Data Driven Avatars Roaming in Digital Museum

Xiaoting Wang, Xiangxu Meng, Chenglei Yang and Junqing Zhang

School of Computer Science and Technology, Shandong University

Abstract—This paper describes a motion capture (mocap) data-driven digital museum roaming system with high walking reality. We focus on three main questions: the animation of avatars; the path planning; and the collision detection among avatars. We use only a few walking clips from mocap data to synthesize walking motions with natural transitions, any direction and any length. Let the avatars roam in the digital museum with its Voronoi skeleton path, shortest path or offset path. And also we use Voronoi diagram to do collision detection. Different users can set up their own avatars and roam along their own path. We modify the motion graph method by classify the original mocap data and set up their motion graph which can improve search efficiency greatly.

Index Terms—Motion Capture, Motion Graph, Digital Museum, Virtual Touring

I. INTRODUCTION

With the fast development of network and virtual reality techniques, more and more virtual museums have been created to show exhibits (items on display). They are points of special interests and provide real-time journeys which offer online multimedia guided tours on the Web. More than 20 digital museums in Chinese Universities have been set up. In digital archaeology museum of Shandong University [1], a walkthrough system was constructed to guide the virtual journey. Although many virtual museums provide the roaming function, but the roaming is either the change of the viewpoint or an avatar transiting here and there without the actual animation of the body. In this paper we implement a data driven roaming system with real walking motions, along the particular paths and avoid colliding to each other(see Fig. 1).

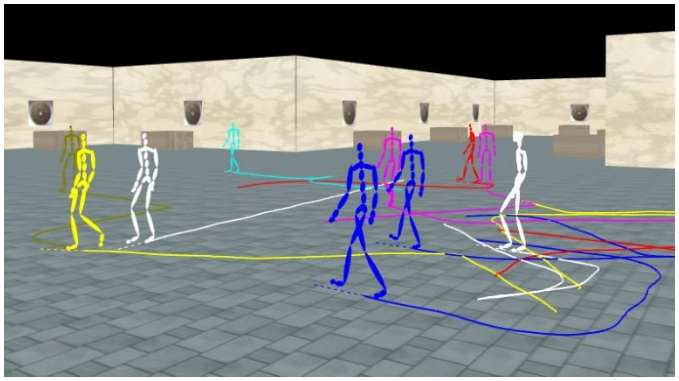


Fig. 1. Avatars roam in digital museum. The colored curves on the ground stand for the random walking path of the same colored avatars.

To realize the roaming system with high reality, this paper focuses on three aspects: how to design the animation of avatars vividly; how to plan the appropriate paths in digital museum; and how to avoid collision in the scene.

In order to animate the avatar with real person's motions, we use several simple walking clips of motion capture data, including stand-go, go forward, go-stand, turn left 90 degree, turn right 90 degree, turn left 45 degree and turn right 45 degree, totally 26 seconds. Using motion graph (MG) method [2], find the most similar transition points between different motion clips or the same clip, and blend the neighborhoods of the transition points in the motions to form phases of new transition motions. This method can connect motions together seamlessly. After the process of searching transition points in MG and synthesizing new motions, we can get continuous natural walking motions.

If the original mocap data are ample and various, we firstly classify the original data into several classes, each class has a MG and a coefficient between classes which represents their similarity. If the following motion does not change much, we only need to search in the same MG. Otherwise, according to the degree of the motion changed, we search the correlative MGs but not all of the classes. This method reduces the searching range greatly.

Suppose the transition points are selected randomly among all satisfied points without any constraint, the roaming will be an aimless screen saver at most. So the path planning is an important question in roaming. In our digital museum, the exhibits are located along the wall and the 2D Voronoi diagram (VD) of the museum is constructed. Avatars can walk along a Voronoi skeleton path, shortest path or offset path.

When multiple avatars roam in digital museum along their own paths, collision detection must be considered. To avoid collision, we use Voronoi diagram to determine avatars' locations quickly.

The main contribution of this paper is: only 26 seconds mocap data is used to complete the continuous roaming in the museum. Improve the MG method by classifying the original mocap data to reduce searching time. And we introduce VD into the digital museum to form the visiting path and avoid collision.

Sections 3 details how to search transition points in motion clips and synthesize transition motions, in order to naturally connect short clips into long motion; in section 4, we construct the Voronoi skeleton path, shortest path and offset path of the museum; in section 5, we use Voronoi diagram to settle the collision detection question quickly in a multi-avatar system; and at last we give some results and analysis， and discuss the limitation and future works.

Manuscript received on March 6, 2009.
E-mail: wangxiaoting@sina.com.

II. RELATED WORK

Roaming with reality is an animation problem. Physically based character animation can be divided into three classes: continuous constrained optimization, simulation and data driven method. Continuous constrained optimization adapts to simple characters or physical simplified situations [3]. When applied to high degree objects such as human, its formula is too complex to be settled. Simulation is focused on Robotics [4] and Biomechanics [5], but the dynamic theory is too complicated, until now there is still not a perfect body control system and the simulation of human body is far from good enough. The simulation has vast potential however, with the deeper understanding of dynamic theory, real motion can be devised [6]. Data driven method [7] has been popularized because of the motion capture device, and is used to modify, synthesize and analyze motions. Because the data is captured from human's action, this method has most reality. Also it has the disadvantage of hard to modify and reuse. Combining simulation and data driven method [8], getting full use of their advantages is a hotspot in recent years.

In this paper we use data driven MG method [2] to animate avatar. MG is a directed graph and its edge is a clip of motion (one or more frames) and a point is a feasible transition. Its basic theory is to find the similar frames (points in MG) between motions as transition points and add a phase of new motion into them (a new line between two transition points). Find out all these points and connected, we can get a MG.

VD has been widely used in computer graphics, robot and so on [9]. The VD of a polygon divides the polygon into a series of cells (Voronoi region,VR) reasonably. Each VR corresponds to an edge or a concave vertex of the polygon, and points inside the VR are closest to this edge or concave vertex. Thus, VD offers a good sort of the points in 2D space according to vicinity, which can finely describe the objects' space adjacency relation in the scene. In VD of the polygon, the common edge of two VRs is called a Voronoi edge. Here we construct the VD of the museum and use the Voronoi edges in path planning.

Collision detection is an inevitable problem in virtual reality [10]. The common method is to use bounding box and we can use Voronoi region to help decision-making.

III. AVATAR ANIMATION

3.1 Motion Graph

To animate the avatar, we first setup the MG. A motion graph is built by creating transition motions that connects different parts of a data set. Try to find the most similar frames in the mocap data and connect them to form a new synthesized transition motions.

Figure 2(a) is the new transitions synthesis from two motion clips and Figure 2(b) gives an example of MG representing the 7 walking clips and their new synthesized transition motions used in this paper. The frame rate is 120 fps and the frames are 444, 441, 285, 424, 503, 520, 471 separately, totally 3088 frames, 26 seconds.

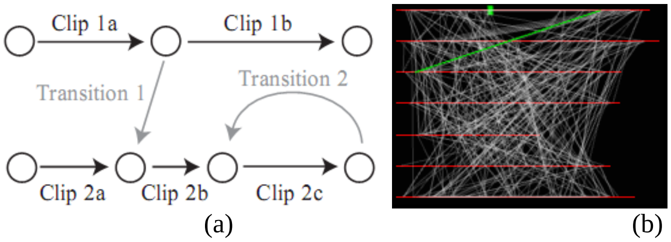


Fig. 2. An example of MG. (a) The schematic diagram (b) Red lines indicate 7 walking clips and white lines are new synthesized transition motions. Green point is the current frame and green line is the coming transition line.

3.1.1 Nomenclature

First, we give some denotations used in this paper:

P is a path in MG, which can be complex path (include circle),.

T is the track produced by an avatar in virtual environment according to a path coming from MG.

$\mathbf{P_{result}}$ is the path synthesis result which initial value is null.

$\mathbf{P_{opt}}$ is the optimal path which initial value is null.

$\mathbf{T_{opt}}$ is the optimal track which initial value is null.

$\mathbf{D_{opt}}$ is the difference between the current track and the optimal track. Its initial value is +infinite.

$\mathbf{L_{opt}}$ is the length of optimal path which initial value is 0.

$\mathbf{L_{step}}$ is the incremental step in synthesis which initial value is 1.00. For example, current L_{opt}=3.5m, then next step will detect to $L_{opt} + L_{step}$ = 4.5m.

$\mathbf{L_{over}} = L_{opt} + L_{step}$is the stop length in incremental synthesis.

$\mathbf{R_{use}}$ is the ratio of incremental synthesis result used and its initial value is 0.25. If a synthesis has 12 points, only 12*0.25=3 points will be added to P_{opt}.

$\mathbf{V_{start}}$ is the start of each step in incremental synthesis which initial value is V_{ini}.

$\mathbf{P_{start}}$ is the initial path, only contains V_{start} at first.

$\mathbf{S_p}$ is the stack used to save the candidate path which initial path is P_{ini}.

A path P in MG is related to a track T in virtual environment. After the MG is constructed, face to the avatar's objective track T_{obj}, the next step is to find the optimal path P_{opt} and let the P_{opt} related T_{opt} has minimal difference with T_{obj}.

3.1.2 Algorithm implementation

Firstly we get the initial transition matrix M_i according to avatar's initial pose and location and find the start point V_{ini} in MG (the simple way is to start with the stand-go clip). Synthesizing the whole path T_{opt} directly needs vast solution space, so we use incremental synthesis. The basic idea is starting from the initial motion to go forward, after an interval, determine whether the current track deviate the objective track, if not, continue this motion; otherwise try to transit to another appropriate motion, in order to match the objective track.

Combined branch and bound with DFS, the pseudocode is as follows:

```
while Length($T_{opt}$) < Length($T_{obj}$)
// the length of synthesis is not long enough
$D_{opt}$ = +infinite
    $S_p$.clear()
// clear stack
    $S_p$.push( $P_{start}$ )
```

```
// push the initial path into stack
while $S_p$ not empty
// stack is not empty
    $S_p$.pop( $P_x$ )
// pop stack
    $T_x$ = pathToTrack( $P_x$ )
// convert path to track
    if Length($T_x$) < $L_{over}$
// length is shorter than stop length
        $V_{tail}$ = $P_x$.lastVertex()
// get the last point in the path
for each neighbour V of $V_{tail}$
// traverse its neighbors
            $S_p$.push( $P_x$ + V )
// form new candidate path and push into the stack
        end for
    else
// the length is enough
        $D_x$ = D($T_x$, $T_{obj}$)
// compute the difference between $T_x$ and $T_{obj}$
        if $D_x$ < $D_{opt}$
// find the better synthesis
            $P_{opt}$ = $P_x$
// renew the optimal synthesis path
            $D_{opt}$ = $D_x$
// renew the value of $D_{opt}$
            $T_{opt}$ = $T_x$
// get the optimal track
        end if
    end if
end while
$P_{result}$.append( $P_{opt}$.head( $R_{use}$ ) )
// use a part of $P_{opt}$ as result
$V_{start}$ = $P_{result}$.lastVertex()
// renew the start of next step
$P_{start}$ = <$V_{start}$>
// renew the initial path
$L_{over}$ = Length( $T_{opt}$ ) + $L_{step}$
// go forward
if $L_{over}$ > Length( $T_{obj}$ )
// too long
$L_{over}$ = Length( $T_{obj}$ )
// truncate $L_{over}$
end if
end while
```

3.1.3 Analysis

The experiment environment used in our programming is: Windows 2003、AMD Athlon64 X2 3600+、NVIDIA 7900、1G RAM, developed by C++. The construction of MG needs preprocess and the time of motion synthesis varied greatly according to the path and parameters. There are several factors infected the speed greatly: the length of the mocap data; the shape of the path; L_{step} and R_{use}; the original position and direction between the avatar and the path. Given more source data, better synthesis result can be realized which matches the objective path better, but the preprocess time is increased in $O(n^2)$. L_{step} stands for the length detected ahead in each step, and the bigger it is, the match is better, but the time is increasing rapidly. R_{use} is the ratio we used the synthesis result in each step, the smaller, the better and more slowly. If the start position is in the path and the original direction is along the path, the time cost used to motion synthesis are small than other situations. Table 1 gives 9 result of different parameters when walking along a 33 meters elliptic arc. Figure 3(1)-(9) are the shapes of path synthesis.

TABLE 1: RESULT OF DIFFERENT PARAMETERS.

	position	direction	Frame Rate	Step Length (L_{step})	Use Rate (R_{use})	Time Cost
1	0 0 0	0.0	120	1.0	0.3	72s
2	0 0 0	0.0	120	0.5	0.3	7s
3	0 0 0	0.0	120	0.5	1.0	2s
4	0 0 0	0.0	120	0.6	0.5	9s
5	0 0 0	0.0	120	0.7	0.5	19s
6	0 0 0	0.0	120	0.8	0.5	46s
7	0 0 0	0.0	120	1.0	0.5	32s
8	0 0 0	-45.0	120	1.0	0.5	166s
9	0.5 0 -0.5	0.0	120	1.0	0.5	217s

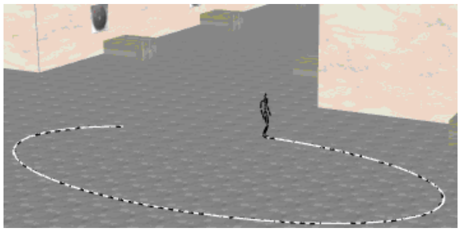


Fig. 3(1)

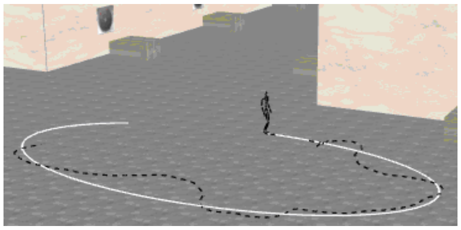


Fig. 3(2)

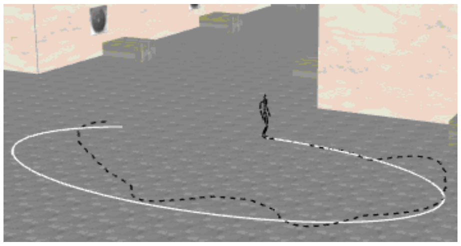


Fig. 3(3)

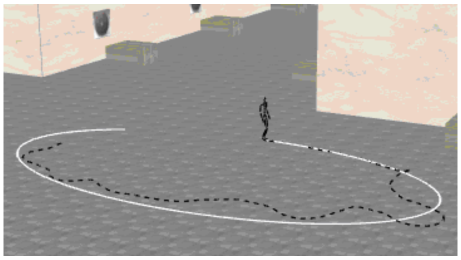


Fig. 3(4)

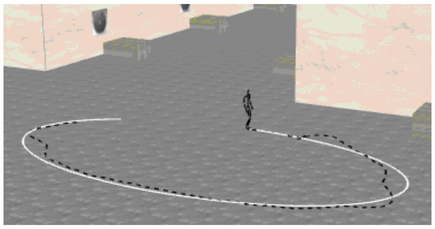


Fig. 3(5)

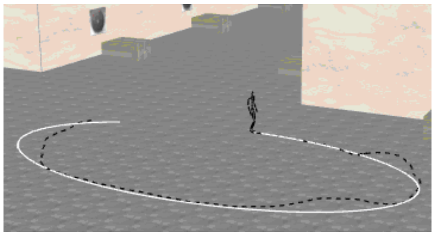


Fig. 3(6)

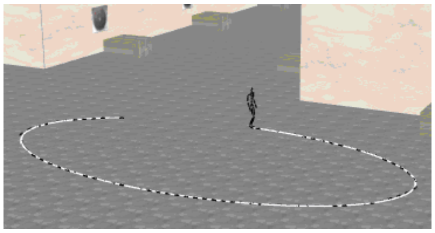


Fig. 3(7)

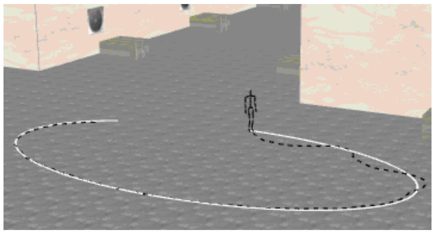


Fig. 3(8)

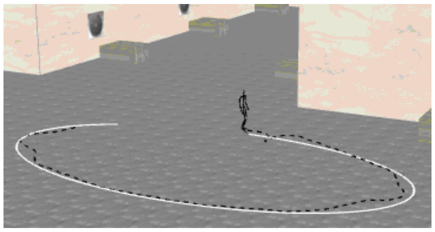


Fig. 3(9)

When realize motion transition, it's not transiting from one frame into another in fact. Although in MG, a line connects two transition points, the transition is not a point, but a neighborhood around the transition point. This is because even the two frames are exactly the same the two motions' tendency may be very different. If these two motions are connected, it will lead to false motion.

Suppose the start of the transition section is the S_p frame in motion S_m, and the end is the D_p frame in motion D_m, represented as (S_m, S_p, D_m, D_p). The radius of the transition section is L, so the source transition section is R_s,: [S_p – L, S_p + L] and the destination transition section is R_d: [D_p – L, D_p + L]. There's an important thing we should pay attention to, R_s may contains the current frame or the last feasible transition point in S_m. In the latter situation, if transit from S_p – L to S_p + L without consideration, it will lead to no transition point to use or motion retreat and the motion cannot be continued. So we need to adjust the radius of transition section in this situation.

3.2 Improved Motion Graph

This method is appropriate to motion which belongs to one kind of class. If the original mocap data includes several kinds of motions, the workload of search is too heavy. So we firstly classify the original data into several classes, each class has a MG and a coefficient between classes which represents their similarity. The algorithm to calculate the coefficient between class A and class B is:

Identify a basic gait sequence $a_1, a_2, \ldots, a_n$ from A;

Identify a basic gait sequence $b_1, b_2, \ldots, b_n$ from B;

The similarity value is

$$\boldsymbol{\Sigma\ (\ (a_1\ -T_1R_1\ S_1b_1)\ +\ (a_2\ -\ T_2R_2\ S_2b_2)\ +\ \ldots\ +\ (a_n\ -\ T_nR_nS_nb_n)\ ).}$$

If the following motion does not change much, we only need to search in the same MG. Otherwise, according to the degree of the motion changed; we search the correlative MGs but not all of the classes. This method reduces the searching range greatly.

IV. PATH PLANNING

In this system, avatar can roam in digital museum along Voronoi skeleton path, shortest path or offset path(see Fig.4).

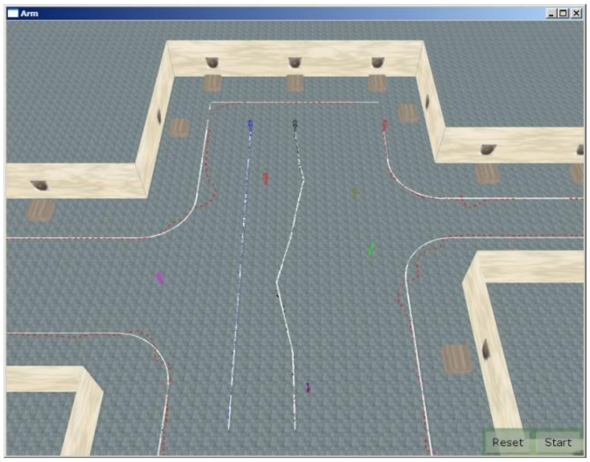
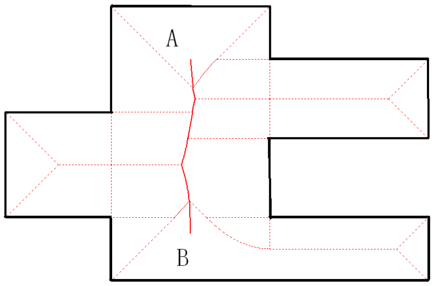


Fig. 4. A real situation in system.

4.1 Voronoi skeleton path

When the viewer walks from exhibit A to B direction, he can look through the museum along Voronoi skeleton path [11]. Voronoi skeleton path is the middle axis of the polygon, so viewers walk through the middle part of the virtual environment and can see exhibits on both sides of the path(see Fig.5).



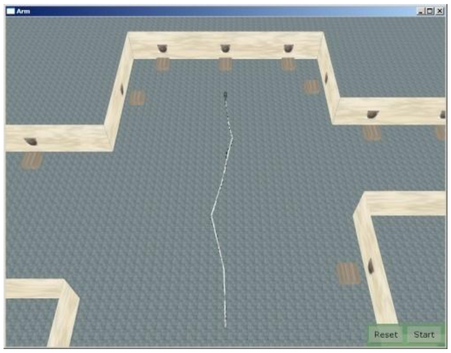


Fig. 5. The scene of digital museum and its Voronoi skeleton path.

4.2 Shortest path

If we are familiar with the exhibits and only want to visit the special one we are interested in, the shortest path (see Fig. 6) is the most appropriate. Using the Dijkstra algorithm to compute the shortest path, the time complexity is O(n^2) in the worst case; here we can use a new algorithm with O(n) running time to compute the shortest path by VD [11].

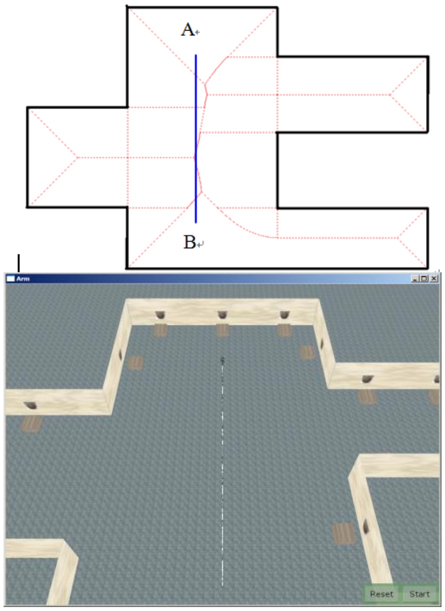
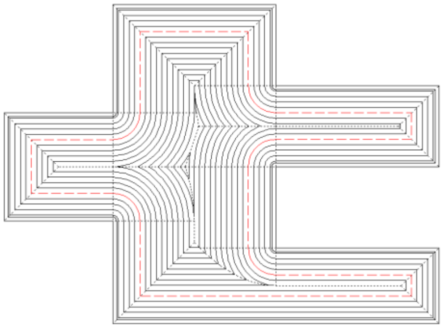


Fig. 6. Shortest path.

4.3 Offset path

When the viewer wants to visit exhibits one by one, offset path (see Fig. 7) can be used. Users can define the offset value in the scene [11].



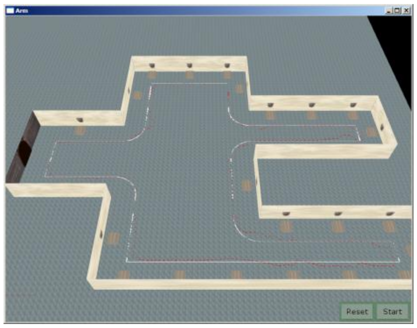


Fig. 7. Offset path

4.4 Analysis

Studying the shape of the Voronoi path, shortest path and offset path, we found they are mainly constructed by some basic patterns shown in Figure 8. And our method can settle these situations well if given appropriate parameters. This ensures the avatars walking naturally along the objective path without big deviation.

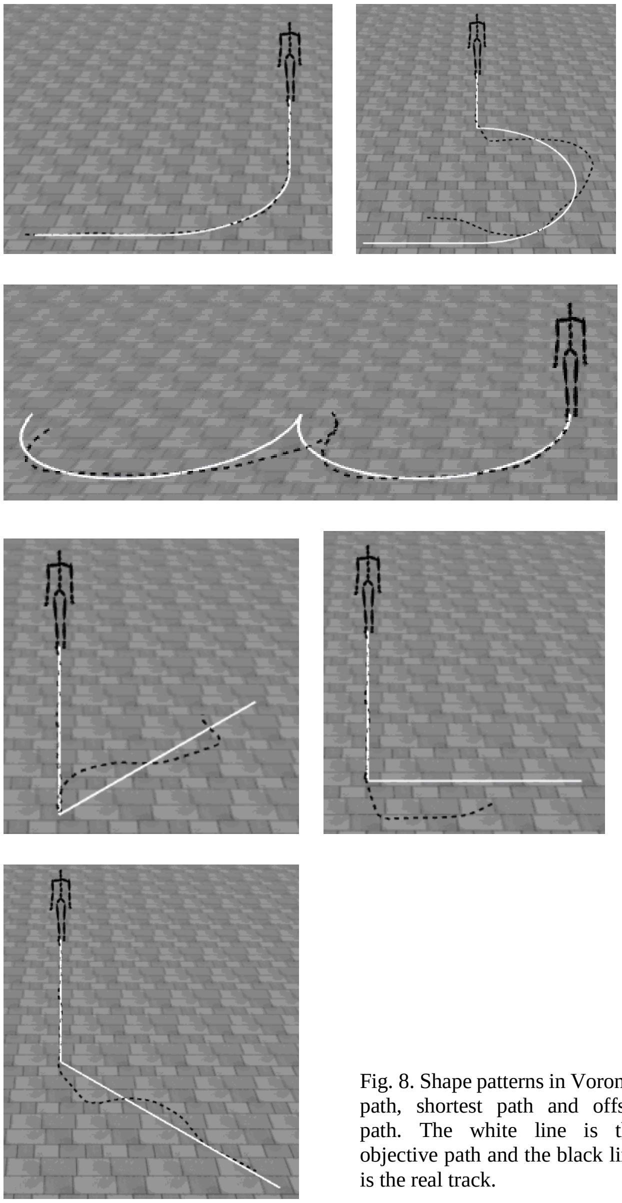


Fig. 8. Shape patterns in Voronoi path, shortest path and offset path. The white line is the objective path and the black line is the real track.

V. COLLISION DETECTION BETWEEN AVATARS

When there are many avatars, the collision detection computing between each two is large. We can decrease computing by restrict computing in a local part, that is, one avatar only need to detect avatars in the same or neighbor Voronoi regions through computing every avatar's Voronoi regions that its bounding box covers. In addition, if the two center points of the two avatars are not visible to each other, then they will not collide in next moment. To speed up, we use circle as the bounding box and deal with it combining with visibility computing described in another paper [12]. Let the center points of the two avatars be p_1 and p_2, r_1 and r_2 be the circle's radius, respectively. We can determinant whether the two avatars collide according to the relation of d(p_1, p_2) and r_1+ r_2.

VI. FUTURE WORK

Animation of avatars is an important function to add reality of digital museum. In our work, a data driven roaming system is designed. The animation of avatars is synthesized using MG based method, walking along the Voronoi skeleton path, shortest path, offset path or random path without collide to each other. We will study dynamics of the hand, companying with continuous motions, to do some gestures such as opening the door, feeling objects and shaking hands in the future.

ACKNOWLEDGEMENT

This work was supported by a grant from The National Basic Research Program (973 Program) (2006CB303102), the National Nature Science Foundation of China (60703028).

REFERENCES

[1] Xiang Hui, Meng Xiangxu, and Yang Chenglei, Design and Implementation of Digital Archaeology Museum of Shandong University, *Journal of System Simulation*, 15(3), 2003

[2] Lucas Kovar, Michael Gleicher, and Frederic Pighin, motion graphs, *SIGGRAPH* 2002.

[3] Andrew Witkin and Michael Kass, Spacetime constraints, *SIGGRAPH* 1988.

[4] R. L. Andersson, *A Robot Ping-Pong Player: Experiment in Real- Time Intelligent Control*, The MIT Press, Cambridge, MA, USA.

[5] M.G. Pandy and F.E. Zajac, Optimal muscular coordination strategies for jumping, *Journal of Biomechanics* 24 (1991), pp. 1–10.

[6] KangKang Yin, Stelian Coros, Philippe Beaudoin, and Michiel van de Panne, Continuation Methods for Adapting Simulated Skills, *SIGGRAPH* 2008.

[7] James McCann, Nancy Pollard, Responsive Characters From motion Fragments, *SIGGRAPH* 2007.

[8] Zordan, V. B., Majkowska, A., Chiu, B., Fast, M., Dynamic Response for motion Capture Animation, *SIGGRAPH* 2005

[9] Yang Chenglei, Wang Lu, Wang Xiaoting, Xu Yanning, Meng Xiangxu, A System Framework and Key Techniques for Multi-user Cooperative Interaction in Virtual Museum Based on Voronoi Diagram, *Proc. of The 9th International Conference on CSCW in Design*, May 24-26, 2005

[10] Ruspini, Diego and Oussama Khatib, Collision/Contact Models for the Dynamic Simulation of Compex Environmnets, *IEEE/RSJ International Conference on Intelligent Robots and Systems:IROS'97*, September 1997

[11] Chenglei Yang, Meng Qi, Jiaye Wang, Xiaoting Wang, and Xiangxu Meng, Shortest Path Queries in a Simple Polygon for 3D Virtual Museum, *7th Annual International Workshop on Computational Geometry and Applications (CGA'07), ICCSA (1) 2007*: 110-121

[12] Yang Chenglei, Wang Lu, Wang Xiaoting, Xu Yanning, Meng Xiangxu, A System Framework and Key Techniques for Multi-user Cooperative Interaction in Virtual Museum Based on Voronoi Diagram, *Proc. of The 9th International Conference on CSCW in Design*, May 24-26, 2005

Xiaoting Wang received her B.E. degree from the school of computer science and technology, Shandong University, China, and the M.Phil. degree in computer science from the University of Hong Kong. Her research interests are in computer graphics, vitual reality and animation.

Chenglei Yang is an associate professor of Shandong University and currently the vice-dean of the school of computer science. He received his B.Sc., M. Eng. and Ph.D. Degrees from Shandong University in 1995, 1998 and 2004 respectively. He has published about 40 papers in the areas of computer graphics, computational geometry, image processing and virtual reality.

Xiangxu Meng received B. Sc and M. Eng Degrees in 1982 and 1985 from Shandong University. He obtained Ph.D. degree from Institute of Computing Technology, Chinese Academy of Science in 1998. Now he is a professor and also Dean of school of computer science and technology. He is a commissioner of information department of science and technology committee of Ministry of Education, specialist of CMIS of National 863 projects, associate director of CAD/CG Committee of Chinese Computer Association. He has published more than 150 papers and undertook 20 national or Shandong province projects. His research interests are HCI&VR, computer graphics, image processing, CAD/CAM/CIMS and grid computing.

Junqing Zhang (1987~),B.S in computer science and technololy school of Shandong University in China,jinan.I am interested in computer geometry.