PERGAMON

Computers & Graphics 27 (2003) 231–238

COMPUTERS & GRAPHICS

www.elsevier.com/locate/cag

Easybowling: a small bowling machine based on virtual simulation

Zhigeng Pan[a,b,*], Weiwei Xu[a], Jin Huang[a], Mingmin Zhang[a], Jiaoying Shi[a]

[a] *State Key Laboratory of CAD and CG, Zhejiang University, Hangzhou 310027, People's Republic of China*
[b] *Institute of Virtual Reality and Multimedia, HZIEE, Hangzhou 310027, People's Republic of China*

Accepted 27 November 2002

Abstract

In this paper we describe a virtual bowling game machine called EasyBowling Machine, which is designed and implemented based on techniques such as Virtual Reality, animation, and image processing. To introduce Virtual Reality technique into this virtual bowling game, our system provides a real game mode: players play the game by throwing a real bowling ball, and then the EasyBowling system uses a PC Camera to detect the motion of the real bowling ball. After the motion parameters (ball direction, ball force, etc.) are computed, the movement of the bowling ball and its collision with pins are simulated in real-time and the result is displayed on a large display screen. The most obvious advantage of such bowling game machine over other existing bowling games is that the system integrates body exercise into game playing. The implementation techniques are discussed in detail, and the prototype system illustrates the feasibility and efficiency of our method.

Keywords: Virtual reality; Bowling game; Motion detection; Collision simulation

1. Introduction

As a new branch of Computer Graphics, Virtual Reality (VR) has attracted a lot of attention in recent years [1]. VR is the use of computer graphics systems in combination with various display and interface devices to provide the effect of immersion in the interactive 3D computer-generated environment. We call such an environment a *virtual environment* (VE). Research and development into VR and VE applications can be found in many places all over the world [2]. VR/VE has been applied into entertainment for long time, and we call this as VR entertainment. There are two typical kinds of applications in VR entertainment: virtual games and amusement park. For VR entertainment applications, they have the following requirements: more interactivity, more feedback, total immersion in the environment, and transformation of the role of the audiences or the riders into an integrated entertaining process.

For virtual games system, usually it needs creation of a virtual environment, and the user should have a real feeling of the game and the system should support immersion in the action. Amusement Park application must combine an actual ride with fancy visual, audio, and small effects. Game is one important motivation to earlier Virtual Reality research. Recently, a lot of electronic games employ virtual reality techniques to improve their attraction, e.g. Driving Simulation, Shoot game, Virtual Golf and so on. VR Games provide a more immersing environment through vivid scenes and real playing-mode, users play a role in game other than only a manipulator.

VR techniques have been employed in many games; some of VR games integrate ball game, such as Virtual Golf, TV Ping-Pong game. In these games, infrared devices are used to detect the real movement of Golf or

*Corresponding author. Tel.: +86-571-87951045; fax: +86-571-87951780.
E-mail address: pzg@cad.zju.edu.cn (Z. Pan).

0097-8493/03/$ - see front matter
PII: S0097-8493(02)00280-7

Ping-Pong racket, and then system simulates the subsequent movement in computer. Inspired by those games, we design the Virtual bowling game machine.

Tenpins is a popular gymnastic game. The simulation of mutual collision of bowl and clubs is also a typical problem in Computer Graphics. As we know, there are some bowling games or game machines in use, but they often require the users to operate simulation device, such as keyboard, rolling ball, to throw the bowl in game, which is an obvious shortcoming [3,4]. We think those games ignore the real meaning of Tenpins: body exercise. Users do not need to throw the real bowl by themselves while playing the game, so the attraction of bowling game is lost seriously.

Therefore, we try to introduce VR techniques into bowling game, and compose a near-immersive bowling environment. Based on simulating collision realistically, the system enables users to throw the real bowl while playing the game. In our prototype system, a PC camera is used to detect the motion of bowl, and then the system simulates the collision according to the detected motion parameter. Since user can throw real bowl in our game, we successfully preserve the feature of original bowling game, body exercise.

The system structure and the features of our system are introduced in Section 2. In Section 3 we describe our motion detection algorithms, including two cases: bowling ball without rotation, and bowling ball with rotation. Collision detection and physical model based simulation are described in Section 4. Future works and conclusion are presented in Section 5.

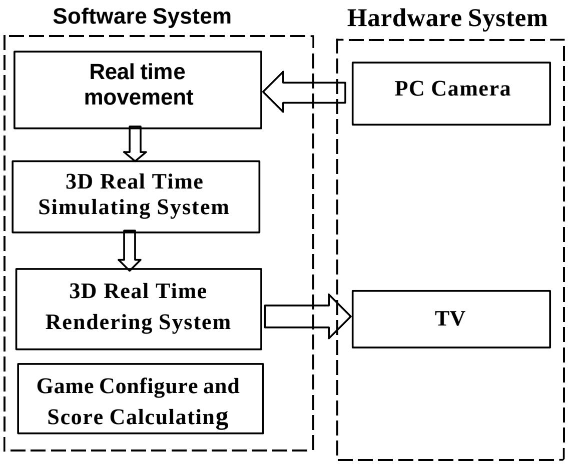

Fig. 1. System structure.

Fig. 2. Virtual bowling game machine.

2. System structure and applicable fields

2.1. System structure and feature

The Virtual bowling game machine provides same play-mode with real bowling game. Users throw the bowl firstly and system employs a PC camera to detect the speed and direction of bowl. Then the real-time collision simulation and the real-time scene rendering will show the user the simulating result quickly. Fig. 1 shows the system structure.

2.2. Places for installation

While using a camera to detect the movement of the rolling bowl, the distance needed to compute movement parameters is about 2 m. So, the system only requires a track about 2-m long (the real track of bowling system is about 19.1 m long). This reduces the area and cost of real bowling game heavily. Since collision is simulated on computer, the complicated mechanical devices are saved. This also reduces the cost. In addition, system can be configured to different game-mode; a tutorial course is also integrated into the system, which can teach the user on how to play bowling game. Compared to real bowling game, our virtual bowling game machine (Fig. 2) has such features: more flexible, low cost and easy to install.

In Fig. 3, a student in our lab is playing virtual bowling game: he is going to throw the bowling ball. Since virtual bowling game machine only needs small space, it is suitable for single user.

Virtual bowling game machine can be installed in game centers; it can also be installed in small hotels as a body exercise machine. Virtual bowling machine can also be a family game machine because it only requires a small space.

3. Motion detection

Motion detection is achieved with two steps: first, segmenting the bowl from the image that is captured by camera and then calculating the position of the bowl center; second, computing linear velocity and rotational velocity [5,6].

3.1. Segmentation of the rolling bowl

The first step of motion detection is to segment the bowl from the image captured by a PC camera. Since the result of segmentation will influence the subsequent computing of movement parameters and it must satisfy the real-time requirement that is very important in electronic game, we have to trade-off between visual effect and response time. Segmentation algorithm based on color feature makes use of the characteristic color of target to clustering. Such algorithm concentrates on calculating the similarity among colors; its computing quantity is relatively small compared to other algorithms. If the color of bowl is limited, this algorithm can satisfy our requirements. For convenience of detecting, we assume that the color of bowl is red in our system. This is reasonable, since the bowl is provided with the game machine and the system can change the system parameters based on the color of the bowl.

To eliminate the disturbance of shadow, a transform is applied to the RGB value of pixels. Since red value is the primary component of the color of the bowl, this transform is designed by

$$T = (R - G) + (R - B). \quad (1)$$

Fig. 3. Playing virtual bowling game.

From statistics of experience, we find that the color of bowl satisfies

$$T > 50. \quad (2)$$

Therefore, we design a rapid segmentation algorithm as follows:

Step 1: Perform the transformation in RGB space, and compare the transforming result with T

Step 2: If T satisfies (2), set the pixel value to 1, else set to zero.

Step 3: Do the *close* operation in mathematics morphologic.

Step 4: Remove the small region

After segmentation, the system will compute the position of the bowl center (Fig. 4). Since the projection of bowl on image planar is a circle, we use the center of the circle to substitute the bowl center. The center of circle is computed through Hough transform by making use of the edge points in the segmented image.

3.2. Computing linear velocity

Modeling camera as a pinhole camera, geometic relationship between track and bowl can be established. Fig. 5(a) shows the image of the bowling track (wooden alley). From this image, one can see the perspective effect of the camera clearly: the two parallel edges of the track will intersect in the image plane. Fig. 5(b) illustrates the simplified geometric representation of the track and bowl. In Fig. 5(b), D_1D_2 and E_1E_2 are two parallel lines drawn on the wooden alley for the purpose of calibration. Assume C_1, C_2 are the two positions of the center of bowl found in image sequence, the x and y displacement of bowl can be calculated using this simple geometric relationship.

According to perspective geometry, two parallel edges of the track will intersect a point M on image plane. We can connect M, and C_1, and prolong it to intersect E_1E_2 with points $F_{1,}M$, and C_2 and prolong it to intersect E_1E_2 at the point F_2. The relationship of $C_1F_1//C_2F_2//D_1E_1//D_2E_2$ should exist in original geometry space. So, the x displacement of the center of bowl is just F_1F_2. With respect to y displacement, we only need to calculate the difference of F_1C_1 and F_2C_2. Measuring the real width W of E_1E_2 in track. After

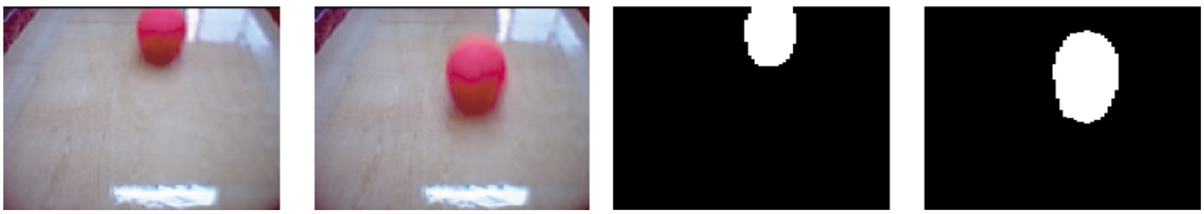

Fig. 4. Segmentation of bowl.

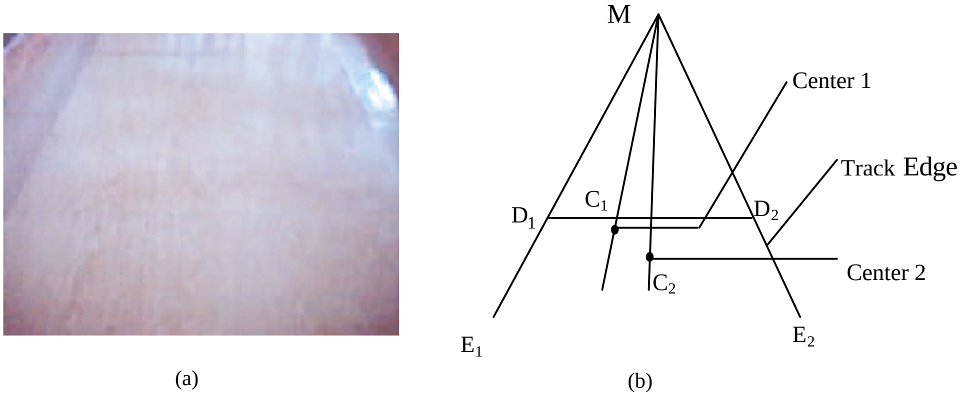

Fig. 5. (a) and (b) The track image and its geometry representation.

finding the position of the two end points of E_1E_2 on image plane, x displacement can be calculated by

$$D_x = W * (E_1F_1 - E_2F_2)/E_1E_2. \quad (3)$$

After measuring the length of D_1E_1 at the same time and finding the position of two end points of D_1E_1 on image planar, y displacement can be calculated as

$$D_y = (F_1C_1/F_1G_1 - F_2C_2/F_2G_2)L. \quad (4)$$

While applying this method, we label the two assistant parallel lines on bowling track in advance. After measuring the data required in (3) and (4) (the assistant lines can be removed after measuring), the linear velocities can be computed rapidly through the position of the center of the bowl.

3.3. Detection of rotational velocity

Compared to the detection of linear velocity, the detection of the rotational velocity is a non-trivial task. The most difficult of the detection arises from the lack of feature points on the surface of the bowl. Thus, two colorful belts are attached to the surface of the bowl as artificial feature points. The normal of belt plane is then computed to determine the rotational speed of the bowl. The colors of the belts are chosen to be far from each other in HSV space. As mentioned above, the color of the bowl is red in our testing system, and hence, green and blue are selected. During the detection, the belts are segmented with the hue value of each pixel, for it is well-known that hue is more stable under different lighting conditions.

The reason why we attach two belts to the bowl is to determine the instant rotational axis of the bowl. Fig. 6 illustrates the procedure. Assume feature point A is rotated to A', and B is rotated to B' (see Fig. 6). If $AA'//BB'$, then the rotational axis must be the intersection line of the planar OAB and $OA'B'$, otherwise it is the cross product of $AA' \times BB'$. Thus angular velocity

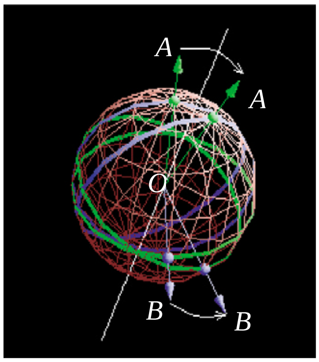

Fig. 6. Feature points and rotation axis.

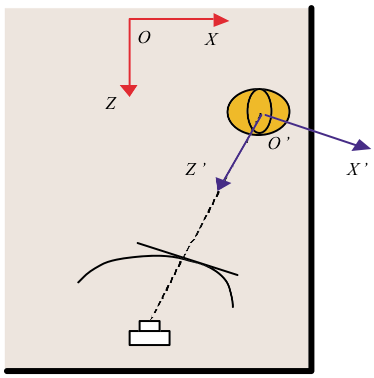

Fig. 7. Frame specification.

can be easily computed with $w = (\textit{the radian between PA and PA'})/\Delta T$, where PA and PA' are the projection of A and A'on the rotation axis respectively, and the ΔT is the time interval.

In our system (Fig. 7), the normal of the belt plane is used to determine the feature points. Thus, we need to compute the normal of the belt plane. First, Hough transform is adopted to determine the eclipse which is the projection of the belt in the image plane. Second, coordinate frame is calculated to get the final result. A rotational frame O' (see Fig. 6) is fixed to bowl. The Z'-axis of this frame directs from the center of bowl to the camera and $X'Y'$ plane is parallel to the image plane. Let the normal of the belt plane be (α, β, γ), the radius of bowl in image is *rad*, the length of the ellipse short axis is b, and the angle between X'-axis and the short axis of eclipse is θ. The following formula deduces (α, β, γ) from *rad*, b and θ:

$$(\alpha,\beta,\gamma) = \left(\cos(\theta)\sqrt{1-\left(\frac{b}{rad}\right)^2}, \times \sin(\theta)\sqrt{1-\left(\frac{b}{rad}\right)^2}, -\frac{b}{rad}\right). \quad (5)$$

Assume that axis Z' is (A, B, C) in world frame O. Then the two angle components of the spherical coordinate of vector Z' are the following ($\alpha'$is elevation angle and β' is azimuth angle):

$$\sin(\alpha') = \frac{B}{\sqrt{A^2+B^2+C^2}}, \quad \alpha' \in \left(0, \frac{\pi}{2}\right),$$
$$\sin(\beta') = \frac{A}{\sqrt{A^2+C^2}}, \quad \beta' \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right). \quad (6)$$

The final direction of the normal of the belt plane in world frame is

$$\left(\cos(\theta)\sqrt{1-\left(\frac{b}{rad}\right)^2}, \sin(\theta)\sqrt{1-\left(\frac{b}{rad}\right)^2}, -\frac{b}{rad}\right) \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha') & \sin(\alpha') \\ 0 & -\sin(\alpha') & \cos(\alpha') \end{bmatrix} \times \begin{bmatrix} \cos(\beta') & 0 & -\sin(\beta') \\ 0 & 1 & \sin(\alpha') \\ \sin(\beta') & 0 & \cos(\beta') \end{bmatrix}. \quad (7)$$

From the above formula, two feature points A and B can be extracted from one image, which are the intersection point of the normal and bowl surface.

4. Physical model based simulation

4.1. Simulating procedure

For simulating movement and collision vividly, dynamics should be considered to construct the physical model of system. Because there is a sudden change in the system when collision take place, we divide the system state into steady state (collision does not take place) and temporary state (collision take place). Correspondingly, every body of system has two evaluators: steady evaluator and temporary evaluator. They will be triggered between the intervals of frames. When collision takes place, temporary evaluator would be triggered to deal with impulsive force caused by collision, calculating the sudden change of velocity. Steady evaluator will be triggered while there is no collision. It deals with contact force (such as support force from plane) and calculate the continuous portion in dynamics equation. Fig. 8 is the flow chart.

Temporary evaluator computes the impulse using the velocity and the coordinate of collision point, then it adds the increment of velocity to both bodies. Steady evaluator uses the approximate formula: $f(t+\Delta t) = f(t) + \Delta t f'(t) + O(\Delta t)$. Set a small value to Δt, the velocity and acceleration can be assumed to be constant in frame interval. Therefore, the velocity and acceleration of next frame can be calculated by

$$S(t+\Delta t) = S(t) + \Delta t V(t) + 1/2 A(t)\Delta t^2,$$
$$V(t+\Delta t) = V(t) + \Delta t A(t). \quad (8)$$

It is not enough to only find out the position of collision point between two frames. For example, assume that Bodies A and B collide at time $t+0.01$, and bodies C and B collide at $t+0.02$, the two collision take place in one time slice ΔT. Then the second collision will not happen or change the collision time because the first collision changes the position of body B. Thus, it is important to get the precise collision time in one time slice. We use dichotomy to get precise collision time. Consider every two body independently, use dichotomy to compute the collision time, then compute collision time of another pair until we find an earliest collision. The subsequent collision will be recalculated.

4.2. Collision detection

Real-time and precise collision detection and subsequent processing are the most difficult in collision simulation. At each interval between frames, we use a bisection algorithm to find the earliest collision point and then advance the simulating clock to do the subsequent collision detection similarly. This procedure is very necessary to ensure the correct simulating result. The sudden change in the velocity of objects caused by the impulsive force can be dealt with the method in [7].

To meet the real-time requirement, we use a simplified model of pins. Approximating its edge with broken lines, the pin can be viewed as the concatenation of several cylinders. While performing collision detection, every cylinder pair of two pins should be checked. The collision detection procedure is as following:

Step 1: For every cylinder of A, judge whether its ceiling penetrate into the cylinders of pin B or not.

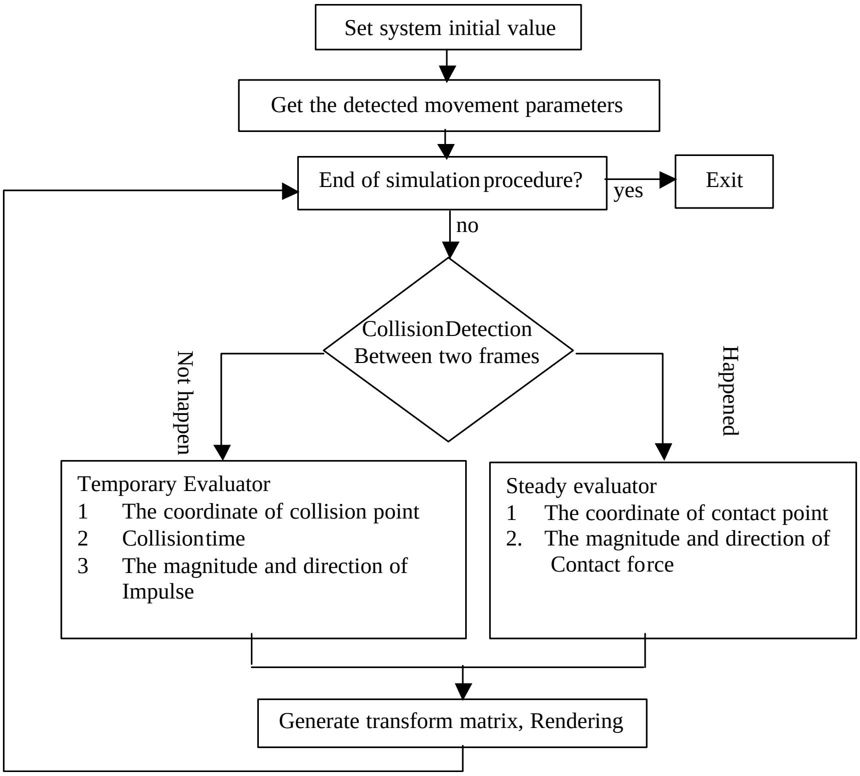

Fig. 8. Collision simulation procedure.

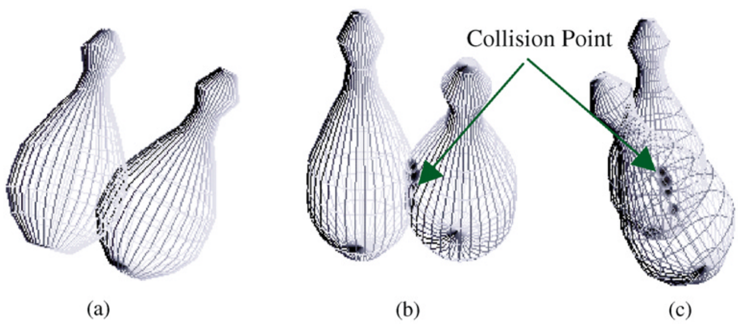

Fig. 9. (a)–(c) The result of collision point detection.

Step 2: If two cylinders intersect, calculate the distance between centers of the two cylinders. Assume that the radii of two cylinders are R_1, and R_2. Use $R_1 - R$, $R_2 - R$ as weight value to segment the connecting lines of two centers. The segment point is the collision point.

Step 3: If there are more than one collision point, average all collision points. We use "intersect depth" of two cylinders as weight. "Intersect depth" is calculated as $R_1 + R_2 - R$.

Fig. 9 shows collision point detection result. Fig. 9(a) illustrates the front view, Fig. 9(b) illustrates the rear view, and Fig. 9(c) illustrates the side view. Little ball represents the collision point; the bigger ball on the pins represents the average collision point.

4.3. Constraints

We mainly consider the wooden alley constraint: pins should not penetrate into the simulated wooden alley. So the mutual affection of pins and the alley should be computed in the simulation process. Assume that pin will never be off alley in this system, the composition force of gravity and support force of alley will have influence on the velocity and the acceleration of pins.

Thus, the alley constraint is satisfied. When the pin lies down, reversing the angular velocity perpendicular to ground and multiplying a decrease coefficient, rebounding of pins can be simulated [8].

Fig. 10 illustrates some results of the collision of the bowling ball with pins. They are snapshots of the collision process.

5. System implementation

This system is developed in PC platform, the operating system is Windows98, CPU is PIII667, memory size is 64M. Fig. 2 is the showpiece of bowling game machine. The machine can be divided into two parts: machine body and track. Track is made of special material, which is similar to real bowling track to guarantee the feeling of the user. Kinescope PC and camera are installed in the machine body. This virtual bowling machine is exhibited on the international Conference of CAD/Graphics '2001, and some conference attendees are very interested in it. The photo is taken with the EasyBowling system, showing the first author (the middle) with computer graphics experts from abroad (photo 1).

In the beginning of the game, we have created an animation showing the bowling game. In this animation, since it is pre-generated, we can have more realistic images. Fig. 11 is a snapshot of the animation clip.

6. Conclusion and future work

The virtual bowling system presented in this paper is an interesting example of application of VR techniques to game industry, with small space requirements. In addition, the cost of installing one set is quite cheap compared with the real bowling game device. Integrating body exercise into the game seamlessly is the most important feature of our virtual bowling system. For the time being, to get real-time effect we have made some assumption. Some actual conditions are simplified in our physical model. How to implement them will be taken into consideration in the future.

Future work of this project includes the following aspects:

(1) Support 3D realistic audio rendering [9]. Currently, we employ recorded audio clips for simulating sound of the virtual bowling ball when it rolls and

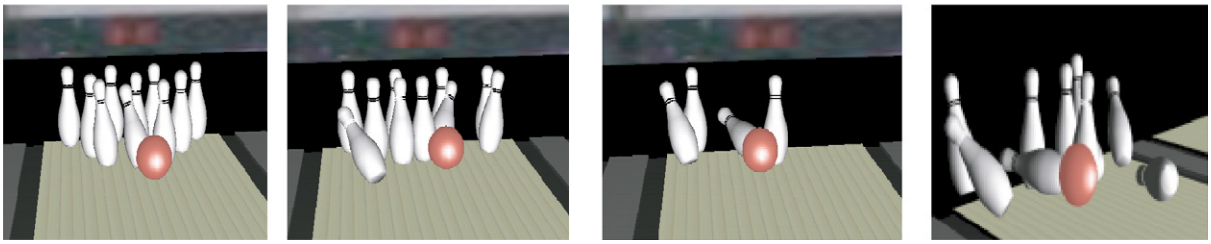

Fig. 10. Some simulating results.

Photo 1. EasyBowling Machine on CAD/Graphics'2001.

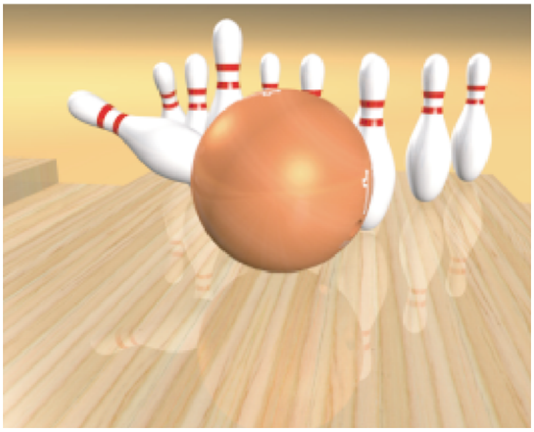

Fig. 11. Snapshot of the animation clip played in the starting period of the bowling game.

the sound of the collision among the virtual bowling ball and pins. We will generate 3D audio based on the physical model to have more realistic sound effect.

(2) Support 3D realistic graphics rendering. Currently, we use simplified models and the image quality is not so good. We are planning to employ some other real-time graphics rendering method (such as multi-resolution [10–12]) to support real-time 3D realistic rendering and make the clients have more immersive feeling.
(3) Use large screen instead of 29-in TV display. With a three-gun projector, the simulated image will be displayed on a large screen, and the players can have more immersion and can perform "aiming at" action, which is used widely in the usual bowling game. This is specially useful when there are some pins left on the virtual bowling lane.
(4) Internet-based game play. It is also a good idea to extend this game to Internet environment. Two players located in different places can play the two-play mode game if two EasyBowling game machines are connected through Internet. Of course, the program will be modified, and one game machine should be available to each player.

Acknowledgements

This research work is supported by 973 project (grant no: 2002CB312100), NSFC group innovation project (60021201), NSFC project (grant no. 60083009), Excellent Young Teacher Award Project of MOE, Excellent Youth Found of NSF in Zhejiang Province(RC00048).

We would like to thank Prof. Wei Zhao in Realtime Graphics Corp. in Beijing, Dr. Kun Zhou, Dr. Huagen Wan, Mr. Yu Liu, Mr. Ming Tao in the State Key Lab of CAD&CG, Zhejiang University, for their inspiration and kind help on the system implementation. In addition, the first generation prototype system was exhibited as the international CAD/Graphics'2001 conference (Kunming, August, 2001). We have got valuable suggestions on improving the system from Prof. Judith Brown from Iowa University in USA, Prof. J. Staudhamer from Florida University, USA, and Dr. Y.Y. Cai from NTU, Singapore.

References

[1] Zhigeng Pan, Jiaoying Shi, Qin Lu. Virtual Reality and its application in China—an overview. The International Journal of Virtual Reality 2000;4(3):2–12.

[2] Jiaoying Shi, Zhigeng Pan. Virtual Reality: fundamental and practical algorithms. Scientific Publishers, Beijing, 2002.

[3] Elf Bowling Game. http://www.freechristmasavers.com/elfbowling.htm

[4] Hough PVC. A method and means for recognizing complex patterns. US Pattern, 1962.

[5] Moore M, Wilhelms J. Collision detection and response for computer animation, proc SIGGRAPH '88. Computer Graphics, 1988;22(4):289–98.

[6] Kahn JK. Realistic animation of rigid bodies proc SIGGRAPH '88, Computer Graphics 1988;22(4):299–308.

[7] Casasent D, Richards J. High-speed acousto-optic mapping modulator for the generalized the Hough transform. Appl Opt 1993;32(35):7217–24.

[8] Mirtich B, Canny JF. Impulse-based simulation of rigid bodies. In: Proceedings of the ACM Symposium Interactive 3D Graphics, 1995. p. 181–8.

[9] Qiong Zhang, Jiaoying Shi, Zhigeng Pan, ARE: an audio reality engine for virtual environments. The International Journal of Virtual Reality 2000;4(3):37–43.

[10] Zhigeng Pan, Mingmin Zhang, Kun Zhou, et al. Level of detail and multi-resolution modeling for virtual prototyping. International Journal of Image and Graphics 2001;1(2):329–44.

[11] Heckbert PS, Garland M. Survey of polygonal surface simplification algorithms. Technical report, Carnegie Mellon University, 1997.

[12] Kun Zhou, Zhigeng Pan, Jiaoying Shi. A New mesh simplification algorithm based on vertex clustering. Chinese Journal of Automation 1999;11(2):83–91.