Automatic Generation of Character Behavior by the Placement of Objects with Motion Data

Kenichi Kobori
Osaka Institute of Technology
1-79-1 Kitayama
573-0196 Hirakata, Osaka, Japan
kobori@is.oit.ac.jp

Kenji Hirose
Mitsubishi Electric Corporation
2-3-33 Miwa
669-1513 Sanda, Hyogo, Japan

Koji Nishio
Osaka Institute of Technology
1-79-1 Kitayama
573-0196 Hirakata, Osaka, Japan
nishio@is.oit.ac.jp

ABSTRACT

Recently, virtual space design with high quality three-dimensional CG has become possible due to a rapid improvement in computer performance. In order to give a CG character the movement of its own, it is necessary to apply the motion data to the character after getting them with a motion capture device or by hand work. Production costs have increased because this is complicated work for an animator. In general, the character movement depends on the objects which it holds. Our idea is that the character does not have motion data of its own, but we give the character motion to the objects that it holds. We have developed a method for automatically generating the character motion data by giving motion information to the objects. Thus, each object includes the motion data that cause the character to act. In addition, we give multiple motion data appropriate for the situations to the objects and define the relationship between objects in a virtual space. By using this relation, we can generate a variety of motion data according to the object which the character holds. The proposed method reduces the number of character motion data which should be prepared beforehand.

Keywords

Motion data, animation, movement, virtual reality

1. INTRODUCTION

Recently, virtual environment design with high quality three-dimensional CG has become possible by a rapid development of the computing power. CG characters have existed in three-dimensional virtual space to realize the space where looks just like reality. In general, we make use of motion capture data to let the CG characters act [Alb00]. We have to prepare many scenes to construct various kinds of virtual space. In addition, it is necessary to prepare many motion data of CG characters. It forces animators to enormous work. Whenever the scene is changed, they have to prepare many motion data. It is necessary that they acquire motion data again or revises it. Therefore, a method to automatically generate motion data has been required and several methods have been proposed [Kan06] [Mar99].

However, these approaches have the restriction that only limited motion data can be generated or one CG character maintains only one motion data. Kan et al. showed that automatic generation can be achieved by using building blocks, called motion patches with motion data. Each patches is annotated with simple movements such as "walk " or "stand up " .
However, we need to prepare a lot of motion patches when we make complicated motion data.
We propose a method to solve these problems. In our method, the character does not have the motion data of its own. Instead, the objects that the character operates preserve movement information of the character. We can give several movement data to the object when it is necessary. In addition, the movement to the object varies depending on the movement to other object by giving the relation between two objects. The data that the object maintains include motion information of the character and start position to be carried out for the object and its geometry data. This kind of approach is known in the field of robotics, but has not been used generally in character animation .

2. PROPOSED METHOD

The object has three kinds of information to give the character some sort of action. We generate entire motion data of the character by synthesizing the following information and walking motion data.

2.1 Information that object maintains

The information includes display information, movement information and relation information.

1) Display information

This is shape geometry data to display an object in three-dimensional virtual space. We can change the appearance of the object easily by changing these data.

2) Movement information

This data is the information to generate the movement of the character. The information includes the movement to be applied to the character, the start position of the movement and occupation area of the object in the virtual space.
Put another way, the movement information is the motion data to be carried out for the object.
For instance, a chair preserves the movement of "sit" and a ball preserves the movement of "throw".
This movement information is basic data that we acquire with a motion capture device in advance.
The character recognizes the movement that it should perform without distinguishing objects because each object has the movement for the character.
The start position of the movement is the position where the character starts the movement for the object. The occupation area is a convex polygon surrounding the object.
When the character advances toward a target object, an avoidance path is generated utilizing the convex polygon so that the character does not collide to other objects. We describe the method in detail in section 2.2.

3) Relation information

The character sometimes operates several objects, as follows:

a) Hammer + Nail →Drive a nail with a hammer
b) Postcard + Mailbox→Drop a postcard into a mailbox
c) Ball + Locker →Put a ball into a locker

In these cases, the character operates an object holding with other object.
For example, the character will do the action that it "reads postcard" when it holds the postcard. However, it is expected that the character drops the postcard into a mailbox when it holds the postcard in front of the mailbox.
In order to achieve above-mentioned mechanism, we need to define the relation between the postcard and the mailbox and add these two objects the information that the character drops the postcard into the mailbox. The object holds three kinds of identifiers, as shown in Figure 1.

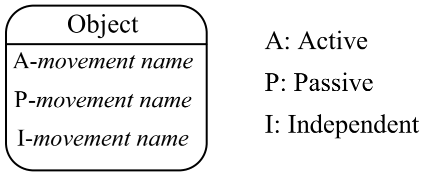


Figure 1. Relation information

Identifier A (Active) means the movement that the character performs. Identifier P (Passive) means the movement that the character performs in the state of holding an object with identifier A.
Identifier I (Independent) means the movement that the character performs independently.
Figure 2 shows the relations among three objects, a mailbox, a postcard and a trash bin.

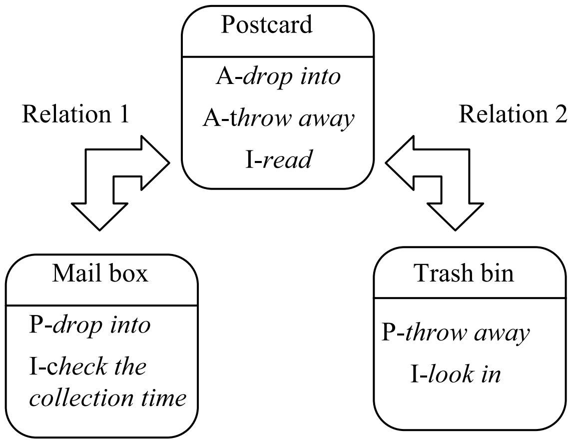


Figure 2. Relations among three objects

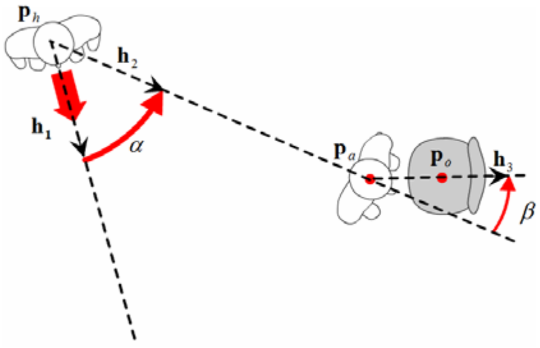
The postcard and the mailbox have the relation 1 each other and the trash bin and the postcard have the relation 2 each other. The character can drop the postcard into the mailbox when it holds the postcard by relation 1. On the other hand, the character does not drop into the mailbox without the postcard, but checks the collection time of mails. Furthermore, the character can throw the postcard away into the trash bin when it holds the postcard by relation 2. If the character does not hold the postcard, it will look in the trash bin.

2.2 Motion data generation

The motion data of the character is determined by using the movement information and the relation information. The final motion data are generated by synthesizing the movement to be applied and walking movement.

1) Outline of generation of movement

Figure 3 shows the situation that a character and a chair are placed in the virtual space.



P_h: Start position of a character

P_a: Position of the movement to be applied

P_0: Position where an object is put on

h_1: Initial direction vector of a character

h_2: Vector from P_h to P_a

h_3: Vector from P_a to P_0

α: Angle between h_1 and h_2

β: Angle between h_2 and h_3

Figure 3. Motion data generation

First, the character turns α degrees to the direction h_2 and walking motion is allocated for the character. When the character arrives at the position P_a, it turns β degrees.

Next, the avoidance path is generated by using walking motion in the case that obstacles exist in the direction which the character is moving in.

The movement which the chair has is applied to the character. As a result, the motion data are generated from walking motion to the motion which the character performs for the object. In the case that plural objects are placed in the virtual space, the process mentioned above is repeated until the character completes the movement that all objects have.

2) Generation of avoidance pass

When the character goes straight on toward a target object, there is the case that it collides with an object which blocks the character's way, as shown in Figure 4. We generate the avoidance pass by using the occupation area of the object, as shown in Figure 5.

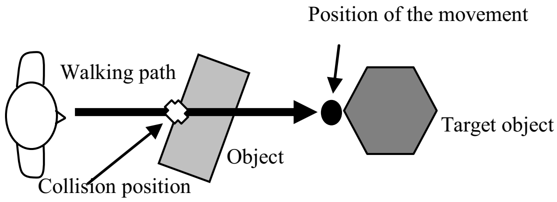


Figure 4. Collision of object with character

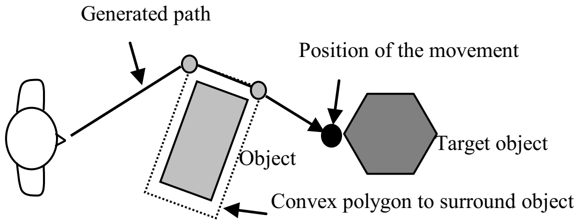


Figure 5. Avoidance path generation

However, the movement of the character becomes unnatural at those vertices of the convex polygon to surround the object because the generated pass is a set of lines through those vertices. Therefore, we interpolate the generated pass with a curve, as shown in Figure 6. Ps is a start point and Pe is an end point of the path. P1 and P2 are the vertices of the convex polygon. First, three points Ps1, P12 and P2e are inserted in the middle point of these three lines. Next, two quadratic Bezier curves are generated by using Ps1, P1, P12 and P12, P2, P2e. The final path consists of the line Ps-Ps1, the quadratic Bezier curve Ps1-P12, the quadratic Bezier curve P12-P2e and the line P2e-Pe.

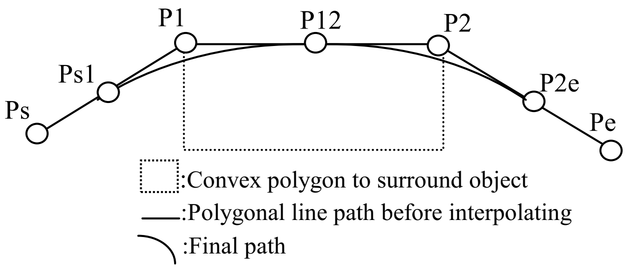


Figure 6. Curve interpolation of avoidance path

3) Generation of the movement to walk

The movement to be applied starts at the position of the movement in the state that the character stands upright. When the character arrives at the position of the movement to be applied, the posture must be standing straight. In order to meet this requirement, we prepare seven kinds of walk movement for the half period in Figure 7 in order to join the walk movement and the movement to be applied smoothly.

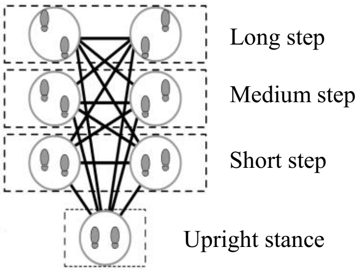
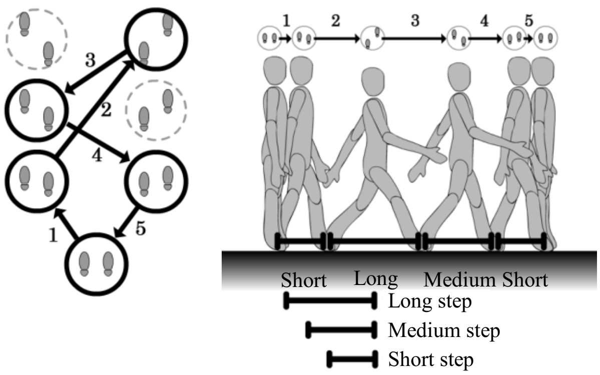


Figure 7. Walk movement for half period

We choose several walk movements among seven kinds of walk movement and connect those movements so that the final posture becomes standing straight. For instance, the walk movement is generated in Figure 8(b) when five walk movements are connected in numerical order, as shown in Figure 8(a).

4) Motion transition

In general, the posture becomes unnatural at the potion where we just connect motion data A to B, as shown in Figure 9.



(a) An example of connection (b) Generated walk movement

Figure 8. An example of walk movement generation

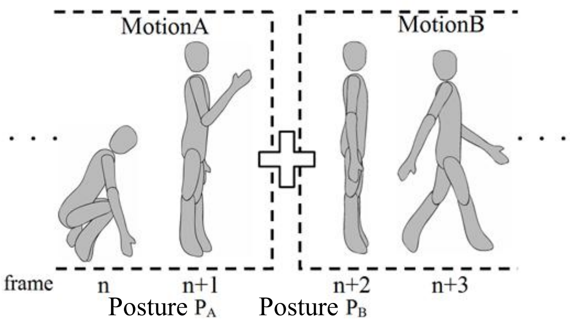


Figure 9. Unnatural motion to be connected

Therefore, we insert transition frames so that the posture P_A gradually resembles posture P_B, as shown in Figure 10. The motion data is modified so that the posture changes from motion A to B naturally by inserting intermediate motion data between those two motions. The posture of the character is expressed by Euler angle of each joint. Interpolation between two postures becomes unnatural if we use the interpolation of Euler angle [Jam90]. We use spherical linear interpolation to avoid Gimbal Lock problem. Two Euler angles to be interpolated are transformed to two quaternions and those are interpolated by using spherical linear interpolation [Tom06].

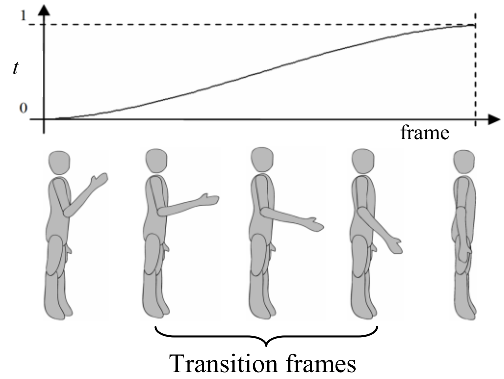


Figure 10. Interpolated postures

3. EXPERIMENTS AND RESULTS

1) Placement of two objects with the relation, as shown in Figure 11.

Table 1 shows two objects and the movement to be applied.

No.	Objects	Movement to be applied
1	Hammer	Pick up
2	Wood	Drive nail

Table 1. Allocated objects and the movements

First, we put a character and two objects which have the relation data in a virtual space.

Figure 12 shows the result of the movement. The number under each figure shows the frame number.

The wood has no movements for the character directly after placement of the character and the objects, as shown in Figure 13(a). After we select the movement “pick up” in Figure 13(b), the movement “drive nail” can be selected, as shown in Figure 13(c) because the hammer and the wood have the relation each other.

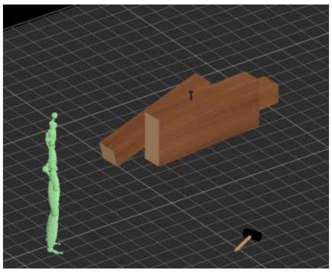
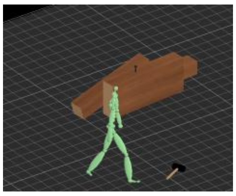
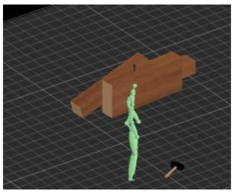


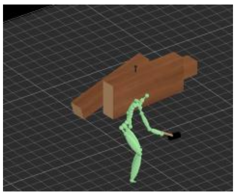
Figure 11. Placement of two objects and a character



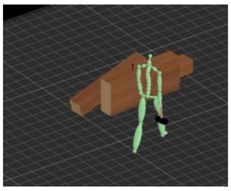
(a) frame #30



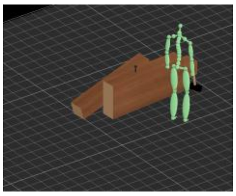
(b)frame #64



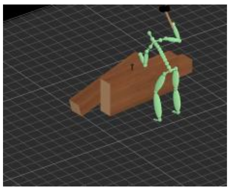
(c)frame #136



(d) frame #234



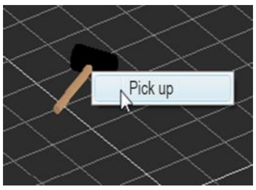
(e)frame#272



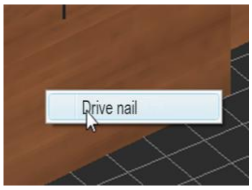
(f)frame#334

Figure 12. Result of motion data generation to two objects with relation

(a) Before picking hammer



(b) Selection of "Pick up"



(c) Appearance of the movement "Drive nail"

Figure 13. Selection of the movement to be applied

2) Placement of five objects (illuminator, television, post card, door and mailbox) and a character in a virtual space in Figure 14.

Figure 14. Three dimensional virtual space

Table 2 shows the movement to be applied to five objects. First, we set the character and five objects in the virtual space. Next, we select the movement to be applied to five objects in the movement order of the character.

No.	Objects	Movement to be applied
1	Illuminator	Turn on
2	Television	Switch on
3	Post card	Pick up
4	Door	Open and pass through
5	Mailbox	Drop into

Table 2. Allocated objects and movements

Figure 15 shows the result of the movement. The result shows that the motion data of the character is generated by the order in Table 2.

(a) frame #67 (b) frame #68

(c) frame #342 (d) frame #524

(e) frame #810 (f) frame #1030

Figure 15. Result of motion data generation

The last movement of "Drop into mailbox" is generated by using the relation between the post card and the mailbox. The character doesn't drop the card into the mailbox in case that it does not hold the post card. The relation information enables generation of variety of motions. When the television is replaced by an electric fan, as shown in Figure 16, we can also generate another objective movement by only replacing those objects. The movement of the character in Figure 16(b) is different from the movement in Figure 16(a) obviously.

(a)Placement of a television (b) Placement of a fan

Figure 16. Scene modification

Even if the modification of the scene results from the change of objects, it is unnecessary to capture motion data afresh. This method produces reduction of motion generation cost.

4. CONCLUSION

In this paper, we proposed an automatic generation method of character behavior by only objects placement. We achieved this motion generation by giving the movement of the character to objects. The relation information between the objects enables generation of various motions depending on the object that the character operates. Experimental results showed that our method is effective.

There are some directions for future works. For instance, unnatural walking motion may be generated when the character changes a direction to walk. This problem is caused by using only seven kinds of walking movement, as shown in Figure 7. In order to solve this problem, we could also improve walking motion by introducing Motion Graphs [Kov02] that is widely used to generate natural walking motion. In addition, another future work will focus on adding the function that a character acts in corporation with other characters when several characters exist in three dimensional virtual space.

5. REFERENCES

[Alb00] Alberto Menache:"Understanding Motion Capture for Computer Animation and Video Games", MORGAN KAUFMANN PUBLISHERS, pp.14-36, pp.121-142,2000.

[Kan06] Kang Hoon Lee, Myung Geol Choi, Jehee Lee:"Motion Patches: Building Blocks for Virtual Environments Annotated with Motion Data", ACM Transactions on Graphics, Vol.25 No.3, pp.898-906 ,2006.

[Mar99] Marcelo Kallmann, Daniel Thalmann: "Direct 3D Interaction with Smart Objects", Proc. of the ACM symposium on Virtual reality software and technology, pp.124-130, 1999.

[Jam90] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, *Computer Graphics 2nd edition* ,ADDISON-WESLEY ,1990.

[Tom06] Tomas Akenine-Möller, Eric Haines: "REAL-TIME RENDERING", Born digital Inc., pp.36-42,2

[Kov02] Kover L., Gleicher M. and Pighin F.:" Motion Graphs", ACM Transactions on Graphics, Vol.21, No.3, pp. 473-482, 2002.