Avatar generation for Real-time Human Proxy

Hayama, Daisuke
Department of Intelligent Systems, Kyushu University

Yoshimatsu, Hisato
Department of Intelligent Systems, Kyushu University

Yoshimoto, Hiromasa
Department of Intelligent Systems, Kyushu University

Arita, Daisaku
Department of Intelligent Systems, Kyushu University

他

出版情報：Proceedings of the 10th International Conference on Virtual Systems and Multimedia, pp.386-395, 2004-11
バージョン：
権利関係：

Avatar generation for Real-time Human Proxy

Daisuke Hayama, Hisato Yoshimatsu, Hiromasa Yoshimoto,
Daisaku Arita, Rin-ichiro Taniguchi
6-1, Kasuga-koen, Kasuga, Fukuoka, 816-8580, JAPAN
{*daizz,hisato,yosimoto,arita,rin*}*@limu.is.kyushu-u.ac.jp*

Abstract. This paper describes techniques to improve human representation 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 the 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 a side of representing human actions. Also we present experimental results.

1 Introduction

Modern communication technologies have made it usual to have communication for people who are distant from each other. For example, telephone is one of the most useful tools to communicate using acoustic information, and e-mail is one of them using literal information. Furthermore, information-richer tools are also available. Video-chat or video-phone are becoming available by data compression techniques and broader network bandwidth. Many people may consider video-phone to be the richest communication tool, since video-phone seems to provide us the feeling of being connected to the other speaker's place, because appearances of speakers are directly transmitted to each other. It has problems, however, in case that many (10 or more) people are participating at once, as on a conference or a meeting. Every participant in a conference must have numbers of windows showing the other participants on his/her display. Each window collocated on the same plane has its own geometrical coordinate system, showing his/her companion who is always just facing the front. This situation causes the user to have spatioperceptual inconsistency and causes the lack of reality. For instance, this does not make participants feel as if they were present alongside. To solve this problem and to establish the spatioperceptual consistency, we have chosen the way to get all participants into a virtual space.

There are several researches on virtual environments for human 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. Their positional relations are consistent virtually. This means that each participant can understand where other participants stand, look at, and point to, or understand where a sound comes from, and also move around in the virtual space. In contrast to a video chat system, participants can easily feel coexistence.

However, these avatar-based interaction systems have difficulty in controlling avatars, where the degrees of freedom of human body are so high that legacy input devices are not sufficient to acquire or input participants' activities.

To solve this problem, we have proposed Real-time Human Proxy; a concept to provide realistic virtual-space-based communication[1],[?]. In this paper, we describe the details of avatar generation for Real-time Human Proxy.

2 Real-time Human Proxy

2.1 Human Motion Sensing

In avatar-based interaction, an avatar is expected to reflect activities of a participant into a virtual space as if he/she were there. An avatar is expected to turn its head to a particular person when a participant does so, for example. Nevertheless, as already mentioned above, legacy input devices are not sufficient to acquire participants' activities in aspects of quality and quantity. Using such devices, participants have to keep feeding their own activities into a system by hand, and acquired information may not be precise. Special input devices are often used as solution to this problem[2]. We have developed a vision-based motion capture system (MCS)[3] as an input device, for acquiring more information of participants without compelling them annoying operations.

2.2 What is Real-time Human Proxy?

Although using an MCS as an input device, it is not possible to reflect all of participant's motions into an avatar. Although more information could be acquired in comparison with legacy devices, an MCS cannot acquire all information such as shape of hands, or movement of eyebrows and a lip. Just feeding captured data into an avatar, the lack of information may cause unnatural avatar motions. On the other hand, it is not necessary for an avatar to act exactly the same as participant's motions, since participants usually want to know only what others are doing, but how others moving. Real-time Human Proxy (RHP) is a new concept for avatar-based interaction, which makes better use of an MCS, and makes avatar act more meaningfully.

RHP is a concept which virtualizes a human in the real world in real-time. The aim is to make an avatar act as if he/she in a distant place is present in a virtual space. Therefore, we focus on acquisition and representation of human action or nonverbal information. We symbolize the human information on acquisition, and complement the symbol on representation, that they are illustrated in Figure 1 and the details are described below.

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 transmits the symbols to the representation side of a virtual space.

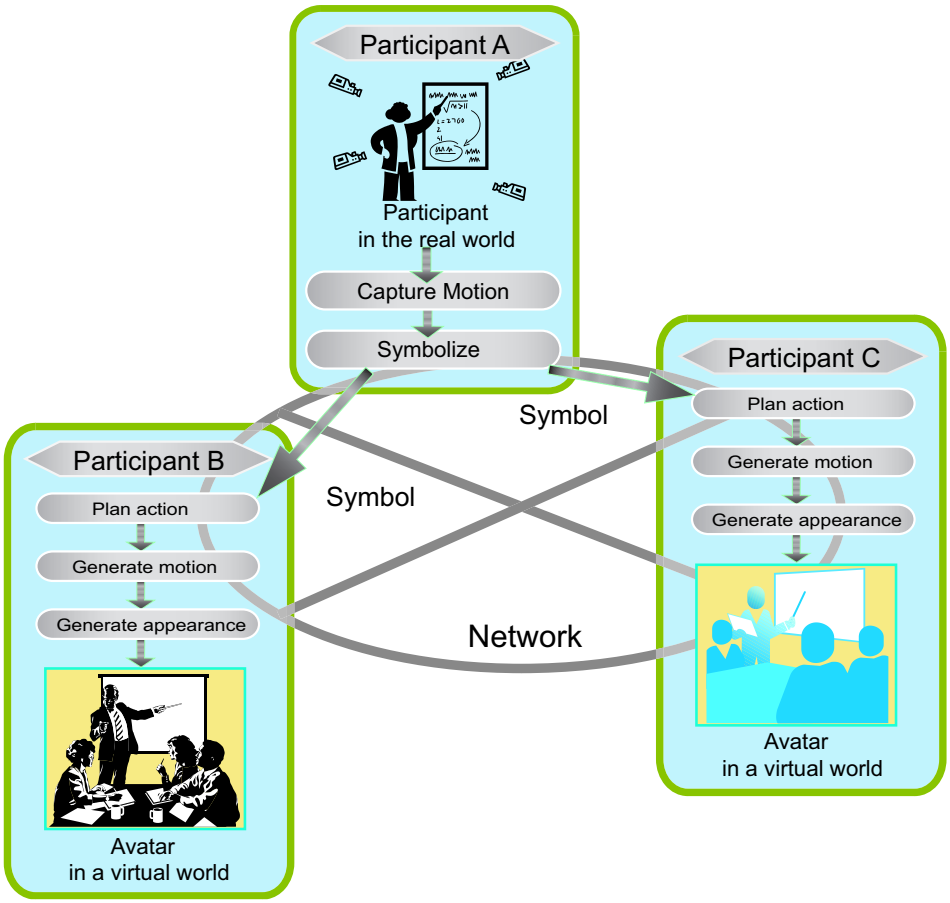


Figure 1: The concept of RHP.

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. Bruce and his colleagues built a virtual pet which is an autonomous and directable CG character[**?**]. They met the above demand by describing the pre-defined knowledge of virtual pet in a layered structure. Their aim is interaction between CG character in a virtual world and a human in the real world. But if a CG character is used as the avatar of an participant to interact with other participants, the motion according to a symbol must be represented immediately. Therefore it is difficult for their tequnique to use be applied to our system. Our pre-defined knowledge is also described in a layered structure, which consists of 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 avatar's appearance performing the motion based on Figure Model. In this paper, motion means posture sequences of avatar's body parts. Posture can be expressed with the Quaternions of avatar's body parts. Quaternion Q is defined as equation (1) using a rotation axis (V_x, V_y, V_z) and a angle θ.

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

The detiles of these models 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. The viewpoint of each participant is anchored where his/her avatar is represented.

2.3 Benefits

The use of an MCS liberates participants from annoying operations of controlling an avatar. On the symbolization process, the system recognizes the action that a participant makes. This operation is intuitive and natural since participants make the same action as they make in the real world.

The concept, i.e., symbolization of actions, and representation of an avatar which have pre-defined knowledge, makes representation process simple. As described above, acquired information about participant's motion does not have detailed information of several body parts such as angles of fingers, which must be compensated proving conformity with other body parts to provide natural representation. Using RHP, actions are already recognized, and it is possible to generate full body motion sequence in an arbitrary manner referring to the pre-defined knowledge.

Moreover, this allows avatars to be designed beyond constraints of physical structure. Ordinarily, each avatar is designed to fit an actor whose motions are captured, so that 3-dimensional positions of body parts acquired from an MCS is correctly represented in computer graphics. Some errors in interpretation of posing can occur from mismatches in the sizes of body parts, such as the height of body or the lengths of arms. RHP transmits the symbolized action, not the 3-dimensional positions directly, and consequently RHP largely relaxes the constraints. We can use any kind of avatar including higher body, shorter arms, bigger head, etc. This improves not only usability of the system, but also variety of avatars in interaction.

3 Avatar generation

3.1 Previous method

The system plans actions of an avatar using a set of state transition graphs[1]. Each graph corresponds to a body part such as right and left arm, right and left leg, and each state in the graphs corresponds to an action or a symbol such as "walking," "raising hand,"and "pointing with finger." When a symbol is transmitted, the current state of the graph is forced to transit to the state corresponding to the symbol. Advantage of this tequnicue is many actions of the full body are generated by the combination of graphs, each of which plans action of a body part. However, planning action with this tequnicue depends on the physical structure of an avatar since each graph is defined for the corresponding body part. Therefore, to change the physical structure of an avatar, we need to change all graphs. Moreover, if we want to plan such action that is carried out by two or more parts, the system has to synchronize multiple transitions on each graphs, which requires heavy work of model constructors.

3.2 New proposal of the pre-defined knowledge

As described above, RHP allows avatars to be designed beyond constraints of physical structure. To make the most of this advantage, we propose a new layered structure

of the pre-defined knowledge. Dividing the pre-defined knowledge into multiple layeres makes the dependence of each part of the knowledge on the other parts smaller. This means that it is easier for knowledge constructors to modify physical structure of an avatar.

In addition, an avatar has a set of mental parameters which represents avatar's mental state such as "happy" and "busy", and a motion buffer which is used for making avatar's motion natural. By using these information and mechanism, an avatar plans its actions, generates its motions and generates its appearance.

In this section, we describe the details of the pre-defined knowledge which consists of Behavior model, Motion model and Figure model.

3.3 Behavior Model and Action Planning

The action planner in an avatar outputs its next action such as "walking" and "raising hand" based on received symbols, Behavior model and mental parameters. Such outputs 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 the models.

3.3.1 Action Planning

A human can perform multiple actions in parallel, if the human's body parts used in them does not overlap, i.e. "walking(an action using right and left leg)" and "raising hand(an action using right or left arm)". So the action planner should allow it. Moreover, on RHP, Needed symbols are changed according to the kind of interaction. So it is desirable that the action planner can be modified easily. In consideration of the above conditions, the action planner plans an action using two kinds of following simple actions.

1. The action which makes an posture from the standard posture called *outward action*.

2. The action which makes the standard posture from an posture called *homeward action*.

The standard posture is the base posture of starting action. For instance of a human avatar, the posture is the standing posture with his/her arm taking down(see Fig.2). The outward action can be planned, if avatar's body parts which are used the action are same with the standard posture's at the time. On the other hand, if avatar's body parts are not same with the standard posuture's at the time (if it collides with other action), the action can't be planned, but if the collided action is the homeward action, then it can be planned, it is because avatar's posture is to be standard posture, after the homeward action is represented. The homeward action can be planned after the outward action was planned.

An action is mainly palnned according to a symbol. But 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." These actions are planned according to the mental parameters. So during no symbols are

transmitted, an avatar can represent actions according to the participant's mental state. But to allows it, we must understand the participant's mental state collectly, and it is the next subject. These actions may work for time filling, and may make participants feel more natural[1]. And of course, these actions do not indicate participant's intentions in order not to influence interaction between participants.

Figure 2: Standard Orientation

3.3.2 Importance

Each action is attached the index of an importance. An action with higher degree of importance is planned preferentially. The case when an outward action with higer degree of importance is wanted to plan at the time of represented an outward action with the lower degree of importance is described below. At first, the homeward action with the lower degree of importance is planned. Secondly, the outward action with higer degree of importance is planned immediately. On RHP, the degree of importance corresponds to the degree of considered that is important in an intaraction. Fundamentally, an action according to a symbol is attached the highest importance, on the other hand, an action which is unrelated to an interaction is attached lower importance.

3.4 Motion Model and Motion Generation

The motion generator in an avatar generates the motion based on the plenned action, Motion model and mental parameters. Motion model stores the detile motion information according to the planned action.

The generated motion(posture sequences) is stored sequentially from a past posture in the motion buffer. The motion buffer's architecture is like a queue. Fundamentally, the motion buffer's head posture(the oldest posture) is output, and the posture is used in appearance generator in an avatar. But the case when a motion was already stored and the generated motion doesn't collides with the motion is exception. In that case, two motions can be represented in parallel, therefore two motions are unified and turn into one motion, and the unified motion is stored in the same place as the already stored motion. 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 don't collide. So two motions can be represented in parallel, the motion "walking" and the motion "pointing with right finger" are unified and turn into one

motion "walking pointing with right finger(using right and left leg and right arm)." And the unified motion is stored in the same place as the motion "walking."

3.4.1 Motion Generation

The motion generator generates the motion from keyframe sequences. A keyframe is a main posture in a motion. Between two keyframes is interpolated according to a bezie function. The motion information is following (1) to (4).

1. Keyframe sequences
2. The frame length from start of the motion to end of it
3. The places of keyframes in the motion
4. Controll points for a bezie function

The equation which solve the bezie function using controll points is described Appendix. The detail of interpolating is described below. Two keyframes which are interpolated are as Q_1 and Q_2, and the place of Q_1 and Q_2 in motion are as $p1$ and $p2$, and the interpolating function of a body part is as $f(i)$ $(p_1 \leq i \leq p_2)$. The difference posture for changing to Q_2 from Q_1, called Q_{diff}, is solved by equations (2), (3) and (4).

$$\bar{Q} = (-x, -y, -z, w) \qquad (however \quad Q = (x, y, z, w)) \tag{2}$$

$$\begin{aligned} Q_A Q_B &= (v_A \times v_B + w_A v_B + w_B v_A, -v_A \cdot v_B + w_A w_B) \\ &\quad (however \quad Q_A = (x_A, y_A, z_A, w_A) = (v_A, w_A), \\ &\quad and \quad Q_B = (x_B, y_B, z_B, w_B) = (v_B, w_B)) \end{aligned} \tag{3}$$

$$Q_{\text{diff}} = Q_2 \bar{Q}_1 \tag{4}$$

The rotation axis $(V_{x\,\text{diff}}, V_{y\,\text{diff}}, V_{z\,\text{diff}})$ and the rotation angle θ_{diff} are solved by the equation (1). These are the rotation axis and the rotation angle for changing to Q_2 from Q_1. The motion $Q_{in}(i)$ $(p_1 \leq i \leq p_2)$ for changing to Q_2 from Q_1 is solved by the euqation (5), using $(V_{x\,\text{diff}}, V_{y\,\text{diff}}, V_{z\,\text{diff}})$,$\theta_{\text{diff}}$ and $f(i)$.

$$\begin{aligned} Q_{in}(i) &= (V_{x\,\text{diff}} \sin\frac{\theta_{in}}{2}, V_{y\,\text{diff}} \sin\frac{\theta_{in}}{2}, V_{z\,\text{diff}} \sin\frac{\theta_{in}}{2}, \cos\frac{\theta_{in}}{2}) \\ &\quad \theta_{in} = \theta_{\text{diff}} f(i) \end{aligned} \tag{5}$$

In this example, we describe the case as 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 interpolating function moving controll points according to the mental parameters. Therefore a motion changes according to the mental states at that time.

3.4.2 Combine the Motions

As described above, we restrict the using actions to two kinds of actions which are outward action and homeward action. It allows animater to reduce the job, and allows action planning to simplify. However our aim is the avatar-based interaction, so

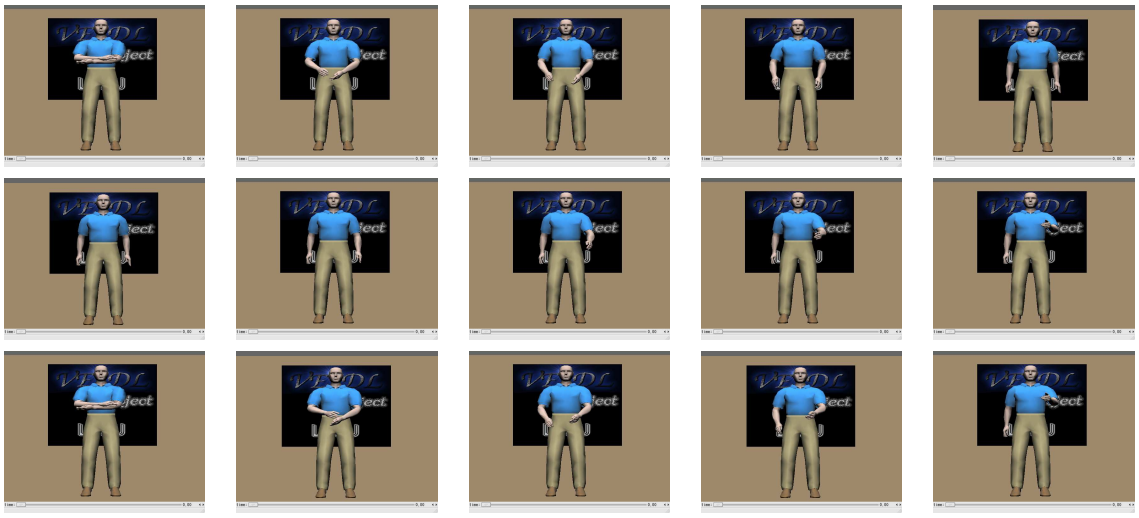


Figure 3: Combined Motion

an action according to a symbol has to be represented immediately. Therefore a system which can't represent an outward action when a collision homeward action is not completed is to be fatal fault in our aim. Moreover, it is unnatural, if all action is represented from standard posture. To solve this problem, we combine the homeward and outward action which collide each other. In concrete, we interpolate two motions.

The detile is described below. To say easily, each motion's frame length is same as "M", and let each motion use only one body part. Motion of homeward action is as $Q_{h(i)}$ and motion of outward action is as $Q_{o(i)}$ $(0 \leq i \leq M-1)$. Difference of each motion, calld $Q_{\mathrm{diff}(i)}$, is solved by equation (4) 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 changing to $Q_{h(i)}$ from $Q_{o(i)}$ are solved by the equation (1), and a combined motion, called $Q_{c(i)}$, is solved by the equation (5) using an interpolating function $f(i)$. The interpolating function is important in order to combine two motions smoothly. We have succeeded in obtaining a result like Fig.3 using a linear function $f(i) = i/(M-1)$. The Fig. 3 of the upper row show the homeward action's motion, the middle row show the outward action's motion, and the lower row show the combined motion. Like the case of the motion generation, in the case of using multiple body parts, it is just to do these operation for every body part.

3.5 Figure Model and Appearance generation

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

4 Prototype System of RHP

In before experiment which was a simple interactive game, we verified the effectiveness of RHP in case that all actions necessary for interaction was able to be listed to recognize and display[?]. This time, to verify the effectiveness of proposal tequniques, we do experiment to verify wether participants can impress the similar results, in spite of being able to cut down the works for making pre-defined knowledge of avatar.

In order to compare the conventional technique with proposal tequnique, we do the same experiment as before experiment. The rule of the game, which is a simplified version of a famous game in Japan, ara as follows:

1. One of participants becomes a leader.
2. The leader says "A" and points to one of participants.
3. The participant who is pointed to at step 2 says "B" and points to one of participants.
4. The participant who is pointed to at step 3 says "C" and puts his/her hands up.
5. The participant who puts his/her hands up and becomes a leader.
6. Return to step 2 until someone fails.

MCS which captures motion information about each participant and other equipments are same as what are used by before. The labels of actions which are parts of pre-defined knowledge are as follows:

- Outward of *finger pointing*, and homeward of it: actions of according to symbol.
- Outward of *hands up*, and homeward of it: actions of according to symbol.
- Outward of *head turn*, and homeward of it: actions of according to symbol.
- Outward of *arms folding*, and homeward of it.
- Outward of *arm bend*, and homeward of it.

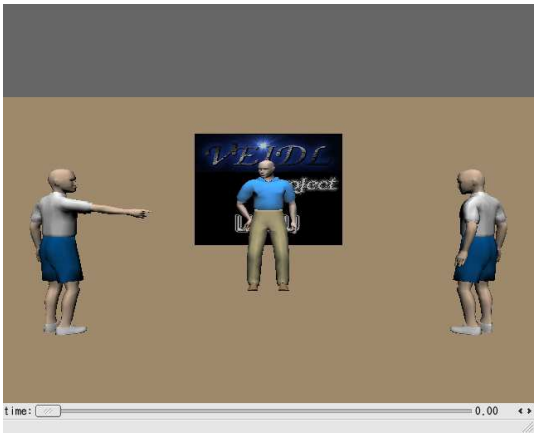
Some snapshots of the game are shown in Fig.4. The impressios of participants about representation as follows:

- Avatar represented the pre-defined action which participants behaved. And presented the action which did not indicate participant's intentions, when participants did not make any pre-defined actions. Therefore avatar's behavior seemed natural.
- If participant behaved the *finger pointing* when avatar presented *arms folding*, avatar pointed at its finger, loosening its arms.
- Similar with conventional tequnique, participants were able to easily understand where avatars looked and pointed.

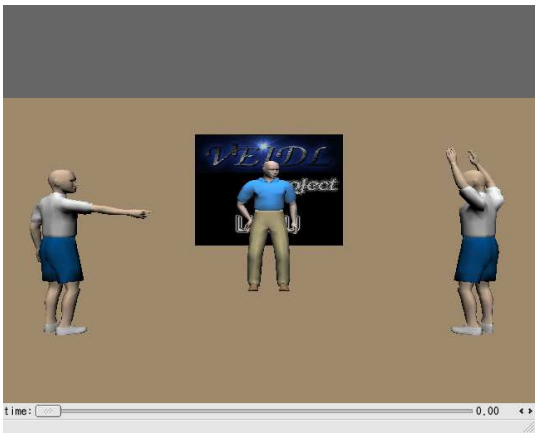
5 Conclusion

In this paper, we propose a concept of real-time human proxy for avatar-based interaction systems, especially we described the details of avatar generation. To verify the effectiveness of proposal tequniques, we compare them with conventional one.

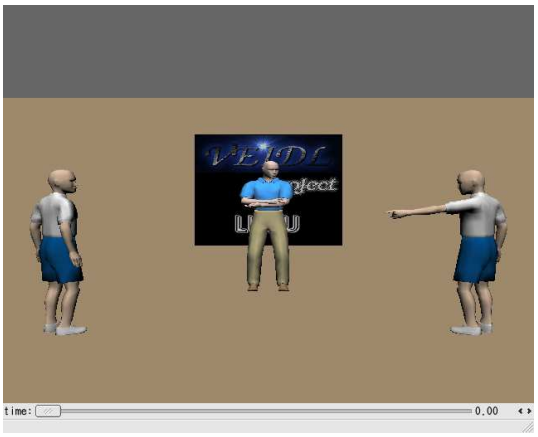
(a) A leader points to a participant.



(b) The participant points to a participant.



(c) The participant puts his/her hands up.



(d) The participant becomes a leader and the next turn starts.

Figure 4: Snapshots of the game

Appendix: Bezie Curve

Bezie curve is N-1-D curve generated by n of controll points. 3-D curve $P(t)$, we use, is solved by 4 of controll points (P_0,P_1,P_2,P_3), as follows.

$$P(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3 \qquad t : 0to1 \tag{6}$$

$P(t)$ draws the curve as Fig.5.

References

[1] Arita, D., Taniguchi, R.: Non-verbal human communication using avatars in a virtual space. In: Proc. of International Conference on Knowledge-Based Intelligent Information and Engineering Systems. (2003) 1077–1084

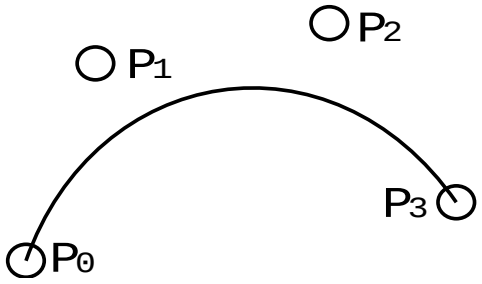


Figure 5: Bezie Curve

[2] Roussos, M., Johnson, A.E., Leigh, J., Vasilakis, C.A., Barnes, C.R., Moher, T.G.: Nice: combining constructionism, narrative and collaboration in a virtual learning environment. ACM SIGGRAPH Computer Graphics **31** (1997) 62–63

[3] 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)