This model consists of 17 joints. Each joint is a *3DRotationBox*. *3DRotationBox* has three DOF (Degrees Of Freedom) and then it rotates along x,y,z-axes. *IntelligentBox* provides a particular *box* called *SGJBox*. This *box* receives data sent from a data-glove device. We use Nissho Electronics Corporation's Super Glove Jr. as shown in Figure 3 (right). This device generates ten joint angles data. Each of these angles is applied to some specific joints of the puppet. Then the real-hand motion controls the puppet motion in real-time. *IntelligentBox* also provides another particular *box* called *ISOTRACKBox*. This *box* receives data sent from a magnetic based motion sensor, Polhemus Inc. 3SPACE ISOTRACK II. This device generates one six degrees of freedom data. The position and orientation of the puppet change according to this data.

As mentioned above, the puppet model, a human-like model, has 17 joints. However, the data-glove generates only ten joint angles data. To control the puppet motion by only one-hand motion, it needs a certain

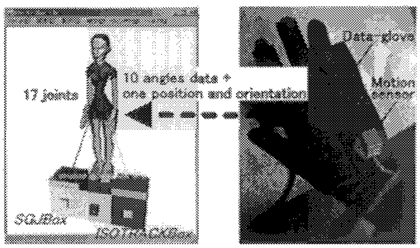

Figure 3. Puppet model and its control interface

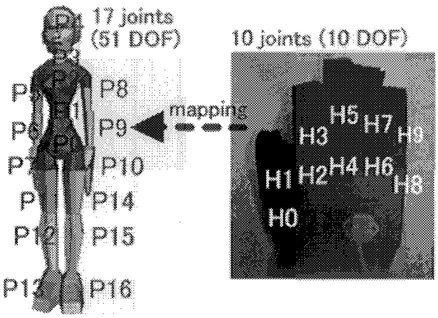

Figure 4. Puppet joints and hand joints

mapping scheme between 17 joints of the puppet and ten angles data of the data-glove as shown in Figure 4. One of our research purposes is to define optimum mapping between the data-glove data and the puppet model's joints angles for controlling the puppet easily and accurately.

3.2 Motion Control Example

Figure 5 shows four poses of the hand and their four corresponding poses of the puppet. We define Pose 1 is an initial pose since a paper shape of the hand seems more natural rather than a stone shape. In this example case, a mapping scheme is specified as shown in Table 1. A Table column 'PJoint' indicates each joint of the puppet, and 'HJoint' indicates each joint of the hand. For example, Joint P11 and P12, a right upper leg and lower leg of the puppet, are controlled by a little finger of the hand. Strictly speaking, each joint of the puppet has x,y,z-angle values so that only x-angle of P11 is controlled by H8 angle value. X-angle of P12 is also controlled by H9 angle value. However, its direction is opposite to P11 since its 'Dir' column is '-'. Actually the following substitutions (1) and (2) are done.

$$P11_x = H8 \quad (1) \qquad P12_x = -H9 \quad (2)$$

Here, $P11_x$ means x-angle of joint P11, and $P12_x$ means x-angle of joint P12.

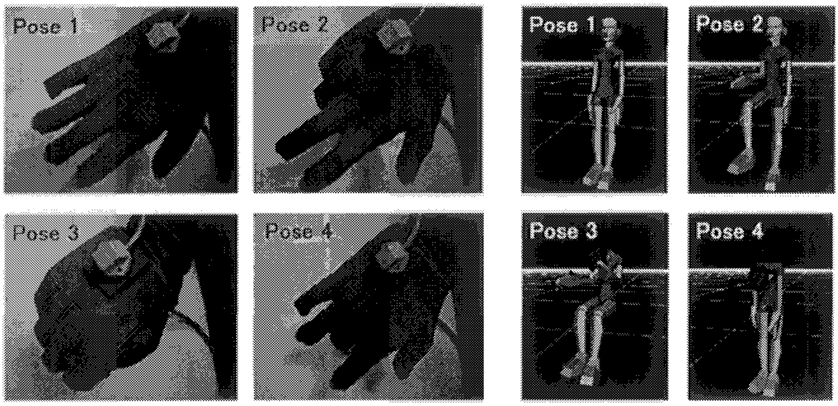

Figure 5. Four one-hand poses and their corresponding puppet poses

Table 1. Mapping table example

PJoint	Dir.	HJoint		PJoint	Dir.	HJoint
P0				P9	+	H3
P1				P10		
P2	+	H4		P11	+	H8
P3	+	H5		P12	-	H9

P4				P13		
P5	+	H6		P14	+	H0
P6	+	H7		P15	-	H1
P7				P16		
P8	+	H2				

In this way, this system becomes applicable to various applications by providing multiple mapping tables each of which is suitable for each different application.

3.3 Puppets except Human-like Model

IntelligentBox also provides an FFD (Free Form Deformation) function for soft object animation as a particular *box* called *FFDControlBox*. For example, Figure 6 shows Triceratops animation using *FFDContorlBox*. Its detail is written in the paper [1]. Using this *box*, the user can employ any arbitral shaped model as his/her puppet.

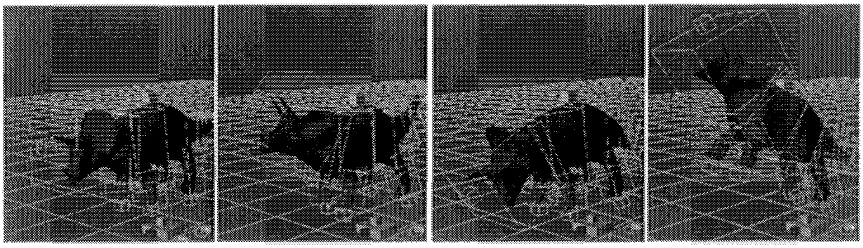

Figure 6. Triceratops animation using *FFDControlBox*

4. APPLICATION EXAMPLES

This section briefly introduces several prospective application examples especially included in the entertainment field.

4.1 CG Puppet Show Theatre

As described previously, *IntelligentBox* has provided a network communication facility as a particular *box* called *RoomBox* [4]. As shown in Figure 7, *RoomBox* manages user operation events and virtually provides a shared 3D space in which several users can work collaboratively. The composite component shown in the figure 3 is also available in such a shared 3D space. Then several users using a different computer can control his/her puppet in the shared 3D space simultaneously and play collaboratively. If they play a given role of an actor/actress, this means the CG puppet show.

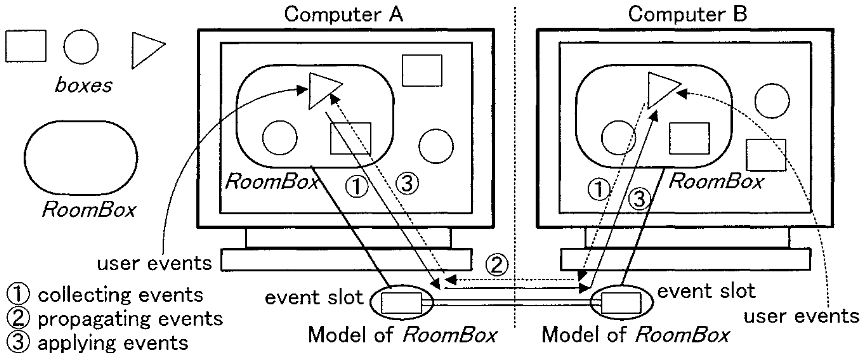

Figure 7. Message flow between two *RoomBoxes* for network collaboration

4.2 Interface for 3D Video Games

Sega *VirtuaFighter*™ is a very famous video game as a 3D fighting game. To specify fighting actions and to prepare a mapping table for them will make possible to develop a 3D fighting game as shown in Figure 8. This prospective game will accept the user's hand gesture as input data for controlling his/her fighter instead of a joystick.

Sega *SpaceChannel5*™ is another video game, a musical action adventure game. The main aim of this game is to control a player's doll to take the same action/motion as enemy's action/motion like a dance. This prospective game will accept the user's hand gesture as input data for controlling his/her doll instead of a joystick.

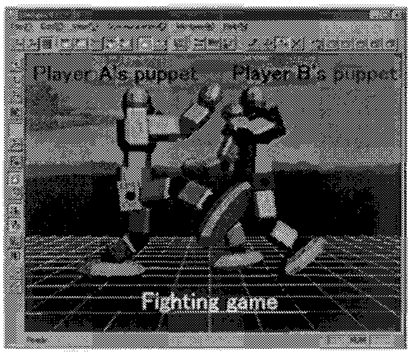

Figure 8. Image of 3D fighting game

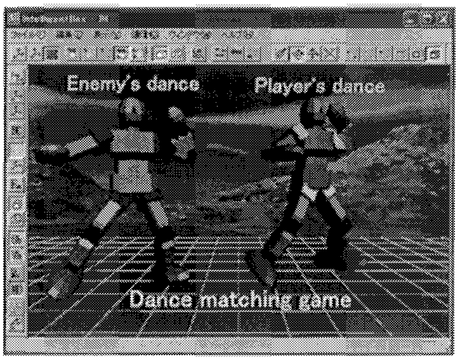

Figure 9. 3D game like SpaceChannel5™

5. CONCLUDING REMARKS

This paper proposed new motion design method using a puppet metaphor and introduced its prospective application examples especially included in the entertainment field. Actually we have to develop such application examples to clarify usefulness of a puppet metaphor. However we have not yet done it since we have just started this research. Currently we are developing such application examples and we will evaluate our proposed puppets. We will soon report its results and new findings.

ACKNOWLEDGEMENT

This work is partially supported by research grant of Ministry of Education, Culture, Sports, Science and Technology of Japan.

REFERENCES

[1] Okada, Y. and Tanaka, Y., IntelligentBox: A Constructive Visual Software Development System for Interactive 3D Graphic Applications, Proc. of Computer Animation "95, IEEE Computer Society Press, pp. 114-125, 1995.

[2] Okada, Y., Itoh, E. and Hirokawa, S., *IntelligentBox*: Its Aspects as a Rapid Construction System for Interactive 3D Games, Proc. of First International Conference on Intelligent Games and Simulation, SCS Publication, pp. 22-26, 2000.

[3] Okada, Y., Component Based Motion Editing Environment for Game Character Design, Proc. of Second International Conference on Intelligent Games and Simulation, SCS Publication, pp. 22-26, 2001.

[4] Okada, Y. and Tanaka, Y., Collaborative Environments in IntelligentBox for Distributed 3D Graphics Applications, The Visual Computer (CGS special issue), Vol. 14, No. 4, pp. 140-152, 1998.

[5] Okada, Y. and Tanaka, Y., IntelligentBox: Its Aspect as an Interactive Animation System, Proc. of SCI'99/ISAS'99, Vol.2, pp. 198-201, 1999.

[6] Witkin, A. and Kass, K., 1988 : Spacetime constraints, Proc. of SIGGRAPH'88, pp. 159-168, 1988.

[7] Gleicher, M., Motion editing with spacetime constraints, Proc. of SIGGRAPH'97, pp. 139-148, 1997.

[8] Noser, H. and Thalmann, D., Sensor Based Synthetic Actors in a Tennis Game Simulation, Proc. of Computer Graphics International '97, IEEE Computer Society Press, pp.189-198, 1997.