Presentation of Human Action Information via Avatar: From the Viewpoint of Avatar-Based Communication

Arita, Daisaku
Department of Intelligent Systems, Kyushu University

Taniguchi, Rin-ichiro
Department of Intelligent Systems, Kyushu University

出版情報：Lecture notes in computer science. 3683, pp.883-889, 2005-09. Springer Berlin, Heidelberg
バージョン：

Presentation of Human Action Information via Avatar: From the Viewpoint of Avatar-Based Communication

Daisaku Arita and Rin-ichiro Taniguchi

Department of Intelligent Systems, Kyushu University
6-1, Kasuga-koen, Kasuga, Fukuoka, 816-8580, Japan
{arita,rin}@limu.is.kyushu-u.ac.jp

Abstract. This paper describes techniques to present human action information on an avatar-based interaction system, using real-time motion sensing and human action symbolization. Avatar-based interaction systems with computer-generated virtual environments have difficulties in acquiring user's information, i.e., enough information to represent the user as if he/she were in the environment. This mainly comes of high degrees of freedom of human body and causes the lack of reality. Since it is almost impossible to acquire all the detailed information of human actions or activities, we, instead, recognize, or estimate, what kind of actions have occurred from sensed human motion information and other available information and re-generate detailed and natural actions from the estimated results. In this paper, we describe our approach, Real-time Human Proxy, especially on representing human actions. Also we present experimental results.

1 Introduction

There are several researches on virtual environments for distant interaction. In these researches, a 3-D virtual space is reconstructed, in which each participant is represented as an avatar by computer graphics techniques. Through the reconstructed virtual space, each participant sees and hears other participants' activities from the position where his/her avatar is represented. Therefore, it is called avatar-based interaction.

In avatar-based interaction, an avatar is expected to reflect activities of a participant into a virtual space as if he/she were there. Nevertheless, legacy input devices, such as a keyboard and a mouse, are not sufficient to acquire participant's activities in aspects of quality and quantity. Using such devices, a participant has to intentionally keep feeding their own activities into a system by hand, and acquired information may be neither precise nor rich. To solve this problem, as an input device, we use a vision-based motion capture system (MCS)[1]. Using the MCS, we can acquire rich information of participants without compelling them to do annoying operations.

According to the above idea, we have proposed a concept of Real-time Human Proxy (RHP), which acquires, symbolizes, transfers and represents human information for avatar-based interaction[2]. As the first step of RHP, we focus on nonverbal, or body movement information of humans. In this paper, we discuss Real-time Human Proxy, especially presentation of human action information, i.e., an avatar generation mechanism of RHP.

R. Khosla et al. (Eds.): KES 2005, LNAI 3683, pp. 883–889, 2005.

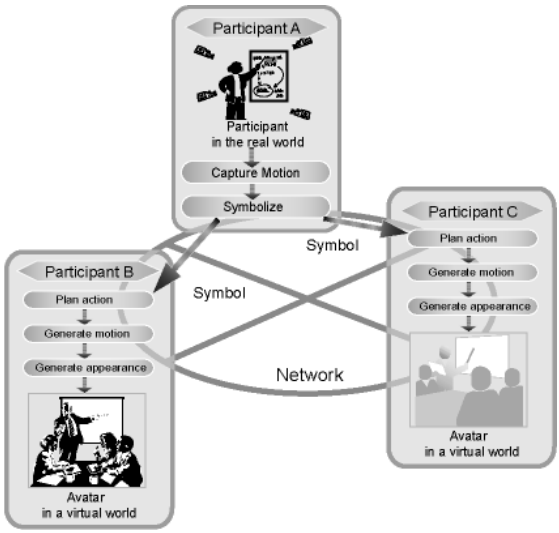

Fig. 1. The concept of RHP

2 Real-Time Human Proxy

Real-time Human Proxy (RHP) is a new concept for avatar-based interaction, which makes avatars act more meaningfully, or expressively, referring to action information acquired by a motion capture system. As the first step, we currently focus on acquisition and representation of human action, or nonverbal information. In the acquisition process, we symbolize the human action information under a given communication environment, such as classroom. In the presentation process, the symbols acquired are presented, or visualized, which are augmented based on the knowledge of the environment. Figure 1 shows a concept of RHP.

The important considerations behind the symbolization are summarized as follows:

- The important aspect of avatar-based communication is that an avatar, or an appearance of a human, can be changed depending on the purpose of communication, attendance, etc. However, only with raw data of human motion, such as motion vectors of body parts, which are acquired by a motion capture system, only an avatar with the same physique as an observed human can be presented. It is quite difficult to present avatars with different physique or avatars with different body structure.
- By a motion capture system, very detailed motion information can not be extracted such as hand postures, face expression at the same time. Such details often express intention and are important for communication. Therefore, here, we interpret, with the aid of knowledge of the purpose communication, limited motion information into intentions of communication, i.e., *symbols*. In presenting the symbols, we can visualize an avatar so as to express the intentions efficiently, i.e., generate detailed motions which are not acquired by a motion capture system.
- The symbolization is also quite helpful to compress the amount of data transfer and to improve QoS (Quality of Service).

Symbolization. On RHP, we acquire human actions instead of human motions. We categorize motion sequences into pre-defined actions, expressing them as symbols. Each symbol is formed by a label of an action and its parameters, such as "walking (p_x, p_y, ν_x, ν_y)" where p_x and p_y are the position, ν_x and ν_y are the velocity of a participant. After recognizing human actions from captured motion data, the system transfers the symbols to the representation side of a virtual space.

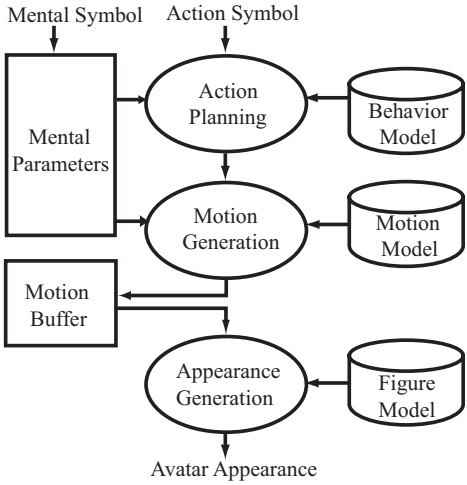

Fig. 2. Process flows

Avatar with Pre-defined Knowledge. We define that an avatar is an object which is participant's substitute in a virtual space. An avatar has pre-defined knowledge to generate its motion and appearance from symbols. But it is time-consuming job to construct or modify the knowledge. Therefore the pre-defined knowledge is to be described in a reusable and extensible form. The details of the knowledge are described in section 3.

Representation of Virtual Space. Generated appearance is represented in a virtual space. A participant is able to see the virtual space in which any participants, including him/herself, are represented as avatars.

3 Avatar Generation

As described in the previous section, RHP allows avatars to be designed beyond constraints of physical structure. To achieve this goal, and to make the avatar generation mechanism general, we have employed a layered structure of the pre-defined knowledge. We divide the pre-defined knowledge into three layers (see Figure 2), and we make them independent as much as possible in order that we can easily modify physical structure of an avatar.

The three layers are *behavior model*, *motion model* and *figure model*. An avatar plans the next action based on *behavior model*, generates a motion corresponding to the next action based on *motion model*, and generates the avatar's appearance with motion based on *figure model*. Here, motion means posture sequences of an avatar's body parts.

3.1 Behavior Model and Action Planning

Action planner generates avatar's next action (action plan) such as "walking" and "raising hand" based on received *symbols*, *behavior model* and *mental parameters*[1]. Action

[1] In this paper, we do not discuss the mental parameters because of page limitation

plans are highly independent of avatar's physical structure. This allows model constructors to modify or replace *behavior model* with taking little care of relations between *behavior model* and other models.

Action Planning. A human can perform multiple actions at the same time if these actions do not require the same body part. For example, "walking (an action using right and left leg)" and "raising hand (an action using right or left arm)" are not mutually exclusive. Therefore, the action planner should generate such multiple actions at the same time. Moreover, on RHP, symbols necessary for interaction depend on the kind of interaction. and it is desirable that *behavior model* can be modified easily, i.e., it should be as simple as possible. From the above considerations, the action planner plans an action referring to *behavior model*, which consists of two kinds of actions; (1)*outward action* is an action transiting from the neutral posture to a specified posture, (2)*Homeward action* is an action transiting from a specified posture to the neutral posture.

The neutral posture is the base posture of starting action. For instance of a human avatar, the posture is a standing posture with his/her arm taking down.

In general, the outward action can be planned in case that the posture of avatar's body parts when a symbol is received is the same as the neutral posture. On the other hand, in case that the posture of avatar's body parts when an action is planned is different from, or collides with, the neutral posture, the action can not be planned. However, if the collided posture is in the homeward action, then it can be planned, it is because avatar's posture is to be the neutral posture soon. A homeward action can be planned after the corresponding outward action was planned.

An action is mainly planned according to a received symbol. However, an avatar often freezes if the avatar acts only when symbols are transmitted, since no symbols are transmitted when a participant does not make any pre-defined actions. Needless to say, such avatar's behavior does not seem natural. To solve this problem, the action planner plans some actions spontaneously such as "folding arms" or "sticking hand into a pocket", which have no influence on interaction. These actions are planned according to the mental parameters. Therefore, during no symbols are transmitted, an avatar can represent actions according to the participant's mental state. To realize it, the system must understand the participant's mental state correctly, which is one of our important future works.

Importance of Action. Each action has a degree of importance for realizing such a function that important actions, or actions according to symbols can be planned more preferentially than others. An example is given below in case when an outward action with a higher degree of importance is selected when an outward action with a lower degree of importance is presented. At first, the homeward action with a lower degree of importance is planned. Then, the outward action with a higher degree of importance is planned immediately. In the opposite case, an action with a lower degree of importance is ignored. Fundamentally, an action according to a symbol is given the highest importance, because the symbol explicitly presents an intention of the participant. On the other hand, an action unrelated to an interaction is given lower importance.

3.2 Motion Model and Motion Generation

There is a motion generator in an avatar which generates the motion based on the planned action, *motion model* and mental parameters. *Motion model* stores detailed motion information corresponding to each planned action.

Motion Generation. *Motion model* is represented as a table of correspondence between an action generated by the action planner and motion information which consists of the following information.

1. Keyframe sequence:$Q_1, Q_2, \cdots, Q_N$
2. The number of frames in the motion:M
3. Frame numbers of keyframes:$p_1, p_2, \cdots, p_N$
4. Interpolation function : $f(i)|i = 0, 1, \cdots, M$

The motion generator generates a motion, or posture sequence, corresponding to a received action. Keyframes expressed with Quaternions are key postures in a motion. Quaternion Q is defined using a rotation axis (V_x, V_y, V_z) and a angle θ as equation (1),

$$Q = (V_x \sin\frac{\theta}{2}, V_y \sin\frac{\theta}{2}, V_z \sin\frac{\theta}{2}, \cos\frac{\theta}{2}). \tag{1}$$

Then, a motion is generated by interpolating between keyframes by using of interpolation function $f(i)$ where i is a frame number in a motion. $f(i)$ is represented in a Bezier function. The process of interpolation between Q_1 and Q_2 is described below. The difference between Q_1 and Q_2, called Q_{diff}, is calculated with equations (2), (3), (4) and (5),

$$Q = (x, y, z, w) \tag{2}$$
$$\bar{Q} = (-x, -y, -z, w) \tag{3}$$
$$Q_A Q_B = (v_A \times v_B + w_A v_B + w_B v_A, -v_A \cdot v_B + w_A w_B) \tag{4}$$
$$Q_{\text{diff}} = Q_2 \bar{Q}_1, \tag{5}$$

where $Q_A = (x_A, y_A, z_A, w_A) = (v_A, w_A)$ and $Q_B = (x_B, y_B, z_B, w_B) = (v_B, w_B)$. $(V_{x\,\text{diff}}, V_{y\,\text{diff}}, V_{z\,\text{diff}})$ and θ_{diff}, which are the rotation axis and the rotation angle between Q_1 and Q_2, can be calculated using equation(1). The motion $Q_{in}(i)$ $(p_1 \leq i \leq p_2)$ for moving from Q_1 to Q_2 is calculated with the equation(6), using $(V_{x\,\text{diff}}, V_{y\,\text{diff}}, V_{z\,\text{diff}})$, θ_{diff} and $f(i)$.

$$Q_{in}(i) = (V_{x\,\text{diff}} \sin\frac{\theta_{in}}{2}, V_{y\,\text{diff}} \sin\frac{\theta_{in}}{2}, V_{z\,\text{diff}} \sin\frac{\theta_{in}}{2}, \cos\frac{\theta_{in}}{2}) \tag{6}$$
$$\theta_{in} = \theta_{\text{diff}} f(i) \tag{7}$$

In this example, we describe about a motion using only one body part. In the case of using multiple body parts, it is just to do these operation for every body part. And the motion generator change the interpolation function by moving control points according to the mental parameters. Therefore a motion can be changed according to the mental states at that time.

Motion Buffer. The motion buffer is a kind of queue. Basically, newly generated motions by the motion generator are added to the tail of the motion buffer, and the oldest motion at the head of the motion buffer is removed to generate avatar's appearance by the appearance generator. However, in case that a generated motion does not collide with an already stored motion, the generated motion is not added to the tail but unified motion is stored where the old stored motion was. This is because that these two motions can be represented simultaneously. For example, when a motion "walking (using right and left leg)" is already stored and a motion "pointing with right finger (using right arm)" is generated, these motions do not collide with each other. So they can be represented simultaneously, then they are unified and turn into one motion "walking pointing with right finger(using right and left leg and right arm)," which is stored where the motion "walking" was.

Motion Fusion. As described before, we restrict actions planned by the action planner to only two kinds of actions which are outward actions and homeward actions. This can reduce animator's job, and make action planning simple. On the other hand, our aim is avatar-based interaction, so an action according to a symbol has to be represented immediately. Therefore such a system does not meet our aim that can not represent an outward action until a colliding homeward action has finished. Moreover, it is unnatural that all actions start from the neutral posture. To solve this problem, the motion generator fuses a homeward action and an outward action which collide with each other, in concrete, interpolates two motions.

The process of fusing two motions is described below. To make the explanation simple, both motions are single body part motions and the numbers of frames of both motions equal to M Motion of the homeward action is $Q_{h(i)}$ and motion of the outward action is $Q_{o(i)}$ $(0 \leq i \leq M-1)$. Difference between these motions, called $Q_{\mathrm{diff}(i)}$, is calculated with equation (5) using $Q_{h(i)}$ and $\bar{Q_{o(i)}}$. Using $Q_{\mathrm{diff}(i)}$, rotation axis $(V_{(i)_{x\,\mathrm{diff}}}, V_{(i)_{y\,\mathrm{diff}}}, V_{(i)_{z\,\mathrm{diff}}})$ and angle $\theta_{\mathrm{diff}(i)}$ for moving from $Q_{o(i)}$ to $Q_{h(i)}$ are calculated with equation(1), and the fused motion, called $Q_{c(i)}$, is calculated with equation(6) using an interpolation function $f(i)$. The interpolation function is important in order to fuse two motions smoothly. We have succeeded in obtaining results like Figure 3 using a linear function $f(i) = i/(M-1)$. In case of using multiple body parts, it is just to do these operations for every body part.

3.3 Figure Model and Appearance Generation

Figure model stores avatar's geometry data and physical structure. The appearance generator generates avatar's appearance using the posture from the head of the motion buffer and figure model.

4 Conclusion

In this paper, we propose a concept of real-time human proxy for avatar-based interaction systems, especially we describe the details of avatar generation. According to the new method, we can easily construct the pre-defined knowledge keeping avatar's behavior natural.

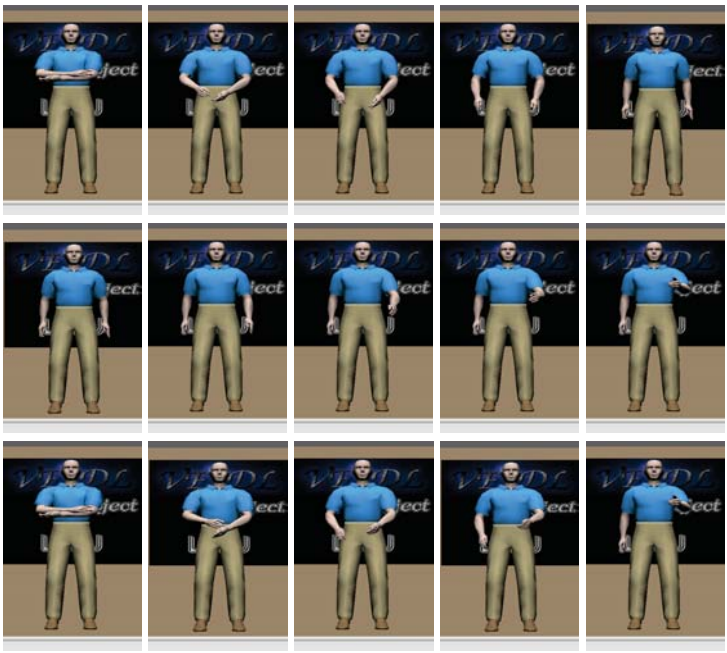

Fig. 3. Fusing Motion: the motion of a homeward action, that of an outward action and the fused motion are shown in the top row, the middle one and the bottom one respectively

Acknowledgment

This work has been partly supported by "Intelligent Media Technology for Supporting Natural Communication between People" project (13GS0003, Grant-in-Aid for Creative Scientific Research, the Japan Society for the Promotion of Science) and "Real-time Human Proxy for Avatar-based Distant Communication" (16700108, Grant-in-Aid for Young Scientists, the Japan Society for the Promotion of Science).

References

1. Date, N., Yoshimoto, H., Arita, D., Yonemoto, S., Taniguchi, R.: Performance evaluation of vision-based real-time motion capture. In: Proc. of Workshop on Parallel and Distributed Computing in Image Processing, Video Processing, and Multimedia, in IPDPS CD-Rom Proceedings. (2003)
2. Arita, D., Yoshimatsu, H., Hayama, D., Kunita, M., Taniguchi, R.: Real-time human proxy: An avatar-based interaction system. In: CD-ROM Proc. of International Conference on Multimedia and Expo. (2004)