Automatic Motion Generation Based on Path Editing from Motion Capture Data

Xiaoyue Guo, Shibiao Xu, Wujun Che, and Xiaopeng Zhang

LIAMA, CAS Institute of Automation, Beijing, China
NLPR, CAS Institute of Automation, Beijing, China
{xyguo,shbxu,xpzhang}@nlpr.ia.ac.cn, chewj@liama.ia.ac.cn

Abstract. In this paper, we present an integrated framework with easy and effective techniques to generate new motions of a character in real-time, using only a short segment of motion data. In this framework, a sequence of segment copies are strung together and then smoothed by a transition method at a junction of two consecutive segments. We use IK thought for the transition to solve the foot-slide problem at the junctions. Given a new motion path, we combine the IK constraints and the sequence of smoothed motion capture data to adjust the human-like model. Based on only a sample motion segment, our method can generate a sequence of motions following an arbitrary path specified by the user. Furthermore, it is easy to implement and can be applied to highly kinetic motions. The experimental results demonstrate our scheme.

Keywords: Animation, Human walk, Inverse kinematics, Motion capture data.

1 Introduction

Character animation is an important research topic in Computer Graphics community, which mainly focuses on human modeling and human motion simulating. Animation of human walking is a crucial problem. It is getting more and more popular and many synthetic scenes involve virtual human characters in both 3D films and 3D video games. However, realistic human walking generation is still a challenge because we are very familiar with human behaviors and highly sensitive to any factitious motion.

Motion capture technique is helpful to preserve the detail information of the performer's behavior for physical motion realness as much as possible, such as the personality or the mood of the performer. But what we obtain with motion capture equipment is just a set of motion data performed by an actor according to requirements, providing a motion toward a specific direction and lasting a certain time. If the requirement changes or a new one is proposed, it's not only inconvenient but also expensive to hire the actor in special clothes for motion capture to perform the required motion. It highly demands that new motion can be automatically generated from the existing motion data library.

In this paper, we make efforts to solve a problem of generating a realistic motion from capture data but in a different motion path. We propose easy and effective techniques for the free walking animation of human in real time regardless of the new

Z. Pan et al. (Eds.): Transactions on Edutainment IV, LNCS 6250, pp. 91–104, 2010.

path shape. In our method, only a short segment of walking data along a straight line is required. We transform it into a lasting motion with a new path. We also overcome the feet-sliding problem in path editing using constraint-based inverse kinematics method. Our method can reuse the existing data while preserving the physical reality. The experiments demonstrate that our method can be put into a variety of applications, e.g. the character detouring a road-block or approaching through a different route. It can also be applied to other periodical motions, e.g. running.

This paper is organized as follows. First we review some related work in Section 2. Then we introduce some basic knowledge on human skeleton model and regular pattern of human walking in Section 3. In Section 4, we consider each problem of our method in turn. The implementation and the experimental results are presented in Section 5. Finally, we conclude with a discussion of future work in Section 6.

2 Related Work

By far, the common methods for character animation can be mainly classified into three categories: kinematic animation, dynamics animation and animation based on motion capture data [1, 2].

- **Kinematic animation**

Kinematic methods treat the human body as a massless linkage. Kinematic animation includes forward and inverse kinematics. Forward kinematics specifies the state vector of an articulated character over time. This specification is usually done for a small set of key-frames, and the positions between them are usually obtained by interpolation techniques. The choice of appropriate interpolation functions has been well studied by A. Watt and M. Watt [3]. Designing key-frames is the task of an animator and the quality of the resultant motion deeply depends on his/her skills. Physical and biomechanical knowledge is helpful to design high-quality key-frames. The inverse kinematics (IK) is more effective than forward kinematics when the motion are limited by some constraints, such as that the support foot is not allowed to penetrate into the ground, to flow above the earth, or to slide slightly. Girard and Maciejewski [4] proposed a method to generate the motion of joints using IK. In this method, the animator specifies the positions of feet in the world coordinate system, and then calculates the rotation angles from the foot joint to the hip joint using inverse Jacobian matrix. It is also useful to solve the foot-slide problem. Boulic *et al.* [5] first use a standard forward kinematics approach to generate interpolated key-positions. Tolani *et al.* [6] developed a set of inverse kinematics algorithms to calculate the motion of human limbs.

- **Dynamics animation**

Dynamics methods also regard humans as an articulated joint system or a multi-rigid-body system, but each component has a mass. These methods can be classified as forward dynamics based method, the trajectory optimization method and controller based method. They may be used either for adding constraints that guarantee certain realistic details to a predefined motion, or for directly synthesizing the walk. Both of forward dynamics based method and trajectory optimization method are based on the dynamic equations of multi-rigid-body system. In order to avoid the singularity of Euler angles, Nikravesh [7] proposed a matrix form equation of dynamics by expressing the rotations of the rigid body as unit quaternions. Armstrong *et al.* [8] and Wilhelm *et al.* [9] proposed a method using forward dynamics integration to simulate the human behavior, but it is not yet intuitive since it is accomplished through setting the force and the torque indirectly. The trajectory optimization method can be understood as the inverse dynamics. As for the controller based method, the proportional-derivative (PD) controller is used in [10, 11, 12], and a finite state automation is built to describe the whole movement. In each joint, the PD controller generates the torque to drive human moving. Both kinematics and dynamics are often mixed to improve the generation quality. Van Overveld *et al.* [13] used the mixed method to stimulate human walk in curve path: At first, controlling the movement of the leg by kinematic methods, and then stimulating and optimizing movement of the other joints by dynamics methods.

Various kinematics and dynamics methods drive the model in character animation to behave as a human, and the utilization of physical and biomechanical knowledge also enhance reality. However, even the most advance motion model cannot generate motions as realistic as a living person does, needless to say the high computation cost. The development of motion capture techniques makes it feasible to preserve the physical realness.

- **Animation based on motion capture data**

Many researches have been carried out to edit motion capture data to make the character behave beyond the captured motion. Rose *et al.* [14] presented a framework of motion blending based on scattered data interpolation with radial basis functions. For a smooth motion transition, they used the sinusoidal function for smoothing and a verb graph (also called motion map) to describe the process. Park *et al.* [15, 16] proposed an on-line motion blending scheme to control a virtual character in real time from one behavior motion to another. To retarget the new motion to the character, an importance-based approach in [17] is used. But the postures of the other joints are computed algorithmically, not so realistically as recorded in capture data. Semancik *et al.* [18] transformed the individual movements to continuous motion by motion blending, in which the motion of various step lengths can be generated, but the motion orientation is limited. Li *et al.* [19] proposed a method based on functional principal component analysis. In this method they transformed the original motion with high dimension to the low dimension; then adjust parameters to blend new motions and control the character. But the analysis of data is linear, which cannot fully describe the relativity of motion sequences. M. Gleicher [20] provided a method allowing for path-based editing based on existing motion data, but it cannot deal with a sharp turning. It is similar to our method, but our method is easier to implement and fits to a sharp turning.

3 Preliminaries

We will give related knowledge on skeleton model and human walking in this section.

3.1 Skeleton Structure of Human Model

Many manufacturers produce motion capture equipments. They employ similar skeleton models except for slight differences in joints for captured motion information.

Here, we choose the skeleton model as shown in Fig. 1, but our method can be applied to the others.

In this model, 18 joints are selected in all besides the End-Sites. The root, also called hips, has 6 degrees of freedom (DOF), including 3 DOFs for translation and 3 DOFs for rotation, while the others have 3 only for rotation, respectively. For the rotational DOFs, we transform the Euler angle into quaternion form to get rid of the Gimbal Lock problem. Gimbal lock is the loss of one degree of freedom that occurs when the axes of two of the three gimbals are driven into the same place and cannot compensate for rotations around one axis in three dimensional spaces. The End-Sites have no DOFs, but only to present the model more vividly.

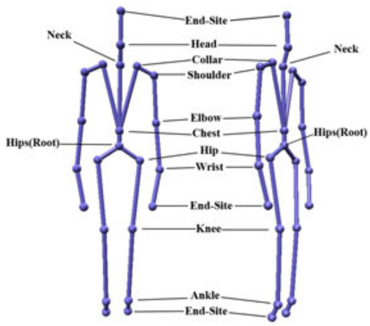


Fig. 1. Skeleton structure of human model

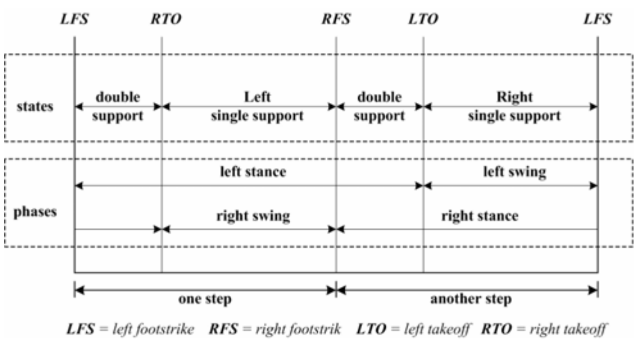


Fig. 2. Characteristic phases of a walking motion [1]

3.2 Circular Motion of Human Walking

Human walking is a kind of regularly circular behavior. An abstract walking model is a succession of separate phases in biomechanics [21, 22]. It characterizes human walking as four phases according to the time when foot-strikes *FS* (the foot contacts the ground) and takeoffs *TO* (the foot leaves the ground). In gait terminology, a stride is defined as a complete cycle from a left foot takeoff to another left foot takeoff, while the part of the cycle between the takeoffs of the two feet is called a step. Four foot-strike and takeoff events occur during a stride: left takeoff (*LTO*), left foot-strike (*LFS*), right takeoff (*RTO*), and right foot-strike (*RFS*). This leads to the characterization of two motion phases for each leg: the period of support is referred as the stance phase and the non-support is referred as the swing phase. As Fig. 2 shows, the continuous shifting between two phases is a walking cycle.

4 Motion Editing

In this section, we will explain in detail how to generate a new walking motion without limits of duration and original path. At first, we transform a segment of one walking cycle into a sequence of cycles in the direction of capture data. Foot-slide is then eliminated by our IK method, which is simple to implement. Next, a new path is designed as required to guide the character's walking. After that, the posture is modified to meet the new path. This step is the key to generate a pleasing motion with arbitrary walking path. We deal with the upper part of the body and the lower part separately. The general flowchart of the whole system is shown in Fig.3.

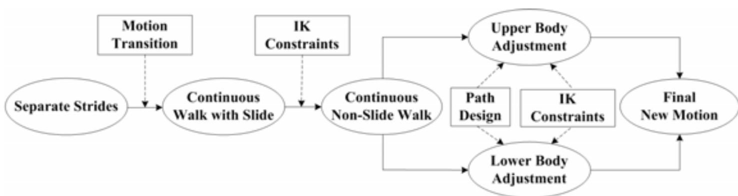


Fig. 3. General flowchart of the system

4.1 Generation of Continuous Walking

As we mentioned in Section 3.2, the continuous walk is formed by a series of separate strides linked end to end. So we regard a stride as a bead and string it repeatedly to produce a necklace. We get a continuous walk map, as shown in Fig. 4.

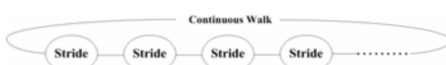
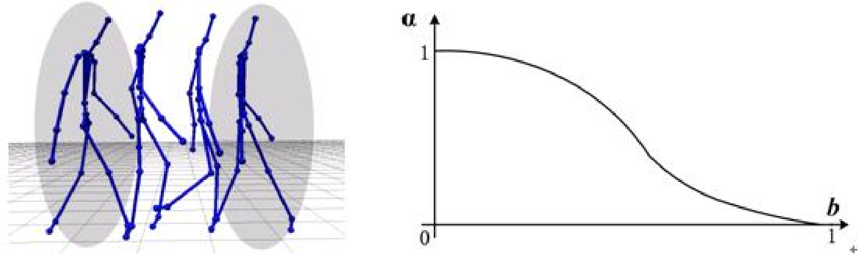


Fig. 4. Continuous walk map

It should be noted that the last posture of a stride is usually similar to the first one, possible with nuances. As shown in Fig. 5(a), four frames are extracted from one stride of walking motion. The two shaded frames are the initial and the last, respectively. Their related postures are quite similar. However, simple links as Fig. 4 usually cause visual shakes at the junction between the last frame of the preceding stride and the first of the succeeding because of such nuances. In order to solve this problem, we use motion transition techniques to smooth the shake. In [15], a motion blending method was proposed to generate a new motion between two different motions, in which the transition is realized by the sinusoidal function [14], as shown in Fig. 5 (b).



Sampled walking of one stride (Left),
Blending function (right)

Fig. 5. Samples of one stride and the blending function

We also use the transition function, but we use different transition period. In other work, the transition begins at a certain key frame in the preceding stride and ends at the first frame of the succeeding. In our work, the transition end at a certain key-frame of succeeding stride instead. This choice makes the transition perform more naturally.

The transition function provides a weight α to mix two source postures to target postures in transition period. Let $T = [t^{*,0}_I, t^{*,1}_I]$ be the period of the transition in Stride I as shown in Fig. 6., where $t^{*,0}_I$ and $t^{*,1}_I$ represent the first and last frame numbers of the period counting from the initial frame of Stride I; b means the beginning part of the current stride to be linked with the preceding stride and e means the ending part of the current stride to be linked with the succeeding one. Usually, the beginning part period is equal to the ending part period.

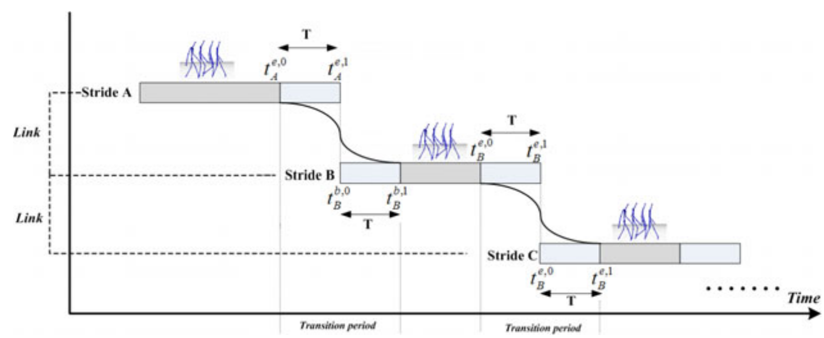


Fig. 6. Linking of strides using the transition function

For the transition between two consecutive Strides A and B, the root joint's position $\mathbf{p}_0(t^i_{A,B})$ of the transition motion at each frame $i \in T$ is computed as:

$$\mathbf{p}_0(t^i_{A,B}) = \alpha \cdot \mathbf{p}_0(t^{e,0}_A) + (1-\alpha) \cdot \mathbf{p}_0(t^{b,1}_B) \quad (1)$$

where $\mathbf{p}_0(t^{e,0}{}_A)$ is the root position of the first frame in the ending part of A and $\mathbf{p}_0(t^{b,1}{}_B)$ is the root position of the last frame in the beginning part of B. The sinusoidal function mentioned previously is presented as:

$$\alpha = 0.5\cos(b\pi) + 0.5 \tag{2}$$

where $b=(f-t_A^{e,0})/(t_B^{e,1}-t_A^{e,0})$, f is the number of the current frame. Equation (1) is suitable for translational DOFs.

For rotational DOFs, the angles are represented with quaternion, so the slurp (spherical linear interpolation) function [23, 24] is employed:

$$\mathbf{q}_n(t^i_{A,B}) = \mathbf{q}_n(t_B^{b,1}) \cdot (\mathbf{q}_n^{-1}(t_B^{b,1}) \cdot \mathbf{q}_n(t_A^{e,0}))^{(1-\alpha)} \tag{3}$$

where $\mathbf{q}_n(t^i{}_{A,B})$ is the nth rotation DOF of the transition motion at frame $i \in T$, $\mathbf{q}_n(t^{e,0}{}_A)$ is the nth rotational DOF at the first frame in the ending part of A and $\mathbf{q}_n(t^{b,1}{}_B)$ is the nth rotational DOF at the last frame in the beginning part of B. The total transition period equals to two times of T. This period should last appropriately, usually less than a step time. Finally, we use the synthesized motion at each frame $i \in [t_A^{e,0}, t_B^{e,1}]$ as the target motions in the transition period.

The transitions between other strides are dealt with in the same way. In our method, it is unnecessary to make time warping and to find the key times [15] before transition.

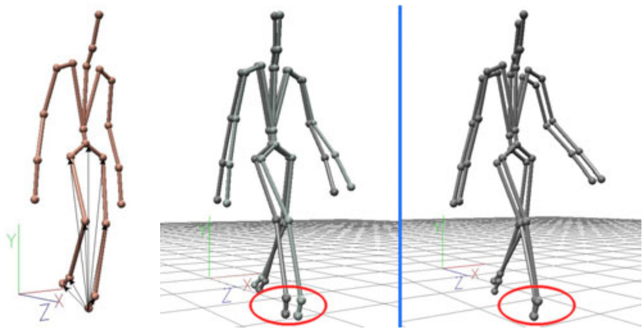
4.2 Elimination of Foot-Slide

In Section 4.1, we have not considered the foot-slide phenomenon in transition yet.

To eliminate the sliding, an IK constraint is subjected to the supporting foot on the ground. IK constraint is commonly used, but it is usually accompanied with complicated algorithms. In this work we propose a new method to do so, which is much easier to implement. We have pointed out in Section 2 that there are two phases of each leg during walking, and for most time, two legs stay in different phases. We define the key-time as the time point when one foot transforms from swing state to stance state. When the key-time begins, the constraint of one foot comes to work to fix the position of this foot on the ground, and hold on until another foot strikes the ground. When the constraint of one foot begins, the constraint of the other ends.

In BVH data from motion capturing, the translational DOFs of the root give the character's position in the world coordinate system. The positions of other joints are determined by the rotational DOFs with their father joints. The position of the supporting foot is recorded when the supporting foot strike the ground before the transition period according the BVH data. Since the position of the supporting foot is fixed on the ground, the positions of other joints can be determined based on this position.

Assume the position of the supporting foot is $\mathbf{p}$ in the world coordinate system. We transform all joints into this system. As shown in Fig. 7 (Left), suppose the left foot is the supporting foot and its position is $\mathbf{p}_{le}$ in a synthesized frame with a foot-slide. The positions of the other skeleton joints are obtained in much the same way according to the names defined in Fig. 1, e.g. the left knee is $\mathbf{p}_{lk}$. The arrows in this figure represent the difference vectors from $\mathbf{p}_{le}$ to the other joints and reflect the positions relative to $\mathbf{p}_{le}$. For example, the difference vector between $\mathbf{p}_{lk}$ and $\mathbf{p}_{le}$ is $\mathbf{v}_{lkle}= \mathbf{p}_{lk} - \mathbf{p}_{le}$. The positions of the other joints can be computed by translating their difference vectors to the position of the supporting foot, e.g. $\mathbf{v}_{lkle} + \mathbf{p}$. Finally, a realistic walking motion without foot-slide is generated. Fig. 7 (Middle and Right) shows the comparison between the walking with and without constraints. In red circle region of Fig. 7 (Middle), the supporting right foot is sliding between two frames. After processing with our method, the right foot is in the fixed position when it works as the supporting foot, as shown in Fig. 7 (Right).



The difference vector (Left), Strides' link without (Middle) and with (Right) constraints

Fig. 7. The difference vector and elimination of foot-slide

4.3 Design of Path

We now propose a scheme to design a new path.

The length of the path is related to the number of frames in motion. A walking curve can be thought of as many a set of line segments connecting head-to-tail. Assuming that the whole motion has N frames and the length of motion path is L, the path is sampled into N-1 segments by N points, each of which has the length of $L/(N$-$1)$. As shown in Fig. 8, the angleθbetween two segments changes the walking direction along a curve path.

A path can be represented as a function to fix the coordinates of the junctions. Here we denote the generic path as c(t) and the path can be defined as:

$$\begin{cases} \mathrm{s}(n) = \mathrm{c}(n \cdot \Delta t) \\ \Delta t = T / N \qquad\qquad n = 0,1,2,\cdots,N-1 \\ L = \int_0^T \left|\mathrm{c}'(t)\right| dt \end{cases} \tag{4}$$

where s(n) is the n-th samples of the path; L is the length of the path; T is the lasting time; N is the number of frames. c(t) can be set as various curves , such as a circle, a sinusoidal curve, or a B-spline curve.

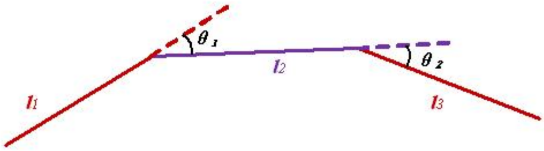


Fig. 8. The straight-line segments and angles between them

4.4 Adjustment of Skeleton for New Path

After the above operations, we are ready to adjust the whole skeleton to walk along the designed route. This a key stage to achieve a good performance.

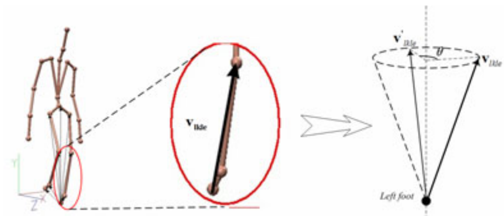


Fig. 9. The rotation of difference vector byθ

The angle θ between two consecutive segments indicates that the character needs to rotate by this angle in order to follow the designed path. In motion capture data, the skeleton rotates around the hips root. However, the intuition tells us that the whole body rotates around the supporting foot rather than the root joint when turning. Another fact we discovered is that, at a turning, people rotates the whole body with stepping forward, not making the rotation during the double support phase (See Figure 2). This discovery is helpful for us to solve the problem.

Although the rotation and forward stepping action proceed synchronously, we deal with them separately. Since the generation of postures at a frame has been completed in foot-slide elimination, the rotation of the skeleton is the last problem to meet the angleθ. The rotation around the supporting foot can be done by rotating other joints, respectively. It is factually the rotation of corresponding difference vectors around the vertical axis crossing the foot. We take $\mathbf{v}_{lkle}$ for example.

As in Fig. 9, the rotation of the left knee around supporting foot is complete by the rotation of $\mathbf{v}_{lkle}$ around the foot. We make $\mathbf{v}_{lkle}$ rotate byθ to the new position $\mathbf{v'}_{lkle}$ as:

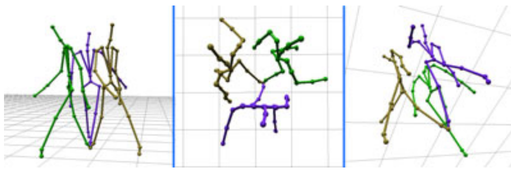
$$\mathbf{v}'_{lkle} = \mathbf{R}_{\theta}\mathbf{v}_{lkle} \tag{5}$$

where $\mathbf{R}_{\theta}$ is the rotation matrix byθ. The treatment can be applied not only to a generic path but also to a sharp turning, which is not considered [15]. But the character needs to rotate by a larger angle at a sharp turning.

Here, we divide the skeleton into the upper part and the lower part at the chest joint. The above operations are performed on joints of the lower half of the body. The upper part can be dealt with in the same manner as the lower half. But a better and easier way is first to translate the chest joint to the skeleton to be generated by using the difference vector between the supporting foot and the chest joint and then to rotate it byθ. Unlike the lower part, the joints of the upper body are all sub-nodes of the chest joint, so the rotation of the upper body can be controlled totally by the rotation of the chest joint. This method simplifies the processing and reserves quite as much the motion details.

5 Experimental Results

Experiments were performed on a Pentium PC (2.4GHz Core2 Quad Processor, 3GB memory).



(a) Side view (b) Top view (c) Bottom view

Fig. 10. Rotation around the supporting foot

Fig. 10 shows the rotation around the fixed supporting foot by 360°. In order to demonstrate the unchanged postures after rotating arbitrary angles, we display the result at three viewpoints: from the side view, from the overlooking view and from the bottom view. Every model rotates by 120°in turn.

Since the rotation of the skeleton by any angle is achieved, walking along arbitrary path can be achievable too. As shown in Fig. 11(a), the original motion is toward the positive z-axis. We choose four different paths for the character to walk: a straight line at 45°from the direction of the positive x-axis in (b), a circle in (c), an ellipse in (d) and a sinusoidal curve in (e). The original walk lasts 75 frames and the resultant walks last 8 times longer. The rate to play is 60fps. All results are satisfying. It should be noted that the path is only a guide direction for human walk in our method. The character doesn't walk exactly on it.

The experiments demonstrate that our method can be used to a variety of applications, such as making the character walk bypass the encountering objects or approach the goal location through different route. Besides human walk, it can also be applied to other periodic motions, e.g. running. Because of low computational cost, the new motion can be generated in real time, which is be beneficial to some real-time applications, such as 3D video games.

Fig. 12 shows the same experiments but for a female character. The original walk has 38 frames and the resultant walks last 8 times longer. The rate to play is 30fps. The performance is also good.

As pointed out in section 4.4, our method is also suitable to the sharp turning. To demonstrate it, we sharpen the turning of the path and put two different characters on this path for experiments.

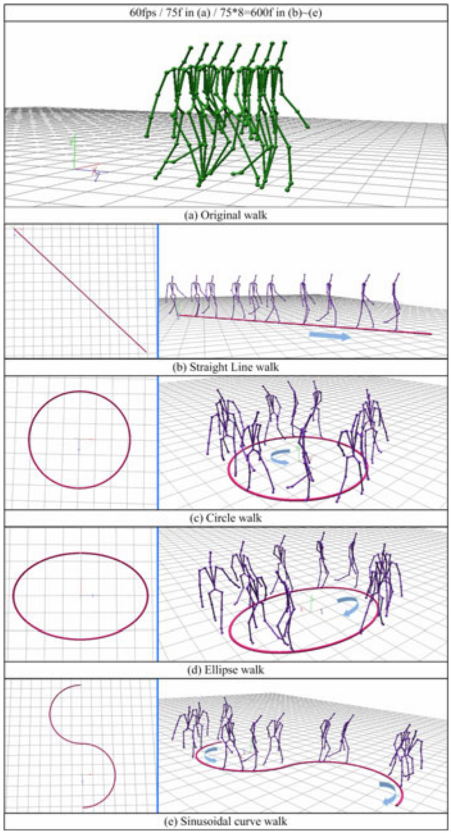


Fig. 11. Various paths of walk

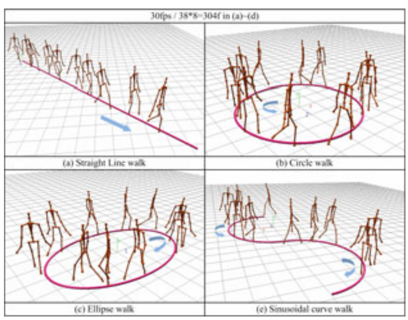
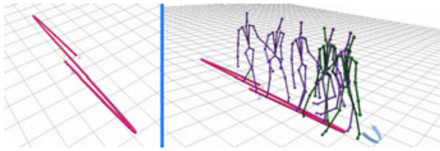
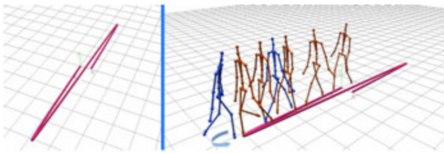


Fig. 12. Various paths of walk for female character



(a)Sharp turning (b) Walk on sharp turning

Fig. 13. The walk on the sharp turning (male character)



(a)Sharp turning (b) Walk on sharp turning

Fig. 14. The walk on the sharp turning (female character)

In Fig. 13, the purple character represents the motion before the turning, while the green represent the after. In Fig. 14, we use the character of a female one with different motion style. The orange character represents the motion before the turning, while the blue represent the after. We can see the performance at the turning is pleasing.

6 Conclusions

Motion capture can record physical information of human motion faithfully. Based on the captured data, highly realistic character animation can be regenerated in contrast to model based methods. In order to enhance the utility of motion libraries, it strongly demands that a single motion segment can be applied in a wide variety of settings. For this purpose, four issues need to be addressed effectively: continuous walking generation, foot-slide elimination, path design and skeleton adjustment for a new path. However, most current work focuses on some parts but not a whole one. In [20], all of these issues are considered, but its framework is just a simple combination of some existing methods, and it could fail for highly kinetic motions, e.g. a sharp turning.

In this paper, we have proposed an integrated framework for the purpose. Our method employs effective techniques for automatic generation of free walking in real-time with optional lasting duration and motion path, based on captured motion data. Our method is purely geometric and is easy to implement, in which there are no complicated mathematical algorithm and notion of physics. Furthermore, it can fit to a highly kinetic motion, e.g. a sharp turning. The experiments demonstrate the effects of our feasible scheme. It can be put into a variety of real-time applications. Besides human walk, our method can also be applied to other circular motions, e.g. running.

Our framework still has much room to be improved. For example, the resultant walk by our current method does not follow the desired path perfectly. We plan to address them to make it more robust and flexible in the future work.

Acknowledgements

The authors would like to thank Jaroslav Semancik for the software platform code and BVH data on his homepage. We would also thank Haiming Wang for the discussion with him during the work. This work is supported by National Natural Science Foundation of China Projects No. 60970093, No. 60872120, and No. 60902078; by National High-Tech Research and Development Plan, 863, of China under Grant No. 2008AA01Z301; by the Strategic Cooperation between Project Guangdong Province and Chinese Academy of Sciences No. 2009B091300044, and by Industry-University-Research Project of Department of Education in Guangdong Province No. 2008B090500207.

References

1. Multon, M., France, L., Cani-Gascuel, M.-P., Debunne, D.: Computer animation of human walking: a survey. Journal of Visualization and Computer Animation 10(1), 39–54 (1999)
2. Xia, S., Wang, Z.: Recent advances on virtual human synthesis. Science in China Series F: Information Sciences 52(5), 741–757 (2009)
3. Watt, A., Watt, M.: Advanced Animation and Rendering Techniques. Theory and Practice (1992)
4. Girard, M., Maciejewski, A.A.: Computational Modeling for the Computer Animation of Legged Figured. Computer Graphics 19(3), 263–270 (1985)
5. Boulic, N., Thalmann, M., Thalmann, D.: A Global Human Walking Model with Real-time Kinematic Personification. The Visual computer 6(6), 344–358 (1990)
6. Tolani, D., Goswami, A., Norman, I.: Real-Time Inverse Kinematics Techniques for Anthropomorphic Limbs. Graphical Models 62(5), 353–388 (2000)
7. Nikravesh, P.: Computer-aided Analysis of Mechanical Systems, pp. 1–370. Prentice Hall, EngleWood Cliffs (1988)
8. Armstrong, W.W., Green, M.: The dynamics of articulated rigid bodies for the purposes of animation. In: Proc. Graphics Interface 1985, pp. 407–415 (1985)
9. Wilhelms, J.: Using dynamic analysis for realistic animation of articulated bodies. IEEE Computer Graphics and Applications 7(6), 12–27 (1987)
10. Yin, K., Loken, K., van de Panne, M.: SIMBICON: Simple biped locomotion control. ACM Transactions on Graphics (Proc. SIGGRAPH) 26(3), article 105 (2007)
11. Wang, M., Fleet, J., Hertzmann, A.: Optimizing walking controllers. ACM Transactions on Graphics (ACM SIGGRAPH Asia) (2009)
12. Coros, S., Beaudoin, P., van de Panne, M.: Robust task-based control policies for physics-based characters. ACM Transactions on Graphics (SIGGRAPH Asia) (2009)
13. Van Overveld, C.W.A.M., Hyeongseok, K.: Small steps for mankind: towards a cinematically driven dynamic simulation of curved path walking. The Journal of Visualization and Computer Animation 5(3), 143–165 (1994)
14. Rose, C., Bodenheimer, B., Cohen, M.F.: Verbs and Adverbs: Multidimensional Motion Interpolation. IEEE Computer Graphics and Applications 18(5), 32–40 (1998)
15. Park, S.I., Shin, H.J., Shin, S.Y.: On-line locomotion generation based on motion blending. In: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation (2002)

16. Park, S.I., Shin, H.J., Kim, T.H., Shin, S.Y.: On-line motion blending for real-time locomotion generation. Research Articles, Computer Animation and Virtual Worlds 15(3-4), 125–138 (2004)
17. Shin, H.J., Lee, J., Gleicher, M., Shin, S.Y.: Computer puppetry: an importance-based approach. ACM Transactions on Graphics 20(2), 67–94 (2001)
18. Semancik, J., Pelikan, J., Zara, J.: Interactive Synthesis of Constrained Motion from Example Movements. In: Visualization, Imaging, and Image Processing: Fourth IASTED International Conference Proceedings (2004)
19. Li, C., Wang, Z., Xia, S.: Motion synthesis for virtual human using functional data analysis. Journal of Software 20(6), 1664–1672 (2009)
20. Gleicher, M.: Motion path editing. In: Proceedings of the 2001 symposium on Interactive 3D graphics, pp. 195–202 (2001)
21. Enoka, R.M.: Neuromechanical Basis of Kinesiology, Edition II. Human Kinetics (1994)
22. Nilsson, J., Thorstensson, A., Halbertsam, J.: Changes in leg movements and muscle activity with speed of locomotion and mode of progression in humans. Acta Physiol. Scand., 457–475 (1985)
23. Kim, M.J., Kim, M.S., Shin, S.Y.: A general construction scheme for unit quaternion curves with simple high order derivatives. In: Proceedings of SIGGRAPH 1995, pp. 369–376 (1995)
24. Shoemake, K.: Animating rotations using quaternion curves. In: Proceedings of SIGGRAPH 1985, pp. 245–254 (1985)