Following Path using Motion Parameters for Virtual Characters

Seongmin Baek, Il-Kwon Jeong, In-ho Lee*

* Virtual Reality Research & Development Department,
Electronics & Telecommunications Research Institute (ETRI), Daejeon, Korea
(Tel: +82-42-860-5369; E-mail: {baeksm, jik, leeinho}@etri.re.kr)

Abstract: This paper presents a new method that generates a path that has no collision with the obstacles or the characters by using the three motion parameters, and automatically creates natural motions of characters that are confined to the path.

Our method consists of three parameters: the joint information parameter, the behavior information parameter, and the environment information parameter. The joint information parameters are extracted from the joint angle data of the character and this information is used when creating a path following motion by finding the relation-function of the parameters on each joint. A user can set the behavior information parameter such as velocity, status, and preference and this information is used for creating different paths, motions, and collision avoidance patterns. A user can create the virtual environment such as road and obstacle, also. The environment is stored as environment information parameters to be used later in generating a path without collision. The path is generated using Hermit-curve and each control point is set at important places.

Keywords: path generation, digital actor, digital extras, motion parameters

1. INTRODUCTION

These days a human-like character is widely used in the multimedia contents like game and film. The need for complex scenes including hundreds of characters is increasing gradually. A large number of characters, so called the digital extras, are not playing an important role in the context of the story of the contents but contribute to enhancing realism of the contents. However, it is labor intensive and time-consuming for an animator to control each of the characters by using a traditional animation method such as key framing though it has the advantage of allowing the animator to control precisely over the motion of the character.

Suppose the following situation, for example. An animator must generate hundreds of paths that hundreds of characters follow. Some of characters may move fast, and others may move slowly. The faster may overtake the slower. The characters, which are walking along the path, have to avoid the other characters as well as the unexpected obstacles as a normal human does. In addition, the motions of the characters should be apparently different in order to avoid clumsiness of a scene.

As seen in this example, it is almost impossible to create a scene by key framing. Although a certain motion is given to all the characters to follow, we need to modify the given motion for a realistic and natural movement.

In this paper, we present a new method that generates a path that has no collision with the obstacles by using the environment information, and automatically creates natural motions of characters that are confined to the path.

Our method consists of two layers: The low layer controls the motion in the joint level by extracting the information of the joint and the high layer controls the overall motion by using the environment information and the behavior information of a character. We propose a collision avoidance method based on human society rule because the generated motions must not have collisions.

The structure of the paper is as follows. In section 2, we describe previous work on motion generation technique and section 3 explains our two layers that control a digital actor. We introduce our path and motion generation method by motion parameter in section 4, and section 5 contains collision avoidance rule. Conclusion is presented in section 6.

2. PREVIOUS WORK

The techniques for controlling a character are growing increasingly popular. Many systems have been suggested for controlling a virtual human but crowd systems not yet. F. Multion [7] surveyed the set of techniques developed in animating human walking. In his survey, he focused on the evolution from inverse kinematics (IK) to dynamics and review motion editing technique based on motion capture data. Tolani et al. [6] solved the IK problem of a human arm and leg by analytic solution. A numerical method of IK relies on an iterative process to obtain a solution and Rose et al. [4] solved non-linear optimization problem with various constraints. Lee et al. [10] presented a technique that adapts a motion of a character by using a hierarchical curve fitting and inverse kinematics solver. Bruderlin et al. [1] presented a hybrid approach to animate the human locomotion, which combines goal-directed and dynamic motion control. Tak et al. [15] proposed motion balance filtering, which corrects an unbalanced motion to a balanced one while preserving the original characteristics as much as possible. The above techniques do not consider many characters, however.

Ulicny [3] developed the crowd simulation for interactive virtual environments such as VR training system for urban emergency situations. Farenc [12] proposed a new architecture for simulating virtual humans in complex environment and Kallmann [11] designed 'smart object' based on a complete definition and representation of interactive objects. Sulivan et al. [5] presented crowd and group simulation with level of detail (LOD). These techniques, however, focused on mainly behaviours in specific environment.

Recently, research efforts have developed a probabilistic model for motion synthesis based on the motion capture data [8][9][13][14]. Lee et al. [8] showed that a connected set of a human-like character can be generated from non-linear sequences of motion, automatically organized for search, and used real-time control of an avatar using three interface techniques: selecting from a set of available choices, sketching a path, and acting out a desired motion in front of a camera. Kovar et al. [9] constructed a directed graph called a 'motion graph' that includes connections among the database for creating realistic, controllable motion. The motion graph consists of original motions and automatically generated transitions. Li et al. [13] described 'motion texture' for synthesizing complex human character that is statistically similar to the original motion captured data. Motion texture can be manipulated by modifying the details of a specific motion at the texton level or by generating a new motion at the distribution level. Pullen et al. [14] discussed a method for creating animations by setting a small number of keyframes and used motion capture data to enhance the animation.

3. PARAMETERS

Our system consists of two layers: the low layer and the high layer. The low layer has one parameter–joint information parameter and the high layer has two parameters–behavior information parameter and environment information parameter.

3.1 Joint Information Parameters (JIP)

In the low layer, motion capture data is loaded and then joint information parameters (JIP) are extracted from the joint angle data of the character. This information is used when creating a path following motion.

A joint of a human has one, two or three degrees of freedom (DOF); we set the appropriate parameters to each joint. For example, a joint that has three DOF such as the thigh has three parameters (y-axis rotation angle, direction, angular velocity), a joint with two DOF has two parameters (y-axis rotation angle, angular velocity), and a joint with one DOF has angular velocity only (figure 1).

In order to create a turning right by about 45 degrees after walking forward motion from the two individual motions, a walking forward motion and a turning motion to the right by about 90 degrees, for example, the low layer extracts JIP from the two motions and searches the relation-function of the parameters on each joint. This function adapts the original motion so that it is confined to the path. The relationship between two motions can be presented by quaternion interpolation of angular velocity. Therefore, motion transitions are natural and realistic.

However, suppose the scene that the motions of all the characters are same! Such situation can be needed for soldiers to parade the road but it is not general. Therefore, we adopt 'motion multi-resolution filtering' [2] for adaptation of existing motion. In addition, start frames of characters have to be differently selected because identical initial poses are awkward, also.

In this way, the low layer is able to generate various motions with a few motions.

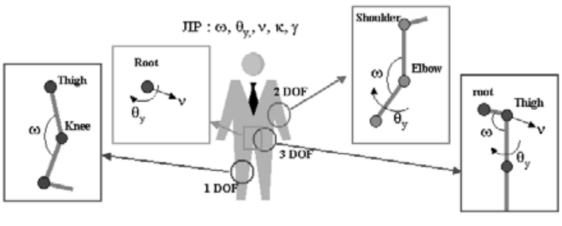


Figure 1. Joint Information Parameters (JIP)
(ω: angular velocity, θ_y: y-axis rotation angle, ν: direction vector, κ: value for multi-resolution filtering, γ: start frame)

3.2. Behaviour Information Parameters (BIP) and Environment Information Parameters (EIP)

A user cannot control a character directly in the low layer because the resulting parameters extracted from motion-captured data are complex. The high layer may be a kind of interface for controlling characters and consists of two parameters.

A user is able to create the virtual environment such as road and obstacle, also. The environment is stored as environment information parameters (EIP) to be used later in generating a path without a collision. EIP includes the positions and the sizes of obstacles and the restricted area information, etc.

A user is able to set the behavior information parameter (BIP) such as velocity, weight, status, and preference. The velocity parameter controls the moving velocity of a character (e.g. slow or fast) and the force vector is calculated by the weight parameter. If this force becomes larger, a radius of rotation of a character becomes larger. The status parameter determines whether the motion is large or small and is connected with multi-resolution filter (κ) within JIP. The preference parameter is used for making decisions. For example, when applying a collision avoidance algorithm, whether the character turns to the left or the right, whether the character avoids obstacles far or near, or whether the character waits for the moving obstacle to go pass or not is determined by using the preference parameter. Figure 2 shows that the created path is also different because of different weight, margin, and velocity.

In this way, an animator can easily creates a realistic scene with moving crowd because the different paths, motions, and collision avoidance patterns are applied using these behavior parameters.

4. PATH GENERATION

A motion path plays an important role in controlling all the motions of characters. A path is made of Hermit-curve with a number of control points. The curve avoids a collision with obstacle by placing a new control point at appropriate position (between two control points) when it passes through any obstacle during the creation of it. Figure 3 shows the method that generates a path. First, the system checks a collision between start position and end position. If not, the path is the line that connects two points: otherwise, it inserts the control points (fig 3(b)(c)). When inserting a control point, the system refers BIP as well as EIP as mentioned before.

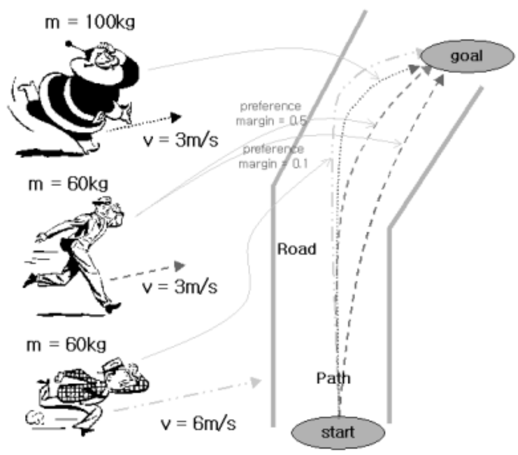


Figure 2. Example of the behavior information parameter

Suppose a motion path proceeds to a goal point. When the system estimates that the path would result in a collision with obstacle by EIP, it determines where to place the control points (whether left or right and whether far or near) by BIP and EIP.

With the placed control points, tangent vector is determined at each control point.

$$\mathbf{V}_t = (\text{norm}(\mathbf{C}_{i+1} - \mathbf{C}_{i-1}) + \text{norm}(\mathbf{C}_i - \mathbf{C}_{i-1})) \times \nu \times \omega$$

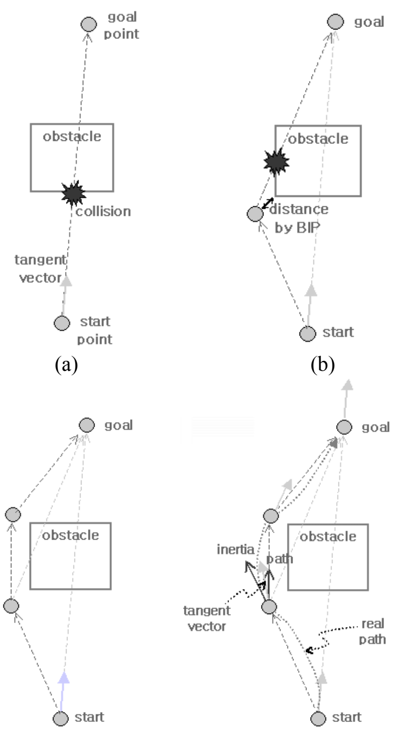
($\mathbf{V}t$: tangent vector, $\mathbf{C}i$: i-th control point,
ν : velcocity, ω : weight of character)

Figure 2 shows relation between a path and BIP (velocity and weight). If character is fat or character is fast then a radius of rotation of a character becomes larger.

A character goes forward based on the path and its motion can be modified by BIP such as velocity and status while walking. For example, if the velocity is large then the motion becomes running. The direction vector is generated in accordance with the tangential vector of the curve and the tangential vector using parameter function automatically tunes the angle of each joint.

As mentioned before, the motion path controls overall motion of a character and does not deal with collisions among characters.

.



(a) (b)

(c) (d)

Figure 3. Path generation

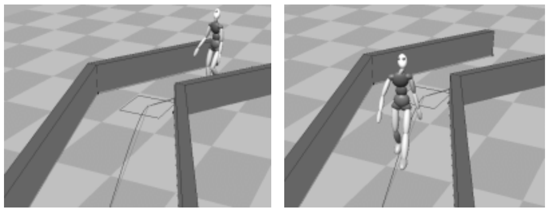
5. MOTION

Figure 4 shows the characters following the paths. A character walks along the road as shown fig 4-(a). Figure 4-(b) presents that the characters moves while avoiding obstacles on the ground. Some characters run and others walk by input velocity. A user can alter the velocity of a character of one's own accord and the character runs after walking motion or walks after running in accordance with the speed. The number of motion frame and a moving distance of a character are determined by a given curve trajectory and the style of the motion is changed by interpolation using quaternion.

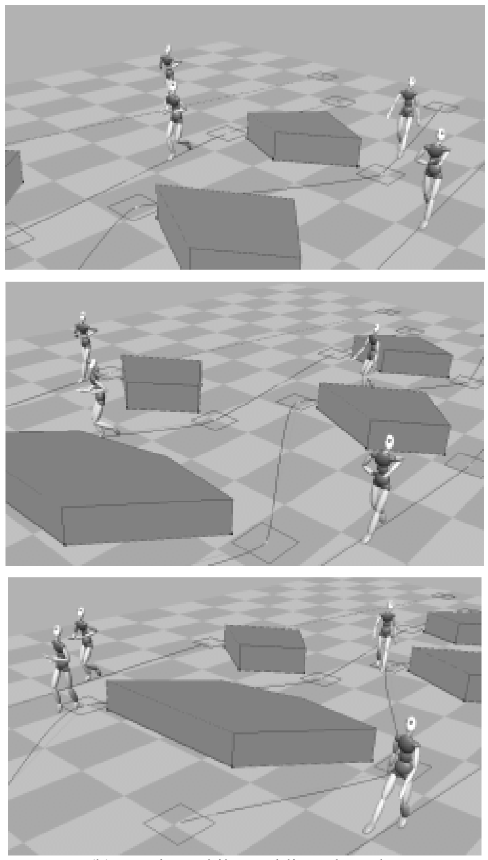
6. CONCLUSION

We have presented a new approach for generating an appropriate path and motion automatically by using the three motion parameters: the joint information parameter, the behavior information parameter and the environment information parameter. We have showed natural collision avoidance method that applies rules as humans do. JIP controls a character in joint level. BIP and EIP relate to collision avoidance and overall motion. This method has two merits; the motion of a character is similar to that of a human because the motion is based on the motion captured data and our system is fast and easy because the motion is controlled by high-level parameters. This system can be expanded to the crowd system because it is able to create many analogous motions with a few motions and is able to apply to lots of characters and avoids a collision naturally based on human rules.

However, it cannot handle special motions such as an exercising on the horizontal bar because the path cannot be generated. We did not consider constraints on creation of motion. A constraint may state that a certain point must be passed or a certain motion must be played. We expect to solve these constraints in the future.



(a) Walking along the road



(b) Moving while avoiding obstacles

Figure 4. Following path

REFERENCES

[1] A. Bruderlin and Calvert, T. W., "Goal-directed, dynamic animation of human walking", In Computer Graphics (Proceedings of SIGGRAPH 89), vol. 23, pp. 233-242.

[2] A. Bruderlin and L. Williams, "Motion Signal Processing", In Proceeding of SIGGRAPGH 95, 1995, pp. 97-104

[3] B. Ulicny and D. Thalmann., "Crowd simulation for interactive virtual environments and VR training systems", Proc. Eurographics Workshop on Animation and Simulation '01, 2001, pp. 163-170

[4] C. Rose, B. Guenter, B. Bodenheimer, and M. F. Cohen., "Efficient generation of motion transitions using spacetime constraints", In Proceeding of SIGGRAPH 96, 1996, pp. 147-154.

[5] C. Sulivan, J Cassell, H Vilhjalmsson, S. Dobbyn, C. peters, W. Leeson, T. Giang and J. Dingliana, "Crowd and Group Simulation with Levels of Detail for Geometry, Motion and Behaviour", *Third Irish Workshop on ComputerGraphics*. 2002.

[6] D. Tolani and N.I. Badler., "Real-time inverse kinematics of the human arm", Presence, 5(4), 1996, pp. 393-401

[7] F. Multon, L. France, C. Gascuel and G. Debunne., "Computer Animation of Human Walking: a Survey", The Journal of Visualization and Computer Animation volume 10, 1999, pp 39-54

[8] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard, "Interactive control of avatars animated with human motion data", *In Proceedings of ACM SIGGRAPH*, 2002, pp. 491-500

[9] L. Kovar, M. Gleicher, and F. Pighin, "Motion Graph", In Proceeding of SIGGRAPH 02, 2002, pp. 473-482

[10] J. Lee and S. Y. Shin., "A hierarchical approach to interactive motion editing for human-like figures", In Proceeding of SIGGRAPH 99, 1999, pp. 39-48.

[11] M. Kallmann and D. Thalmann, "Modeling Behaviors of Interactive Objects for Real Time Virtual Environments", Journal of Visual Languages and Computing 13(2), 2002, pp. 177-195

[12] N. Farenc, S. R. Musse, E. Schweiss, M. Kallmann, O. Anue, R. Boulic and D. Thalmann., "A Paradigm for Controlling Virtual Humans in Urban Environment Simulations", Applied Artificial Intelligence 14(1), 2000, pp. 69-91

[13] Li, Y., Wang, T., and Shum, H–Y, "Motion texture: A two-level statistical model for character synthesis", In Proceedings of SIGGRAPH 02, 2002, pp. 465-472

[14] Pullen, K. and Bregler, C., "Motion Capture assisted animation: Texturing and synthesis", In Proceedings of SIGGRAPH 02, 2002, pp. 501-508

[15] S. Tak, O. Song, and H. Ko, "Motion Balancing Filtering", In Eurographics 00, 2000

[16] Y. Kang, H. Cho and E. Lee, "An Efficient Control over Human Running Animation with Extension of Planar Hopper Model", The Journal of Visualization and Computer Animation, 10(4), 1999, pp. 215-224