POSTER: Controlling 3D characters by hand motion

Misae Saitoh
Hosei University
Koganei Kajinocho 7-3-2
184-8584, Tokyo, Japan
saitohmisae@philos.k.hosei.ac.jp

Kenji Miyamoto
Hosei University
Koganei Kajinocho 7-3-2
184-8584, Tokyo, Japan
miyaken@hosei.ac.jp

ABSTRACT

A method to control motions of complex 3D characters interactively using hand movements is described. After capturing actor's motion, by motion capture systems transfer it as it is to 3D character's motion. Motion capture fails, however, when the target character is of shape far different from that of the actor captured. This article proposes manipulating 3D character by controllable parameters extracted from monocular image of the actor's hand without reconstructing its posture. Using Digital Puppet System which implements this method allows a user to easily manipulate characters of various shapes ranging from a human-type robot to a mollusk with large-scale set of joints. By extracting as many controllable parameters as possible, our method makes maximal use of available information which is not necessarily enough for posture reconstruction.

Keywords

3D character, hand motion, interactive animation, motion capture, model-based

1. INTRODUCTION

A method to control complex motions of 3D characters using hand movements is described.

For controlling 3D characters, motion capture first reconstructs the actor's posture from positional information of a set of markers and then, the postural parameters are used as they are to determine the character's posture. Recent researches [Cha05a] [Nor93a] [Sem98a] [Yin03a] aim to reduce the number of required markers. In general, motion capture has advantage in reproducing realistic motion.
However, the number of independent parameters that can be controlled than the accuracy of the motion capture is more important to move the character with connecting topology different from that of the human body. For example, to reconstruct hand posture, it needs a lot of markers because the hand has 19 joints, notwithstanding that actually controllable information is much less.

Digital Puppet System (DPS) doesn't reconstruct posture but extracts directly the feature parameter, a quantity related to posture, that can be controlled by the actor. More specifically, 1) DPS extracts feature parameters from the location information of the markers by model-based method [Att89a] [Kak00a] [Ued03a], and 2) uses these parameters as control parameters that determine the posture of the target character. In the actual system, watching user's hand with a marker attached to the tip of each finger, DPS estimates how much the fingers bend or stretch. Then, these feature parameters are assigned to the joints of the parts of the character, determining their bending angles. Here, a single parameter can possibly be used for more than one parts of 3D character.

Experiment showed that by using this system a user can control the movement of 3D character by his hand with rotating, moving and, bending or stretching its three fingers, Wriggle motions of a 3D character with multiple joints were also able to be controlled. Thus, whole body of 3D character was able to be controlled as if working on puppets.

Our method shows how to maximally make use of the available information, even if there is not enough information for reconstruction.

2. RELATED WORK

In methods other than motion capture, user moves sensors that can detect its own movement. [Las00a] controls the movement of the character by the direction and extent the mouse is moved. Video game machine Nintendo's Wii [Nin] controls the character by moving a remote control equipped with motion sensor. In these methods, a few parameters corresponding to the number of freedom could be controlled independently but they are difficult to change concurrently.

3. EXTRACTING FEATURE PARAMETERS

Digital Puppet System directly extracts feature parameters without reconstructing actor's posture. Here, from the usability's point of view, it is preferable that the feature parameters can be controlled independently. From expressive power's point of view, it is preferable that as many parameters as possible are available given fixed number of markers. This chapter describes *model-based feature extraction* (*MBFX*) a model-based method to extract independent parameters from small number of markers attached to the tip of a finger tracked. Then, conversion of posture information on 3D character from feature parameter are also briefly described

3.1 Feature Extraction for Hand

3.1.1 2-D of hand model

The *feature model*, which relates feature parameters to quantities to be measured, is used to estimate *feature vector*, each element of which is a feature parameter, from the measured quantities. The proposed feature model, called *2-D hand model* (Fig. 1), is a model of a hand that bends and stretches three fingers, translates and rotates in the plane vertical to the optical axis of the camera. The model consists of a point P on the plane, and three lines which pass P and preserves their relative crossing angles, and points Q_1, Q_2 and Q_3 on these lines. The lines mean the directions of the fingers and Q_i means the tips of them. Based on this model, *feature vector* in 2-D hand model is represented by triple (p, γ, a). Here, $p = (p_x, p_y)$ is the two dimensional position of P, γ is the rotation angle from some initial value, and a is the vector whose elements are $|q_i|$ (i=1,2,3).

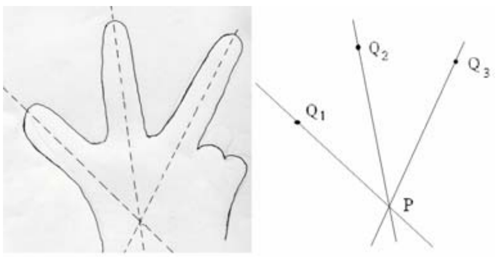

Figure 1. 2-D hand model

The next section describes the method to estimate, given an initial position of point P, the value of parameter (p, γ, a) changing at each time from observed position of points Q_1, Q_2, and Q_3.

3.1.2 Parameter Estimation

Initial value

Initial position P_0 of point P is assumed to be given. In the actual system, 2-D model is displayed on the screen. When three fingers fit to the lines, P_0 is obtained by regarding the displayed figure as the initial state of the model. Thus, the initial value is $(p, \gamma, a) = (0, 0, |P_o Q_i|)$.

Estimation of feature vector

Change in (p, γ, a) according to change in markers' positions are estimated. The i-th marker's new position q_i' can be represented by their old position q as:

$$q_i' = a_i R_\theta u_i + \ell \quad (i=1,2,3) \quad \cdots (1)$$

Here,

$$R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \quad u_i = \frac{q_i}{|q_i|},$$

ℓ : displacement vector $p' - p$

Here, the unknowns in *eq*.(1) are a_i, R_θ and ℓ. Because the position changes continuously,

$$R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \cong \begin{pmatrix} 1 & -\theta \\ \theta & 1 \end{pmatrix}$$

assuming $\theta \ll 1$. Under this approximation, θ can be solved as:

$$\theta = b/c$$

$$b = u_{1y}u_{3y}u_{2x}q_{3x} - u_{1y}u_{2y}q_{2x}u_{3x}$$
$$- u_{1y}u_{3y}u_{2x}q_{1x} + u_{1y}q_{2y}u_{2x}u_{3x} + u_{1y}u_{2y}q_{1x}u_{3x} - u_{1y}q_{3y}u_{2x}u_{3x}$$
$$- u_{2y}u_{1x}u_{3y}q_{3x} + u_{3y}u_{2x}u_{1x}q_{1y} + u_{2y}q_{2x}u_{1x}u_{3y} + q_{3y}u_{2y}u_{1x}u_{3x}$$
$$- q_{1y}u_{2y}u_{1x}u_{3x} - q_{2y}u_{2x}u_{1x}u_{3y}$$

$$c = -u_{1y}u_{3y}u_{2x}q_{3y} - u_{1y}u_{3x}u_{2x}q_{3x} + u_{1y}u_{2x}q_{2x}u_{3x} + u_{1y}u_{3y}u_{2x}q_{1y}$$
$$- q_{1y}u_{1y}u_{3x}u_{2y} + u_{1y}q_{2y}u_{2y}u_{3x}$$
$$- u_{1x}q_{1x}u_{3x}u_{2y} + u_{3x}u_{2y}u_{1x}q_{3x} + q_{3y}u_{2y}u_{1x}u_{3y} + u_{3y}u_{2x}q_{1x}u_{1x}$$
$$- u_{1x}u_{3y}u_{2x}q_{2x} - u_{1x}u_{3y}q_{2y}u_{2y}$$

a_i, ℓ in *eq*. (1) is solved using this θ.

$$l_x = d/e, \; l_y = f/e$$

$$d = s_{2x}s_{1x}q_{1y} - s_{2x}s_{1y}q_{1x} - s_{2x}s_{1x}q_{2y} + s_{1x}s_{2y}q_{2x}$$

$$e = -s_{2x}s_{1y} + s_{1x}s_{2y}$$

$$f = s_{2y}s_{1x}q_{1y} - s_{2y}s_{1y}q_{1x} + s_{2y}q_{2x}s_{1y} - q_{2y}s_{2x}s_{1y}$$

$$a_1 = g/eq_1$$

$$a_2 = h/eq_2$$

$$a_3 = k/eq_3$$

$$g = s_{2y}q_{1x} - s_{2y}q_{2x} + s_{2x}q_{2y} - s_{2x}q_{1y}$$

$$h = -s_{1x}q_1 + s_{1y}q_{1x} + s_{1x}q_{2y} - q_{2x}s_{1y}$$

 ISBN 978-80-86943-99-2

$$k=-s_{2x}s_{1x}q_{1y}-s_{2x}s_{1y}q_{3x}+s_{2x}s_{1y}q_{1x}+s_{2x}s_{1x}q_{2y}+s_{1x}s_{2y}q_{3x}-s_{1x}s_{2y}q_{2x}$$

Here, $R_\theta u_i = s_i$ is assumed.

After all, putting the above all together, we obtain the tracking procedure.

Tracking

1. $P := P_0$, $\gamma := 0$, $u_i := P_0 Q_i$;

2. Each time when the position of the markers change, repeat the following.

Obtain θ, ℓ, and a by calculation described above. Then

$p := p + \ell$;

$\gamma := \gamma + \theta$;

$a_i := |PQ_i|$;

3.2 Determining Target Posture

The remaining task is to transfer the parameter extracted above into posture information of 3D character in each frame. This is done simply by setting the value of bending parameter, latitude or longitude, of each joint of target to the value of some extracted parameter. Here, the structure of target 3D character is supposed as follows.

Posture information of 3D character

3D character consists of two or more parts, which are connected by joints. The parts form a tree structure with some part as its root and joints as its edges. Thus, parent-child relation naturally be introduced among parts. Using this structure, when the parent part moves, child part moves together in the coordinate of the parent part. The direction of flexing of a joint is represented by the latitude and the longitude and, thus, the entire posture of a 3D character is determined by flexing of all joints.

4. DIGITAL PUPPET SYSTEM

This chapter describes the overview of Digital Puppet System.

Fig. 2 shows the arrangement of the Digital Puppet System. The screen displays 3D character image together with the image of the user's hand. Actor controls 3D character while watching 3D character.

Figure 2. Digital Puppet System

Fig. 3 shows the screen part displaying the actor's hand with red, green and blue markers attached. Red, green, and blue markers are attached to the tip of the thumb, the forefinger, and the middle finger.

Figure 3. Hand with markers

Example 1

This example demonstrates manipulating 3D robot of the human type. Fig. 4 shows a screenshot during the operation. Acting person's thumb, forefinger, and middle finger correspond to the right arm, the left arm, and both feet of the 3D robot respectively. When the forefinger is bent, the right arm of 3D character is lowered, and is heighten when stretched. If the hand is rotated, the head of 3D robot slants, and if the hand is slid, the whole body of 3D robot moves in the same direction.

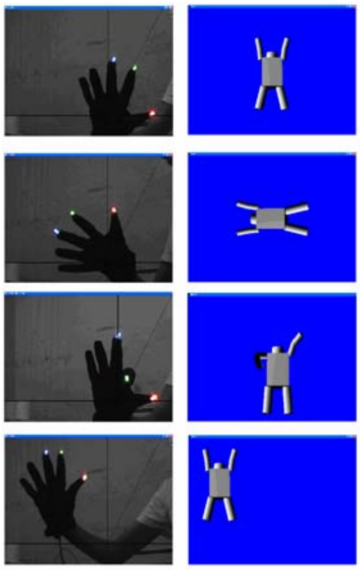

Figure 4. Controlling 3D robot

ISBN 978-80-86943-99-2

Example 2

The second example demonstrates manipulating 3D object that has foots of a multi joint. Each foot consists of many parts linearly chained. Fig. 5 shows a screen shot during the manipulation. Actor's thumb, forefinger, and middle finger correspond to the three feet of 3D object respectively. When the middle finger is bended, the corresponding foot winds smoothly according to the parameter of each joint.

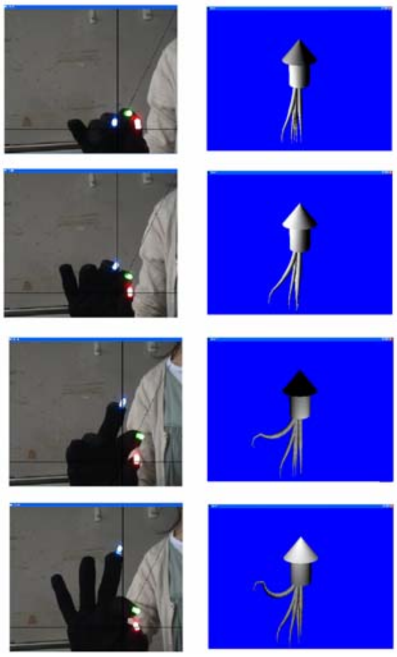

Figure 5.Controlling 3D squid

5. DISCUSSION

A method to control a 3D character by hand is described. In this method, first feature parameters are extracted from motion capture with small number of markers and then, these parameters are transferred as feature parameters to the target character. The experimental use showed that the movement of the whole body of a 3D character can easily be controlled by hand movement as if playing a puppet. Moreover, by spreading a single feature parameter to multiple joints, a character with large-scale set of joints can be controlled. Thus, variation of controllable 3D characters was extended.

Recent researches in motion capture are making efforts to reduce the number of required markers using structural dependency among posture parameters to be captured. Our work, to the contrary, aims to extract as many 'independent' parameters as possible, independent in the sense that each can be independently controlled by the user. In this sense, both motion capture with reduced markers and DPS aim to make maximal use of available information. Note here that the size of parameters extracted by DPS presented in this article is maximum because the degree of freedom of the set of marker is equal to the number of parameters extracted.

These two methods have, however, different intended applications. DPS is suitable for manipulating a character different far from the captured shape, whereas motion capture is for reproducing realistic motion of a character of similar shape.

DPS controls only the movement of 3D character. However, it can also be applied to controlling the speed of the operation or the shape change such as expansion and contraction. Generally, it is applicable to any parametrizable quantity.

Future work

The control of the surface shape by the movement of the hand will be the next step of this research.

ACKNOWLEDGMENTS

The authors thank to Akinori YOSHINO (graduate of Hosei Univ.) for a part of the program used in this system.

REFERENCES

[Att89a] Attwood,C.I., G.D.Sullivan,and K.Baker, Model based recognition of human posture using single synthetic images. *Proc.* the fifth Alvey Vision Conference, pp25-30,1989.

[Cha05a]Chai,J. , and Jessica K. Hodgins, Performance Animation from Low-dimensional Control Signals. *Proc.* ACM SIGGRAPH, No. 3, pp.686-696,2005.

[Kak00a] Kakadiaris,L and D.,Metaxas. Model-based estimation of 3D human motion. IEEE Transactions on Pattern analysis and machine intelligence. Vol.22, No.12,2000.

[Las00a]Laszlo.J, M., van de Panne, and E., Fiume. Interactive Control for Physically-Based Animation. *Proc.* ACM SIGGRAPH, pp. 201-209, 2000

[Nor93a]Norman I. Badler, Michael J. Hollick, John P. Granieri: Real-Time Control of a Virtual Human Using Minimal Sensors. Presence 2(1), pp.82-86, 1993

[Sem98a]Semwal, S., Hightower, R., and Stansfield, S. 1998. Mapping algorithms for real-time control of an avatar using eight sensors. Presence 7, 1, 1-21.

[Ued03a] Ueda,E, Y.Matsumoto, M.Imai, T.Ogasawara. Hand pose estimation for vision-based human interface. IEEE Transactions on Industrial Electronics,Vol.50, No.4,pp676-684,August,2003.

[Yin03a]Yin, K., Pai, D. K, An interactive animation system. *In Proc.* ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp.329-338, 2003.

[Nin]Nintendo,Wii Website http://wiiportal.nintendo-europe.com/13.html