Front. Electr. Electron. Eng. China 2008, 3(1): 49–54
DOI 10.1007/s11460-008-0009-3

RESEARCH ARTICLE

Wenhu QIN, Yuhui WU, Zhengxu ZHAO

Research on virtual actor action editing and movement control

Abstract To directly use a virtual surface model for action editing and movement control, a general method for creating virtual actor skeleton models and controlling movement is presented. The method includes judging borderlines of the block virtual surface model, calculating the joints, confirming the above block, and using the block hierarchical layout to create the skeleton model. Then, according to the virtual actor model and movement restriction, the study focuses on the generation of movement animation using the key frame technique and smoothing movement technique by automatically adding animation and adjusting the actor's pose by different weights on movement amplitude. Finally, movement control of the actor in the virtual environment is implemented by real-time control and path point control, which achieve a good result.

Keywords virtual actor, skeleton model, move model, move control

1 Introduction

A virtual actor is defined as an autonomic virtual human, virtual animal or virtual insect, etc., in a virtual environment. It has extensive applications in manufacturing, entertainment, military utilization, space flight, and public security. The research on virtual actors, including topics such as skeleton modeling, surface distortion, facial expression, clothing alteration, motion capture, and motion editing [1–3], is attracting increasing attention. In these fields, the animation is based on skeleton modeling and the motion editor, which plays an important role in certain applications. For instance, users concerned about the conduct and gesture of a virtual diving athlete in a virtual competitive diving simulation would pay more attention to the hand operation act of the virtual engineer in repairing the simulation [4]. Moreover, in a virtual criminal scene application, the motion process and the act of the criminal or victim are emphasized.

There are several groups working on motion editing and are concerned about control techniques of virtual action worldwide. Badler, who is from the Human Body Modeling and Simulation Lab of the University of Pennsylvania in the United States, studies the control and capture of virtual human movement and the application of a parameterized key frame method in a joint-based virtual human modeling system. Thalmann from the virtual reality (VR) Lab in Lausanne, Switzerland, has carried out many researches on human body modeling and distortion, as well as motion control systems for walking and grasping [5,6]. Wang Zhaoqi from the Chinese Academy of Sciences employs automatic skeleton match and visual skeleton editor to accomplish skeleton modeling. This method has been brought into use in sign language and performs well [7]. Zhuang Yueting in Zhejiang University obtains two dimensional motion skeleton series using video and motion picture series, which are taken as the basis for animation [8]. Hong Bingrong at the Harbin Institute of Technology accomplished a simulation of walking and running of virtual humans [9].

Most of the researches on modeling mentioned above adopt kinematics and dynamics methods. Before controlled motion is realized by kinetic equations, skeleton modeling is required. The resolution may be very difficult due to too many kinetic equations, which makes controlled motion hard to perform in real time. The paper presents a general skeleton modeling method for a virtual actor, i.e., the skeleton modeling of the actor is established automatically through the identification of the surface model. Process animation is generated by employing the parameter key frame technique, and then real-time control and path point control of the character in a virtual environment are realized.

Translated from *Journal of Southeast University* (*Natural Science Edition*), 2006, 36(5): 720–724 [译自: 东南大学学报 (自然科学版)]

Wenhu QIN (✉), Yuhui WU, Zhengxu ZHAO
Department of Instrument Science and Engineering, Southeast University, Nanjing 210096, China
E-mail: qinwenhu@seu.edu.cn

2 Modeling of virtual actor

The surface model (3Ds file format) of a virtual actor can be exported from professional software such as 3Ds Max, MAYA and Poser. These character models are designed in several chunks according to their characteristics with a hierarchical structure [10], as shown in Fig. 1. Motion control cannot be realized directly using the surface model, thus the skeleton model of a virtual actor is required.

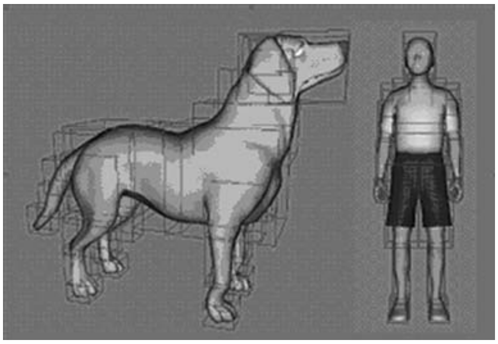

Fig. 1 Chunks of virtual actor

2.1 The boundaries and central points of chunks

The file of the character surface model contains the geometry shape information of the facial array $F\{F_1, F_2,\ldots,F_n\}$ and vertical array $V\{V_1,V_2,\ldots,V_n\}$. Each unit of the vertical array is composed of coordinate values (x, y, z) of the vertex. Each unit of the facial array $F_i = \{n_1, n_2, n_3\}$, in which n_1, n_2, n_3 represents vertical numbers in the array respectively, denotes that every face is made up of a triangle. The boundary of the chunk is only used by one triangle, so it can be obtained by improved intersection arithmetic which covers the whole face. Four steps are involved in the arithmetic as follows:

1) Define a line array E, of which each unit has three variables: A, B, N. A and B represent the two terminals of a line, whereas N stands for the times that the line is used.

2) Draw out a line from the face and check the line array E. If the line is excluded from the array, add it. Otherwise, replace N with $N+1$for this line.

3) Draw out the other 2 lines of the face and mark them as step two.

4) Check the other lines of all the faces and mark them.

The boundary of the chunk, i.e., the lines whose value $N=1$, can be drawn out using the method above. The central point of the boundary is calculated according to the average value of points which belong to the boundary.

2.2 Skeleton modeling

Taking the central point of the chunk as the joint, the skeleton is then acquired by connecting all the joints. If there are two joints of a chunk, the connection of these joints composes the skeleton. Otherwise, if the number is 1 or n, the ensuing method is adopted:

1) If the number of the joint is 1, just connect the joint with the central point of the chunk to form the skeleton.

2) If the number is n ($n>2$), then the chunk has n bones, which are composed by linking the central point with every joint respectively.

The chunk is of a hierarchical structure. Thus, the top chunk whose motion includes the translation and rotation of the axis of x, y, z should be defined, while the other chunks can only rotate around the axis of x, y, z. The motion of a chunk affects its own sub chunks. Conversely, a sub chunk has no influence on its parental chunk. The rotation and translation of the top chunk result in the motion of the model. Generally, the top chunk is the central part of the model, like the chest that is regarded as the top chunk of a virtual human or dog.

The skeleton model is generated after acquiring all bones. The top bone is taken as the parent vertex, and then the bones of its sub chunks are searched. Each sub bone is taken as the parent bone, and the steps above are repeated until the number of chunks turns into one. The rule for judging sub chunks is that the distance between two central points of the boundaries is the shortest, as boundaries of the parent and the child chunks are almost coincident with each other. The skeleton model of a virtual human is shown below in Fig. 2.

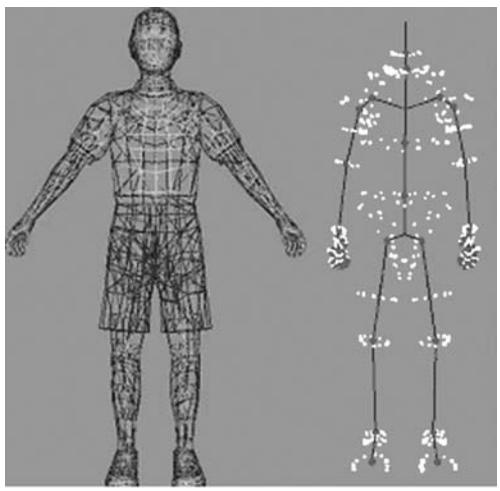

Fig. 2 Skeleton model of a virtual human

3 Virtual actor motion model analysis

For the human body, the three most familiar motions of man are walking, running and jumping. An analysis of man's walking is presented as follows.

3.1 Walking model

The walking movement can be divided into 3 steps: start step (A: from stillness state to stride one step), walking step (B: walking in turn) and stop step (C: from walking state to stillness state). Therefore, the three steps are enough to finish the walking action, as shown in Fig. 3.

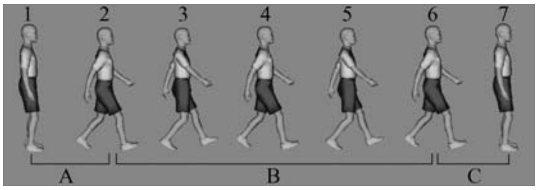

Fig. 3 Walking model of a human

3.2 Restriction relation of movement

There is a limitation for each joint during the movement of the joint-based skeleton model of the virtual actor, i.e.,

$$R_{(x,y,z)\min} \leqslant R_{(x,y,z)\alpha} \leqslant R_{(x,y,z)\max}, \tag{1}$$

where $R_{(x,y,z)\min}$ and $R_{(x,y,z)\max}$ are the minimum and maximum angle allowed for a joint rolling with x, y and z axis.

The data are based on GB10000-88<Chinese adult body size>. The joint range of an adult human is shown as Table 1 [4,7].

When editing the movement of the skeleton model of a virtual actor, the movement of the joint cannot go beyond the range. Otherwise, the motion will be unnatural.

4 Key technology of action editor and movement control

4.1 Action editor based on key frame

Key frame technology is a usual method of controlling a virtual actor's movement. The fluent actions of a virtual actor can be obtained by interpolating the parameters of the key frames. At present, the simplest method is linear interpolation, while higher order interpolation is also adopted for smoother animation. Considering system cost and the requirement of real-time rendering, this paper utilizes linear interpolation. Suppose that the translation and rotation parameters of the ath key frame are $T_j(a)$ and $R_j(a)$, where the value of j can be x, y and z; the translation and rotation parameters of the bth key frame are $T_j(b)$ and $R_j(b)$. We use linear interpolation between key frame a and b, then the translation and rotation parameters of the ith ($a < i < b$) frame can be calculated by the following formulas

$$\begin{gathered}
\begin{pmatrix} T_x(i) \\ T_y(i) \\ T_z(i) \end{pmatrix} = \begin{pmatrix} \frac{T_x(b)-T_x(a)}{b-a} & 0 & 0 \\ 0 & \frac{T_y(b)-T_y(a)}{b-a} & 0 \\ 0 & 0 & \frac{T_z(b)-T_z(a)}{b-a} \end{pmatrix} \times \begin{pmatrix} i-a & 0 & 0 \\ 0 & i-a & 0 \\ 0 & 0 & i-a \end{pmatrix} + \begin{pmatrix} T_x(a) \\ T_y(a) \\ T_z(a) \end{pmatrix} \\
\begin{pmatrix} R_x(i) \\ R_y(i) \\ R_z(i) \end{pmatrix} = \begin{pmatrix} \frac{R_x(b)-R_x(a)}{b-a} & 0 & 0 \\ 0 & \frac{R_y(b)-R_y(a)}{b-a} & 0 \\ 0 & 0 & \frac{R_z(b)-R_z(a)}{b-a} \end{pmatrix} \times \begin{pmatrix} i-a & 0 & 0 \\ 0 & i-a & 0 \\ 0 & 0 & i-a \end{pmatrix} + \begin{pmatrix} R_x(a) \\ R_y(a) \\ R_z(a) \end{pmatrix}.
\end{gathered} \tag{2}$$

4.2 Manner of motion control

4.2.1 Real-time control manner

Real-time control refers to control of an actor's movement directly in a virtual scene through the keyboard, mouse or other controlling inputs. Its main considerations are real-time performance, actor's position translation and actor's angle change. The key points are described as follows.

1) Real-time performance mainly relates to the complexity of the actor model and the performance of

Table 1 Joint range of an adult human

joint name	front bend and back bend /°	left askew and right askew /°	left roll and right roll /°
neck joint	−35 — 40	−55 — 55	−55 — 55
waist joint	−50 — 100	−50 — 55	−50 — 50
coax joint	−15 — 120	−15 — 30	−70 — 110
knee joint	0 — 135		
anklebone	55 — 110		−70 — 110
shoulder joint	−30 — 180	−45 — 180	−40 — 140
elbow joint		0 — 145	
wrist	−60 — 75		−20 — 30

the machine. Generally, to achieve real-time, precision is compromised. To achieve multi-actor real-time movement under certain hardware conditions, the simpler the actor model, the quicker the rendering.

2) Actor's position translation is controlled by the move key; and the offset between frames should be taken into account. When using key frame animation to realize man's walking, if the model doesn't move correspondingly, the model will recur to the beginning point of walking after play walking animation and walk again. Meanwhile, when the virtual human is walking, if the position doesn't match the direction, the human's direction will not match the movement direction. Therefore, it's necessary to calculate the offset $D(i)$ between the frames and project the frame offset to Oxz plane according to the direction angle δ, and then calculate the offsets on x axis and on z axis respectively to solve the problem, as shown in Fig. 4.

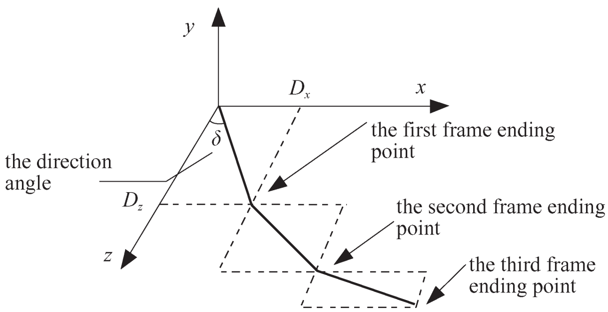

Fig. 4 Matched method of position and direction

The formula to calculate offset between the frames are:

$$D_x(i) = (T_x(i) - T_x(i-1)) \times \sin\delta$$

$$D_z(i) = (T_z(i) - T_z(i-1)) \times \cos\delta$$

Here, $T(i)$ is the displacement of a key frame and i is the number of the frame. When $i = 0$, $T(0) = 0$, $T(-1) = 0$, $D(0) = 0$.

3) The changes in direction of actor models are related to the rotating vectors of top chunks of models. Altering the top chunk's rotating vector R_y which is on the y axial, can change actor's direction.

4) If an actor goes through a walking-running-walking process, it consists of the consistent movements of walking to running and running to walking. The general way to solve it is to play the walking animation first, and then to reiteratively play the running animation, which would make the two movements' connection inconsistent. A better solution proposed is to automatically use the adding frame technique. When an actor is walking and suddenly receives a running command, the walking animation renders to a frame that is stochastic. This stochastic frame will be identified as the first key frame. The second key frame is the aim pose, which should be a smooth transition. Between the two frames, Eq. (2) can be adopted to automatically add a 5–10 frame animation to achieve a smooth transition in different movements. Choosing the second key frame plays a vital role in creating a smoothing effect. Generally speaking, the more similar the destination frame is to the start frame, the better the smoothing effect is after automatically adding frames. According to analyses of actors' walking and running animation, key frames of walking and running are similar. Therefore, we just need to make the running key frame as a destination frame.

5) In the real world, people's walking postures vary from one to another; and virtual actors are the same. However, it will waste much manpower and time to edit each walking posture of each virtual actor by action editors. A variety of different walking postures can be created by adjusting the amplitude and frequency of virtual actors' walking. The frequency can be adjusted by changing numbers of added frames between two key frames. The more added frames, the lower the actor's walking frequency. In the meantime, walking postures of actors are determined by the rotating vector of each skeleton in a local coordinate system, and rotating vector of the corresponding skeleton multiplied by t can adjust the amplitude. For example, while running, arms move back and forth alternately, i.e., the arms' corresponding rotating vectors in a coordinate system continuously change. Suppose that in standard conditions the ante-brachium, which concerns x axial direction, rotates 15°. As long as ante-brachium rotating vectors are multiplied by 2, movement amplitude can increase to 30°. Nevertheless, it cannot go beyond the movement restriction. Simply multiplying by t $(0<t<1)$ decreases the movement amplitude. Because the virtual actor's skeleton model is made up of different chunks, it is better to multiply different t for each chunk to achieve individual walking postures. To directly obtain new movement from two or more movements that have already been done, the t value of each movement is just reset. For example, to converge running and walking postures together, the new movement should have $a\%$ running and $b\%$ walking, where $a\%+b\%$ equates to 100%. By adopting the method mentioned above, a certain skeleton's mixed rotary parameters can be created. In addition, it can also obtain new movements which are mixed by several movements in different degrees.

4.2.2 Way of path point control

Movement control of virtual actors can also be achieved by pre-establishing path points as shown in Fig. 5. Set path points A, B, C and D in a virtual scene and connect them in a straight line. Set movement of actors at each path point, such as walking, running, and marking time. Actors' walking direction is determined by a spatial straight line's direction, and the number of rendered

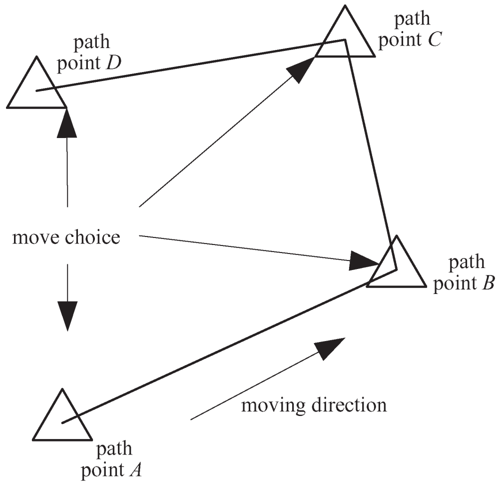

Fig. 5 Virtual actor moves along path point way

frames between the two path points determines the walking speed.

5 Application

Adopting the above method with fltk and OpenGL graphic library, a virtual actor's movement edit and control software are designed on a PC. Figure 6(a) is the virtual actor's movement edit system, which uses key frame technique. Figure 6(b) is the virtual actor's real-time movement control system. It can control movement manners, direction and speed of virtual actors through the keyboard. Figure 6(c) is a 3D scene edit system in which the virtual actor moves along a path point way. And Fig. 6(d) is using the virtual actor's movement edit and movement control method introduced in this paper to achieve a criminal 3D reappearance.

6 Conclusions

This paper investigates a method to automatically create a virtual actors' skeleton model, based on research focused on the virtual actor walking model, movement restriction, movement animation, real-time movement control and path point control. Simulation application shows that the automatically created skeleton model can meet the requirements of joint control. Moreover, it can also meet the needs of taking an actor's movement as the main basis in simulation. This lays the groundwork for virtual maintenance and criminal 3D reappearance applications.

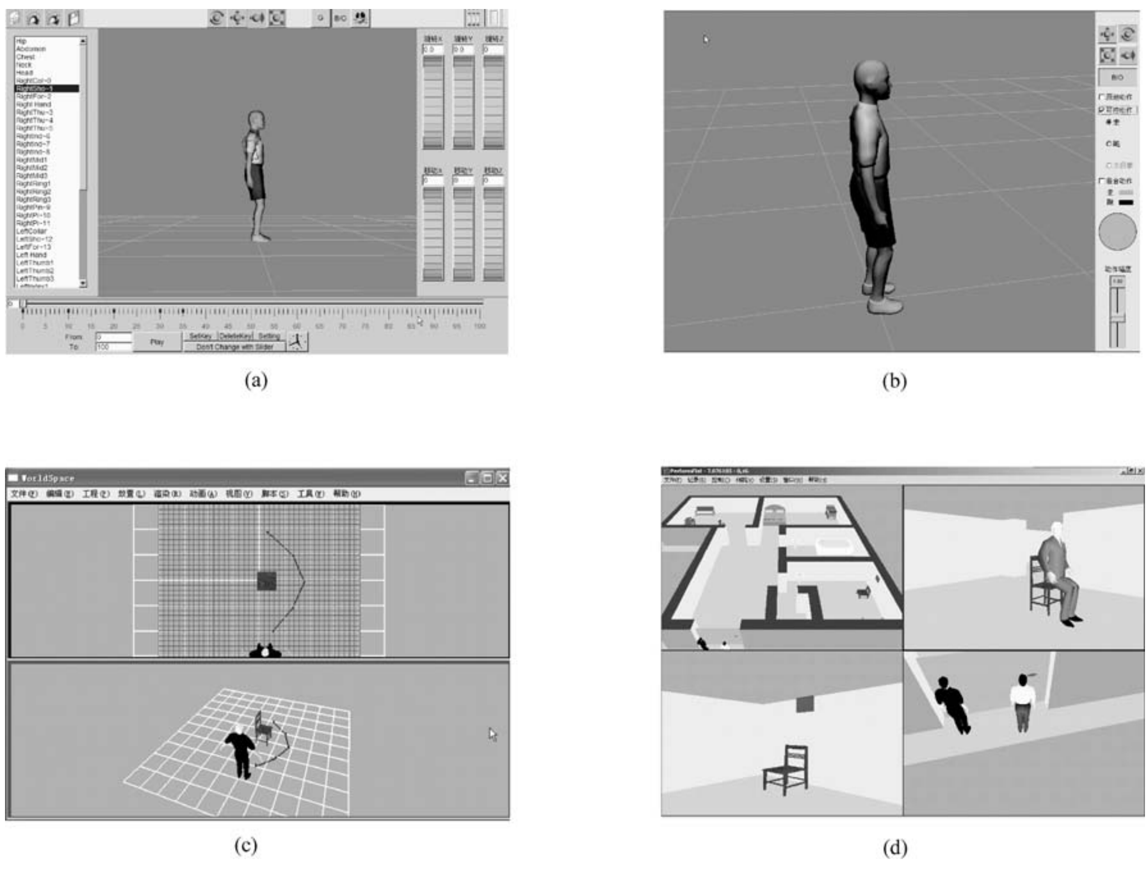

Fig. 6 Application of virtual actors. (a) Movement edit system; (b) movement control system; (c) movement control along path point way; (d) apply virtual actors in criminal 3D reappearance software

Acknowledgements This work was supported by the Southeast University Fund (No. 9222001286).

References

1. Musse S R, Thalmann D. Hierarchical model for real time simulation of virtual human crowds. IEEE Transactions on Visualization and Computer Graphics, 2001, 7(2): 152–164
2. Seron F J, Rodrigues R, Cerezo E, et al. Adding support for high-level skeletal animation. IEEE Transactions on Visualization and Computer Graphics, 2002, 8(4): 360–372
3. Aubel A, Boulic R, Thalmann D. Real-time display of virtual humans: levels of details and impostors. IEEE Transactions on Circuits and Systems for Video Technology, 2000, 10(2): 207–216
4. Lu X J, Li Y, He H G. Modeling and simulation of articulated virtual human's arm action. Journal of System Simulation, 2005, 17(2): 410–413 (in Chinese)
5. Abaci T, Ciger J, Thalmann D. Speculative planning with delegation. In: Proceedings of the 2004 International Conference on Cyberworlds, 2004, 100–106
6. Kalra P, Magnenat-Thalmann N, Moccozet L, et al. Real-time animation of realistic virtual humans. IEEE Computer Graphics and Applications, 1998, 18(5): 42–56
7. Wang Z Q, Zhang Yongdong, Xia Shihong. 3D human motion simulation and a video analysis system for sports training. Journal of Computer Research & Development, 2005, 42(2): 344–352 (in Chinese).
8. Zhuang Y T, Liu X M, Pan Y H. A video-based human skeleton extraction technique for animation. Journal of Computer Research & Development, 2000, 37(4): 498–506 (in Chinese)
9. Hong B R, He H Q. The realize for virtual human walk and run movement. Journal of Computer Application Research, 2000, 17(11): 15–19 (in Chinese)
10. Kiss S. 3D character modeling in virtual reality. In: Proceedings of the Sixth International Conference on Information Visualization, 2002, 541–548