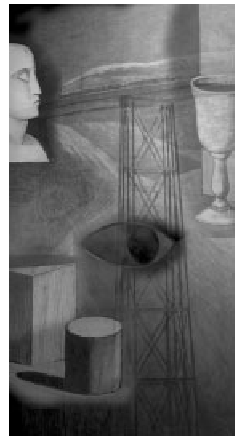
THE JOURNAL OF VISUALIZATION AND COMPUTER ANIMATION
J. Visual. Comput. Animat. 2002; **13:** 275–285 (DOI: 10.1002/vis.298)

Multiple animated characters motion fusion



By Luo Zhongxiang, Zhuang Yueting, Liu Feng and Pan Yunhe*

One of the major problems of the motion capture-based computer animation technique is the relatively high cost of equipment and low reuse rate of data. To overcome this problem, many motion-editing methods have been developed. However, most of them can only handle one character whose motions are preset, and hence cannot interact with its environment automatically. In this paper, we construct a new architecture of multiple animated character motion fusion, which not only enables the characters to perceive and respond to the virtual environment, but also allows them to interact with each other. We will also discuss in detail the key issues, such as motion planning, coordination of multiple animated characters and generation of vivid continuous motions. Our experimental results will further testify to the effectiveness of the new methodology. Copyright © 2002 John Wiley & Sons, Ltd.

Received: November 2001; Revised: April 2002

KEY WORDS: animation; motion fusion; motion editing; motion collaboration

Introduction

Animation based on motion capture has become one of the most promising areas in computer animation. But the motion capture equipment is costly and the movement of the players is restricted by the sensors. To be able to reuse the data and make new movements, we need to modify or edit the motion capture data before retargeting it to animated characters. Recently, extensive study has been done on these issues, which basically consists of three motion-editing techniques: (1) movement curve fitting and control points adjusting method;[1] this method is not suitable when the movement is big; (2) data signal processing;[2,3] this promotes reuse of the existing data; for example, motion warping, presented by Witkin and Popvic,[4] uses blending and overlapping algorithms to generate new motion based on existing motions. Bruderlin regards motion as signal. Signal-processing techniques such as multi-resolution filter, time warping, multi-target motion interpolation, motion wave-shaping and motion displacement mapping are used to regenerate motions based on captured motion data.[5] (3) Space-time constraints method,[6,7] which is suitable for generating interactive motion. It generates constrained movements by solving objective movement equations.[8–10] Using this method, Popvic and Witkin[11] were able to transform captured motion into new motions that preserve the original properties of motion and satisfy users' specifications, by solving dynamic equations. Rose *et al.*[12] were able to generate the transition of two clips of motion. Lee and Shin[13] were able to generate constrained motion using curve fitting and inverse kinematics. But each instance requires its unique equations and solutions, and it is hard to generalize.

The above-mentioned methods have largely solved the problem of reuse of data and regeneration of motion. The problem is that they can only handle one character, while in reality, multiple animated characters move simultaneously in one scene. The space-time constraint, imposed by the virtual environment and other characters, also changes constantly. Because the original motion is captured in a structural environment, before the data is retargeted to animated characters in a complex, non-structural environment, motion editing is always needed. We think it is important to enable the automatic perception of virtual scenes and the collaborative work of multi-characters.

In this paper, we present the idea of motion fusion of multiple animated characters and the approaches to fuse multiple motions into one non-structured virtual scene by improving the perception and self-decision-making abilities of characters. Our basic idea is that, first, we

*Correspondence to: Luo Zhongxiang, Department of Computer Science and Engineering, Microsoft Visual Perception Laboratory, Zhejiang University, Hangzhou 310027, PR, China. E-mail: zhongxiangluo@yahoo.com

Contract/grant sponsor: National Natural Science Foundation of China; contract/grant number: 69803009.
Contract/grant sponsor: Foundation of Education Ministry for Excellent Young Teachers.

Copyright © 2002 John Wiley & Sons, Ltd.

apply motion decision-making and motion collaboration to perceive the virtual environment and other moving characters automatically. Second, a definite state machine (DSM), dealing with motion mode and space-time constraints imposed by the ambient scene and other moving characters, is utilized to select a suitable motion mode for each character intelligently. Finally, for each character, available space-time constraints can be applied to solve continuous movement based on constraints imposed by other characters, ambient virtual scene and animators. Different from available space-time constraints, our methods automatically perceive the space-time constraints and improve the perception and decision-making of each animated character.

The Architecture of Multiple Animated Characters Motion Fusion

We assume that human movement is a four-layer thinking-and-moving process. The process of animated characters movement is analogous to human movement, as shown in Table 1. Motion decision-making, motion collaboration, motion solving and motion execution are directly next to implement motion planning, collaboration, solving and displaying.

Based on the analogy between human movement and animated character movement analysed above, the general architecture of multi-character motion fusion is conceived as consisting of four parts as illustrated in Figure 1:

1. Motion decision-making layer: major steps are, first modelling virtual scenes and writing the scripts of producing animation; then planning multiple animated characters' paths in non-structured virtual scenes and breaking these paths into single character's path.
2. Motion collaboration layer: character–character collaboration and character–environment collaboration are realized by settling two basic problems: (a) decision-making of discrete motion mode; (b) generating space-time constraints imposed by virtual scenes and other characters.
3. Motion-solving layer: two main steps are transforming captured motion and resolving continuous movement. The first step transforms the captured motion and gets the approximate pose at each instance for each character, while the latter solves continuous motion of every discrete motion mode.
4. Motion execution layer: we retarget regenerated continuous movement to different animated characters in a complex non-structured environment, and produce one clip of vivid animation of multiple animated characters moving collaboratively and simultaneously.

In the study of fusing multi-character motion, there are two main challenges: (1) animated characters' ability of self-determination to select the optimum path in non-structured virtual environment is limited; (2) the decision-making strategy of the discrete motion mode of each animated character and the solving of continuous movement along a specific path are difficult to implement. Discrete motion mode means the preferential motion mode, such as walking, running and jumping, selected by one character moving on the specific path, while continuous movement means a sequence of poses for one character moving on the specific path in a period of time. Here we apply the idea presented by Kalisiak to

Layer	Motion of human	Motion of animated character
Motion decision-making	The path and behaviour are decided by the brain	Animator plans the routine and process of animated character movement
Motion collaboration	Breaking motion decision into units, and collaborate with ambient scenes	Computer collaborates the relation among characters and environment, and achieves the discrete motion strategy in each part of the path
Motion solving	The specific pose at one time instant is solved	Computer solves the continuous movement of each animated character
Motion execution	Each part of human executes the solved motion	Motion data drive the animated characters to move

Table I. The analogy of human movement and animated characters movement

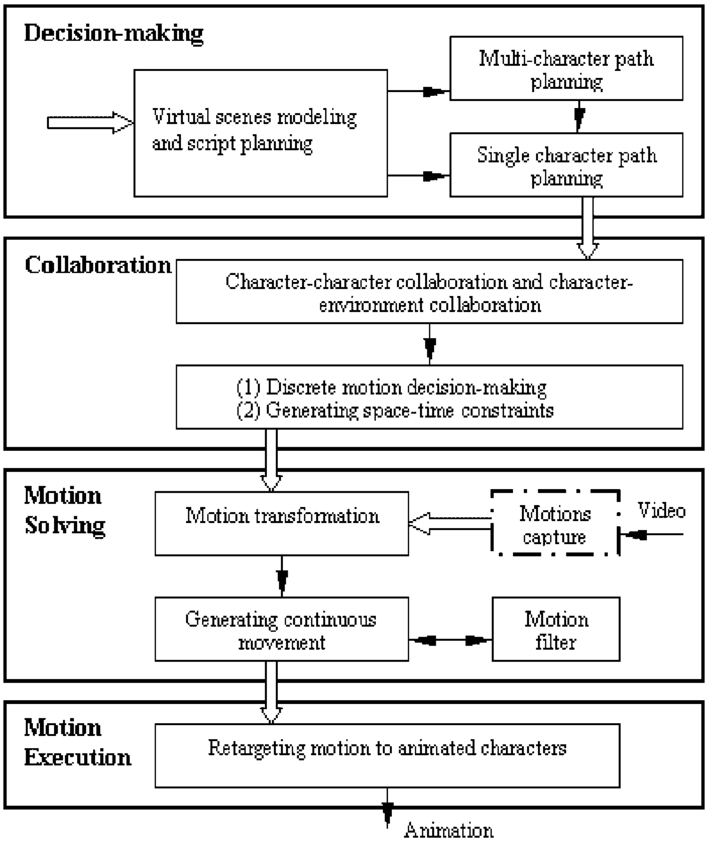


Figure 1. General architecture for motion fusion.

path planning for multiple characters. A more thorough introduction can be found in Kalisiak and van de Panne.[14]

In the later discussion, we will give the idea and approaches to fuse multiple animated characters' motion into one scene and show the experimental results.

Collaboration of Multiple Animated Characters

When we fuse multiple motions into one scene at the same time, various characters should collaborate with each other in thought and action, and work without collision. On the other hand, a complex virtual environment always imposes various constraints on the characters; thus each character must also collaborate with the virtual environment. As a rule, in motion fusion of multiple animated characters, collaboration includes character-to-character collaboration and character-to-environment collaboration. In the study of motion fusion, the key issues relating to motion fusion are motion collision detection and avoidance, motion mode selection for each character moving on the specific paths, analysis and specification on space-time constraints imposed by virtual environment and other characters moving in the same scene, and so on.

Collaboration Between Characters and Virtual Environment

Nowadays, most commercial software devoted to computer animation and games runs in a structured virtual environment. When augmenting the vividness and diversity of the virtual environment, we need to make a further study of the effects on character movement resulting from the diversity and complexity of the virtual scene as well as the bumpiness and stochasticity of planned paths.

Generally, there are several steps in the collaboration between animated characters and the virtual environment. First, space-time constraints imposed by the environment are analysed and specified. Second, by taking

Copyright © 2002 John Wiley & Sons, Ltd.

the space-time constraints as input, a definite state machine is applied to deduce the motion mode of animated characters satisfying the space-time constraints. Some of the constraints imposed by the virtual scene are specified through a human–computer interface; others are detected and specified by our system automatically.

Collaboration among Characters

Collaboration in Action. Collaboration in action consists of active collaboration and passive collaboration. Active collaboration means that various animated characters, like autonomous agents, readily react to the ambient environment and other characters. For example, if one character (A) finds another one (B) at time instant t, A will make a self-decision of whether to shake hands with B or not. Passive collaboration means that the animator specifies the action of animated characters in advance in order to accomplish one assigned task. For example, the animator specifies that two characters accomplish handshaking in a specific position at time instant t.

In essence, the collaboration between animated character and environment can be regarded as the collaboration between active character and immobile character. Thus we can apply a DSM to deduce the mode of motion in a specific path, and then solve continuous movement by space-time constraint approaches and the numerical optimum method.

Collaboration in Locomotion. Except for the action collaboration among multiple animated characters, multi-character motion fusion should guarantee the collaboration of various characters in motion. As an example, let us consider the situation of two characters moving on the same path, where they may collide with each other; or the situation of passing through a bridge which actually only allows one person through at a time. This bridge is regarded as an exclusive resource an computer technology. If two characters want to grab an exclusive resource synchronously, a deadlock will occur. Therefore, before solving continuous movement for each animated character, we need to detect the probability of collision of various characters and deadlock resulting from grabbing exclusive resources.

(1) Collision among multiple animated characters: As a rule, any two rigid human bodies should not penetrate each other. To avoid collision before deadlock occurs, motion simulation is applied to detect collision. Once collision has been detected, motion mode will be updated to avoid collision accordingly.

Most of the presented approaches to detect collision are reasonable but complicated to implement. In motion fusion of multiple animated characters, we assume that collision takes place when the Euclidean distance between character m and character n is less than a maximal threshold $d_{\max}$. When $\|q_m(t) - q_n(t)\| \leq d_{\max}$, we calculate the distance between one joint in one character and a triangular mesh composed of joints in the other character. We then detect the point of intersection. If the point of intersection is found, it means that collision will occur in the process of movement; otherwise, two characters will work smoothly. Based on a knowledge of physiology, the maximal threshold can be regarded as approximately half of the height of the human model.

Once collision has been detected, there are different strategies to avoid it: (1) maintain the former speed, update the direction of movement; (2) maintain the former direction, update the relative speed to other characters; (3) change both the speed and direction.

(2) Management of exclusive resources: The commonly used approach to control exclusive resources is by using tokens. But this cannot guarantee efficiency, and therefore does not fit for managing non-exclusive resources. In a collaborative virtual environment, different characters have different authentication to apply to and operate on an exclusive resource; therefore this approach cannot distribute an exclusive resource equally. In order to overcome the shortcomings of approaches using tokens in a multiple animated characters motion fusion, we apply a two-layer management strategy to control share of exclusive resources during characters' movements: (1) detect all exclusive resources in the virtual environment, and then detect the characters using exclusive resource; (2) based on the detected results, we control concurrently the characters' movement using preference.

Discrete Motion Decision-Making

In the character-to-character collaboration and character-to-environment collaboration, a character's movement is constrained by the ambient scene. From the viewpoint of vividness and naturalness, in certain phases of the path, different reasonable motion modes should be selected. Therefore, depending upon the constraints imposed by other characters and the virtual environment, preferential motion modes suitable for

Copyright © 2002 John Wiley & Sons, Ltd.

specific phases of the path, such as walking, running, climbing and jumping, should be selected before solving continuous movement.

The discussions above are mainly to perceive the space-time constraints imposed by the ambient virtual environment and other moving characters. In the later discussion, we need to make reasonable decisions based on these constraints in order to help the virtual characters to select preferable motion modes. In fusing motions of multiple virtual characters, a DSM is introduced to make intelligent decisions regarding motion modes for the virtual characters. Creating a database of constraints and a database of characters' behaviour is essential for intelligent decision-making. The database of constraints cover the constraints imposed by the ambient virtual scene, such as barrier, tunnel and threshold, and the constraints imposed by other moving virtual characters, such as shaking hands, nodding, stooping down and standing aside. The database of behaviour covers all possible motion modes in the unstructured virtual environment.

To select a preferential mode of motion from various types of motions, a DSM is introduced to solve discrete motion. We introduce DSM not only as a method to select the motion mode, but also as a breakthrough in traditional computer animation directed by an animator. By using a DSM, we implement an active collaboration methodology in computer animation, which enables characters to perceive the ambient environment, to make self-decisions of action and movement according to the constraints imposed by the ambient scene and other characters.

A DSM for selecting the motion mode can be depicted as a set $(\boldsymbol{Q},\Sigma,\boldsymbol{\delta})$ composed of triple elements. $\boldsymbol{Q}$ is a definite set of states in which each element is one type of motion mode, Σ is a finite set of inputs in which each element is a space-time constraint imposed by other characters or virtual scene. $\boldsymbol{\delta}$: $\boldsymbol{Q} \times \Sigma \rightarrow \boldsymbol{Q}$ is the conversion function of states, which means that when character is in state $q_0 (q_0 \in Q)$, if the constraint $\omega (\omega \in \sum)$ is input, the DSM will output a definite state $q_1 (q_1 \in Q)$.

An example of DSM as shown in Figure 2 to explain how it realizes discrete motion decision-making. Let the motion mode be $\boldsymbol{Q}$ = {skipping, walking, running, nodding, spanning, handshaking}. If the motion mode for one character is q = walking, space-time constraints imposed by the virtual environment and other characters are *con*, satisfying $con \in \Sigma' = \{a_0, a_1, a_2, a_3, a_4, a_5\}$; the mode $q_1 \in Q'$ = {skipping, running, nodding, spanning, handshaking} will be selected. For example, if *con* is meeting people, the motion mode of character will be changed to handshaking. When *con* encounters a trap, the motion mode of character will be changed to spanning.

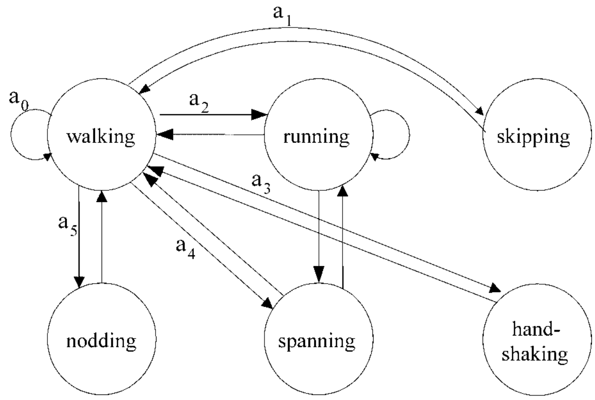


Figure 2. Example of DSM.

Solving Continuous Motions

Approximating Continuous Motions

To enable animated character moving on the planned path in a virtual environment which differs from the realistic one, we transform the captured motion to make the character pass the planned path first and get the approximate pose at each instance for each character moving in the new scene, which simplifies the process of solving continuous movement of each character. We use Euclid angle and translation vector to represent the movement sequences of the animated character. The original motion is then transformed in order to cover the planned path using cycle extension and displacement mapping.

Human Model. The human model adopted in the video-based motion capture system developed by our lab is different from the markers and special reflective objects adopted in Fua *et al.*,[15] which are costly and inconvenient. We have designed a suit of tight clothing. At each joint of the tight clothing there is a block with a unique colour. Compared with special markers or sensors appended to the body, it has several advantages: (1) it is inexpensive, easy to implement and convenient to track; (2) it is free from constrained movement. Based on the tight clothing, we have defined the corresponding human body as a set of rigid body parts connected by joints, and human motion as the movement of the

Copyright © 2002 John Wiley & Sons, Ltd.

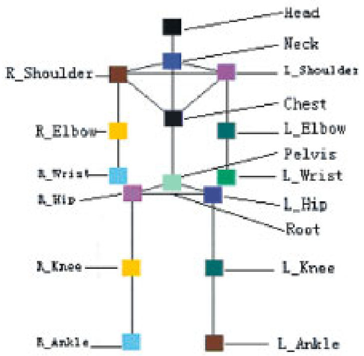


Figure 3. Human model.

human skeleton. Figure 3 is the adopted skeleton model, which consists of 16 joint points. Our target of motion capture is to extract the 3D human body movement sequences of each colour block's centre on the tight clothing.

Motion Sequence. Depending on motion capture data and human motion as rigid body movement, we can represent the motion sequence $\boldsymbol{Q}(t)$ as:

$$\boldsymbol{Q}(t) = (\boldsymbol{q}(t), \boldsymbol{p}_0(t), \ldots, \boldsymbol{p}_n(t)) = (\boldsymbol{q}(t), \boldsymbol{p}(t)) \qquad (1)$$

where $\boldsymbol{q}(t)$ is the translation vector of point root (as seen in Figure 3) at time t, and $\boldsymbol{p}_i(t)$ is the rotation angle of joint i at time t.

Cycle Extension. The frame number of captured motion may be too small to be used directly in an animated character of a new environment; therefore, cycle extension should be used to extend the cycles of motion. If the movement sequence of original motion in cycle T is $\boldsymbol{Q}(t) \sim \boldsymbol{Q}(t+T)$, the orientation to extend will be $(\boldsymbol{p}(t+T) - \boldsymbol{p}(t))/\|\boldsymbol{p}(t+T) - \boldsymbol{p}(t)\|$. The corresponding movement of n cycles will be depicted as:

$$\boldsymbol{Q}'(t) = \boldsymbol{Q}(t) \bullet (n(\boldsymbol{q}(t+T) - \boldsymbol{q}(t)), (\boldsymbol{p}(t+T) - \boldsymbol{p}(t))/\|\boldsymbol{p}(t+T) - \boldsymbol{p}(t)\|)$$

Displacement Mapping. Since the start point and direction of movement of different paths are different from the original motion, displacement mapping should be applied to transform the original motion. Let original motion be $\boldsymbol{Q}(t)$ and transformed motion be $\boldsymbol{Q}'(t)$ after applying displacement mapping to $\boldsymbol{Q}(t)$. Let the displacement vector from $\boldsymbol{Q}(t)$ to $\boldsymbol{Q}'(t)$ be $(\boldsymbol{U}(t),\boldsymbol{V}(t))$. $\boldsymbol{U}(t)$ and $\boldsymbol{V}(t)$ are translation vector and rotation angle at time instant t, respectively. $\boldsymbol{T}(t)$ is the translation vector from $\boldsymbol{Q}(t)$ to the original point of the world coordinate system at time instant t. Displacement mapping can be depicted as:

$$\boldsymbol{Q}'(t) = \boldsymbol{Q}(t) \oplus (\boldsymbol{U}(t), \boldsymbol{V}(t)) \qquad (2)$$

We can work out $(\boldsymbol{U}(t),\boldsymbol{V}(t))$ depending upon the planned paths. We can calculate the movement sequences which make animated characters move on the planned paths using equation (2).

A few artificial motion properties, such as slipping and two legs hanging in the air, will be involved in the initial motion resulting from above motion transformation. The reason for these artefacts is that some essential space-time constraints are not imposed by the ambient virtual scene or animators. Therefore, we need to apply motion collaboration among the characters and the virtual environment to produce enough essential constraints for solving natural continuous movement for each character in the following subsection.

Solving Continuous Motions Based on Space-Time Constraints

In a scene, specific space-time constraints should be imposed for multiple animated characters to cooperate with each other at the same time. The aim of a space-time constraint method presented by Gleicher is to produce motion, which satisfies space-time constraints and minimizes objective function of movement. Generally, space-time constraints specify what a character will do in a certain space and time, while objective function specifies how to accomplish this motion. Continuous motion solving produces a dynamic motion, which satisfies both the specified space-time constraints and objective function.

Space-Time Constraints. Space-time constraints are special restrictions on a character's movement imposed by the virtual environment and other characters in time and space. It specifies what the character does, which makes the character move depending upon ambient circumstance but preserves the vividness and inherent properties of motion. For space-time constraints, it can be specified dynamically or cinematically. Dynamically, the constraints to be considered are very complex. Therefore, only kinematic space-time constraints are generally considered so as to limit the movement of

Copyright © 2002 John Wiley & Sons, Ltd.

character in space and time domains. Typically, space-time constraints are described as a set of equations. In continuous motion resolving, differential constraints are specified during a period of time, and they can be transformed to the constraints at one time instant. We solve each frame to satisfy constraint functions and objective function in movement sequences.

Space-time constraints are commonly described as a set of constrained equations, which can be described as:

$$\boldsymbol{F}(\boldsymbol{q}(t), \boldsymbol{p}_0(t), \ldots, \boldsymbol{p}_n(t)) = \boldsymbol{C}$$

where t is the time instant of character movement, $\boldsymbol{q}(t)$ and $\boldsymbol{p}_i(t)$ are the parameters of character movement and $\boldsymbol{C}$ is a constant.

Major space-time constraints used in continuous movement resolving include:

1. Position constraints: $\boldsymbol{q}(t) = \boldsymbol{C}_0$, where $\boldsymbol{C}_0$ is the position constant.
2. Pose constraints: $\boldsymbol{p}_i(t) = \boldsymbol{C}_1$, where $\boldsymbol{C}_1$ is the Euclid angle constant.
3. Position constraints supporting animated character body: $\boldsymbol{q}_i(t) = \boldsymbol{C}_2$, where i is joint i and $\boldsymbol{C}_2$ is the position constant.

Objective Function of Movement. Objective function concerns how to accomplish a motion. In resolving continuous motion, after specifying objective function for character movement, the system is able to select an exclusive solution from a set of reasonable solutions. In general, it is difficult to get a perfect objective function, and users directly use constraints as objective function, the interactive performance of which enables the user to make adjustments by adding more constraints. Gleicher pointed out that there are some drawbacks to simply minimize the magnitude of the parameter vector. One particular problem is that different parameters often have vastly different effects.[8] To overcome these problems we use a weighted sum-of-squares of the parameters. Our objective can be seen as an approximation to the function that minimizes the displacement from the ideal position.

In motion fusion, two major objective functions are:

1. Minimizing the difference of position between ideal joints and practical joints, which can be described as:

$$\text{Min}\left(\text{sum} = \int_{t_{\text{start}}}^{t_{\text{end}}} \sum_{k=1}^{16} W(t,k)\|p_k(t) - \hat{p}_k(t)\|_2 \mathrm{d}t\right)$$

   where $p_k(t)$ is the coordinate of joint k after motion fusion, $\hat{p}_k(t)$ is the coordinate of joint k before motion fusion, t_{start} and t_{end} is the start and end in the region of motion edition, respectively, and $W(t,k)$ is a weighted factor of joint k at time t.
2. Minimizing the difference of movement before and after motion editing, which can be described as:

$$\text{Min}\left(\text{sum} = \int_{t_{\text{start}}}^{t_{\text{end}}} W(t)\|\boldsymbol{Q}(t) - \boldsymbol{Q}'(t)\|_2 \mathrm{d}t\right)$$

   where $\boldsymbol{Q}(t)$ is the pose after motion editing, $\boldsymbol{Q}'(t)$ is the ideal pose and $W(t)$ is a weighted matrix.

Re-solving Continuous Movement. The collision and constraints in motion fusion are described as a set of equations; therefore, we can view the motion fusion problem as constrained numerical optimization. Non-linear equations cover potentially large numbers of constraints and variables since we create a single problem for multiple motions, which makes solving multi-character motion fusion all the more challenging.

For simplicity, We can build approximations of the non-linear problem for solving non-linear problems. Iteration algorithms and initial solution, discussed above, are used to obtain an exact solution. We use the algorithms discussed in Gleicher,[8] which belong to a class of sequential quadratic programming that has linear constraints and quadratic objective functions. Moreover, only equality constraints are considered in our solver.

Motion Rectification

The solved motion after discrete motion decision-making and continuous motion solving reveals a few limitations: (1) jitter resulted from linear interpolation, which lacks vividness and naturalness; (2) bumpiness of motion trajectory was because continuous motion solving does not consider the smoothness of the trajectory; (3) an unnatural motion mode was selected for some characters. Therefore, motion rectification, namely the smoothing of the solved motion, must be applied.

Filter of Motion Trajectory. From the discussion above, we know that the translation vector of motion is composed of a set of discrete moving points. We adopt the following motion filter method: fit the discrete points, and then generate the curve of motion, which minimizes the difference between the specified trajectory and the solved curve. The generated continuous curve is then used to update the translation vector of motion sequence.

Copyright © 2002 John Wiley & Sons, Ltd.

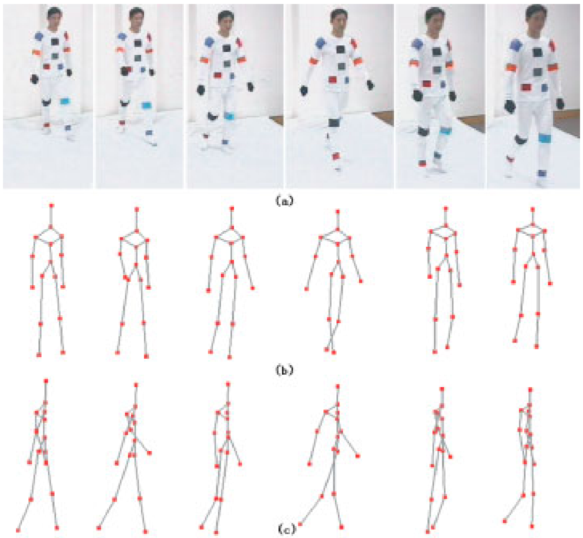


Figure 4. Captured motion of walking.

Imposing Constraints of Feasible Motion. Usually, it is impossible to create transitional motion by linear interpolation. For instance, the body may hang in the air. We reimpose feasible constraints on the motion and use inverse kinematics to solve the position of each joint for vivid motion.

Filter of Limb Movement. By using a Kalman filter, we can remove noise and overcome the inconsistency of limbs resulting from the above methods.

Experimental Results

Based on the discussion in the previous sections and other research,[16] our lab has developed a video-based human animation (VBHA) system with two cameras. It is implemented with Visual C++ and can run on the Windows X platform. Its main functions include calibration, feature tracking, 3D reconstruction, motion editing, multiple animated character motion fusion and production animation.

Figure 4 shows the results of walking motion captured from the video sequences of walking. Two general CCD cameras are used to capture walking motion performed by the player in tight clothing. Figure 4(a) presents key frames in video sequences. Figure 4(b) and (c) show 3D motions extracted from video. Figure 4(b) presents the captured 3D human model sequences from elevation. Figure 4(c) presents human model sequences from side elevation. These figures demonstrate the effectiveness and vividness of our motion capture.

In Figure 5, we demonstrate the main ideas of multi-character motion fusion. The motion trajectories of

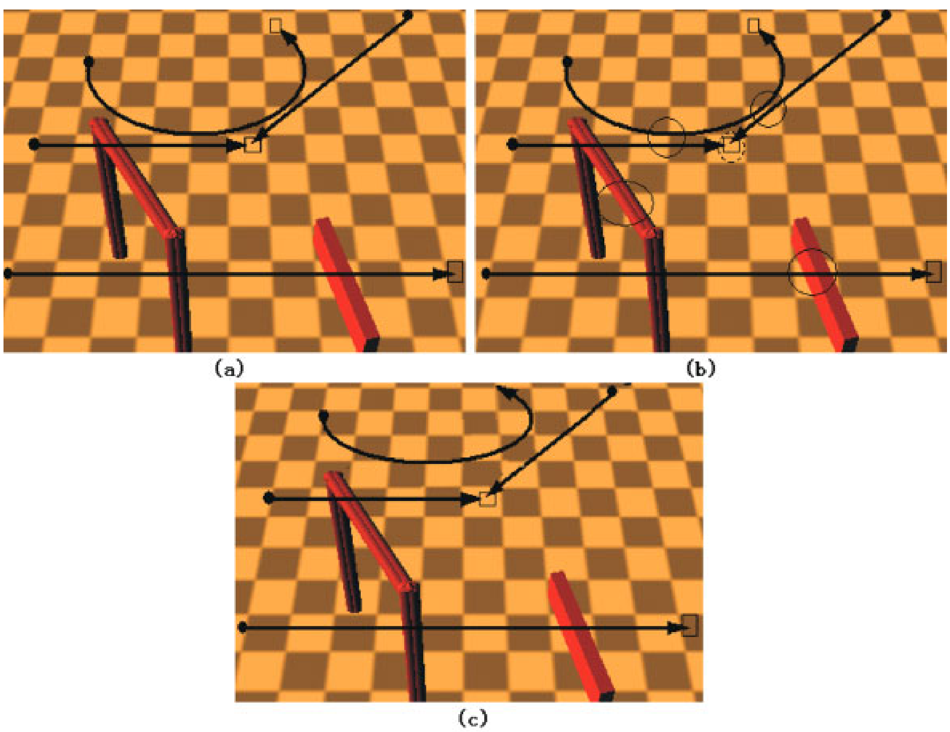


Figure 5. Process of motion collaboration and solving.

Copyright © 2002 John Wiley & Sons, Ltd.

four characters planned by the animator are shown in Figure 5(a); the spheres denote starting points and the rectangles denote ending points of animated characters. Figure 5(b) shows four possible collisions of multiple animated characters' motion detected by the system (denoted by a circle) and the functions specified by the animator (denoted by a dashed circle). Discrete motion modes and constraints, which result from motion collaboration and collision detection, are transported to the motion-solving layer. The trajectory of continuous movement solved by our system is seen in Figure 5(c).

To test the feasibility and efficiency of the multi-character motion fusion approach presented in the paper, we fuse multiple motions of animated characters into one scene with captured motion data. Figure 6 shows the experimental result that fuses motions of four characters into one non-structured virtual environment with a door and a barrier. Animator specifies initial motion trajectories of each character and passive collaboration, such as handshaking, while the system automatically selects the motion mode and produces continuous movement of each character. In the motion collaboration layer, once having detected the door in the scene, the system selects the motion mode depending on the relative height of the character and the door. Similarly, when the barrier is detected, the system makes the decision of spanning the barrier independently. In the motion-resolving layer, continuous movements of each character are generated based on space-time constraints and motion mode. From Figure 6, we can see that feasible discrete motion mode and vivid continuous movement can be generated by our system.

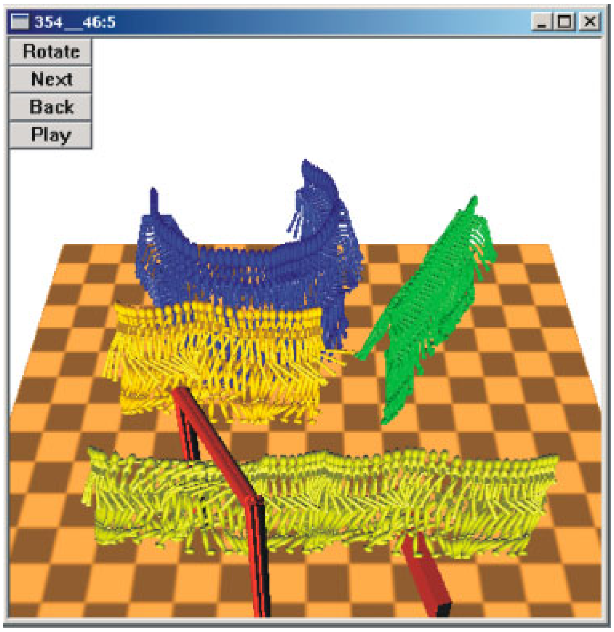


Figure 6. Multiple animated characters motion fusion.

Analysis of the results of our system revealed three limitations of fusing the motions of multiple virtual characters.

First, it is essential for our system to save the 3D position of the virtual environment beforehand. Unlike the human, our system cannot reconstruct a 3D virtual environment automatically. Consequently, a complex virtual environment is still not practical.

Second, a DSM can make an intelligent decision of motion mode. But a perfect database of constraints and behaviour is required, which is difficult to obtain. Moreover, once a suitable rule of DSM for some character is not found in the database, DSM will fail. As the result, we think that machine learning should be introduced to improve the decision-making of DSM.

Finally, in fusing the motions of multiple virtual characters, our system has not finished planning the path for each character automatically. At present, we specify the path for each character manually.

Conclusions

This paper presents a new idea of multi-character motion fusion and its relevant implementation approaches. By fusing motions captured in a structured environment into one non-structured virtual environment, we studied and implemented the collaboration between characters as well as the collaboration between character and virtual environment. The DSM method, adopted in solving discrete motion mode, enhances the perception and self decision-making of the virtual characters as well as improving the automation degree of multiple animated characters working cooperatively in complex circumstances. Moreover, motion transformation presented in this paper improves the reusability of motion capture data and simplifies the process of solving continuous movement by redirecting original motion data to a new path in a virtual environment through cycle extension and displacement mapping. High automation and high reusability make our presented approach highly applicable in computer animation and computer games.

Further studies on our multi-character motion fusion approach should be concentrated on the following. (1) In discrete motion mode decision-making, the number of rules in DSM determines the correctness of reasoning.

Copyright © 2002 John Wiley & Sons, Ltd.

A better decision-making capability should be studied for a complex environment. (2) In motion fusion, constraints imposed by a complex virtual environment increase as the number of the characters increase, and computation complexity rises. Further research is expected to improve the efficiency of the algorithm and lower the computation complexity. (3) There is a trend for studying autonomous cooperation among several characters in the computer animation field. We should lucubrate the artificial life science application in computer animation.

ACKNOWLEDGEMENTS

This work is supported by grants from the National Natural Science Foundation of China (No. 69803009) and the Foundation of Education Ministry for Excellent Young Teachers. We would like to thank Xu Yi and Chen Yisheng for creating the Open-GL animations. We are grateful to Li Yuemei for fruitful discussions.

References

1. Ko H, Badler NI. Straight line walking animation based on kinematic generalization that preserves the original characteristics. In *Proceedings of Graphics Interface '93*, May 1993; pp 9–16.
2. Unuma M, Anjyo K, Takeuchi R. Fourier principles for emotion-based human figure animation. *Computer Graphics (Proceedings of SIGGRAPH '95)* 1995; **29**: 91–96.
3. Wiley DJ, Hahn JK. Interpolation synthesis for articulated figure motion. In *Proceedings of IEEE Virtual Reality Annual International Symposium '97*. IEEE Computer Society Press: Los Alamitos, CA, 1997; pp 157–160.
4. Witkin A, Popvic Z. Motion warping. *Computer Graphics (Proceedings of SIGGRAPH '95)*, 1995; **29**: 105–108.
5. Bruderlin A, Williams L. Motion signal processing. In *Proceedings of SIGGRAPH '95, Computer Graphics Proceedings, Annual Conference Series*, Los Angeles, CA, August 1995; pp 97–104.
6. Cohen MF. Interactive space time control of animation. *Computer Graphics (Proceedings of SIGGRAPH '92)* 1992; **26**(2): 293–302.
7. Liu Z, Gortler SG, Cohen MF. Hierarchical space time control. *Computer Graphics (Proceedings of SIGGRAPH '94)* 1994; 35–42.
8. Gleicher M. Motion editing with space time constraints. In *1997 Symposium on Interactive 3D Graphics, ACM SIGGRAPH*, April 1997; pp 139–148.
9. Gleicher M, Litwinowicz P. Constraint-based motion adaptation. *Journal of Visualization and Computer Animation* 1998; **9**(2): 65–94.
10. Cohen MF. Interactive space time control for animation. *Computer Graphics (Proceedings of SIGGRAPH '92)* 1992; **26**(2): 293–302.
11. Popovic Z, Witkin A. Physically based motion transformation. *Computer Graphics (Proceedings of SIGGRAPH '99)* 1999; 8–13.
12. Rose C, Guenter B, Bodenheimer B, Cohen MF. Efficient generation of motion transitions using space time constraints. *Computer Graphics (Proceedings of SIGGRAPH '96)* 1996; **30**: 147–154.
13. Lee J, Shin SY. A hierarchical approach to interactive motion editing for human-like figures. *Computer Graphics (Proceedings of SIGGRAPH '99)* 1999; 39–48.
14. Kalisiak M, van de Panne M. A grasp-based motion planning algorithm for character animation, http://www.dgp.toronto.edu/~mac/masters/ [January 2002].
15. Fua P, Gruen A, Plankers R, D'Apuzzo N, Thalmann D. Human body modeling and motion analysis from video sequences. In *International Archives of Photogrammetry and Remote Sensing*, Vol. 32, Hakodate, Japan, 1998; pp 866–873.
16. Liu X, Zhuang Y, Pan Y. Video based human animation technique. In *ACM Multimedia '99*, Orlando, FL, 1999; pp 353–362.

Copyright © 2002 John Wiley & Sons, Ltd.

Authors' biographies:

Luo Zhongxiang was born in 1976 in Hunan Province, PR China. He received the BS degree with honours in mechanical engineering from Xiangtan University in 1997, and the MS degree in automation from Zhejiang University in 2000. He is currently studying computer science and engineering for a PhD degree in Zhejiang University. His interests cover video-based motion capture, intelligent animation production, multimedia computing and intelligent CAD.

Liu Feng was born in 1979 in Zhejiang Province, PR China. He received the BS degree with honours in computer science and engineering from Zhejiang University in 2001. He is currently studying computer science and engineering for an MS degree at Zhejiang University. His interests cover video-based motion capture, intelligent animation production, multimedia computing and intelligent CAD.

Zhuang Yueting was born in 1965 in Zhejiang Province, PR China. He received the BS degree in computer science and engineering from Zhejiang University in 1986, and the MS and PhD degrees in computer science and engineering from Zhejiang University. He is currently a full professor in the Department of Computer Science at Zhejiang University. His current research interests include video-based human animation, content-based retrieval of image and video database, multimedia database systems and multimedia reasoning.

Pan Yunhe was born in 1946 in Zhejiang Province, PR China. He received the BS degree in architecture engineering from Tongji University in 1968, and the MS degree in computer science and engineering from Zhejiang University. He is currently a full professor in the Department of Computer Science and the President of Zhejiang University. His current research interests include multimedia computing, intelligent CAD and multimedia reasoning.

Copyright © 2002 John Wiley & Sons, Ltd.