3D scenes simulation, animation, and synchronization in training systems with force back-coupling

Aleshin V, Klimenko S, Astskhov J, Bobkov A, Borodina M, Volegov D, Kazansky I, Novgorodtsev D, Frolov P

Department of General and Applied Physics

Moscow Institute of Physics and Technology, Moscow, Russia
e-mail: aleshin_vl@mail.ru, nov-dmitrij@yandex.ru

Abstract

In this project, we are trying to create a training system with force back-coupling and realistic visual entourage imitation. All existing training systems either have no visual environment (e.g. high-g training centrifuges for astronauts) or do not provide sufficient dynamic loads (e.g. flight and driving training systems) [1].

We designed the method of 3D environment creating and rendering and synchronized the resulting 3D scenes with a training system with force back-coupling. For our investigations, we chose the alpine skiing training system produced by the SkyTec Interactive company as this training system involves physical activity with complex coordination. For 3D scenes rendering, we used the OpenSceneGraph high performance open source 3D graphics toolkit. Our method provides deeper user immersion into the virtual environment and as a result more efficient training and corresponding skills mastering. The project was supported the Russian Foundation of Basic Researches grant 08-07-00399

Keywords: *Training system, 3D visual environment, simulation, animation, synchronization*

1. INTRODUCTION

Training systems with force back-coupling are often used by astronauts, sportsmen, in military training, and for entertainment. [2] Such training systems imitate real power loads and allow improving necessary skills. However, most of them either have poor visualization or have no visualization at all.

The virtual environment technology allows to experience deep immersion in the visual scene and train more efficiently.

In this project, we used the active alpine skiing training system produced by the SkyTec Interactive Russian company [3, 4]. This training system imitates a wide range of real physical loads that a sportsman experiences while skiing. The system simulates centripetal acceleration loads that the sportsman experiences while performing a turn. At the same moment (while performing a turn), the sportsman shifts sideward alone the training system platform the same as it happens on a real slope. The total range of sideward shifting is 5.5 m. This training system exceeds all its analogues in its characteristics.

For 3D scenes rendering, we used the OpenSceneGraph high performance open source 3D graphics toolkit. OpenSceneGraph uses the OpenGL programming interface. For 3D objects controlling, OpenSceneGraph uses a full-featured and industry-proven scene graph. Using the scene graph feature provides the 3D objects rendering and animation adaptive management.

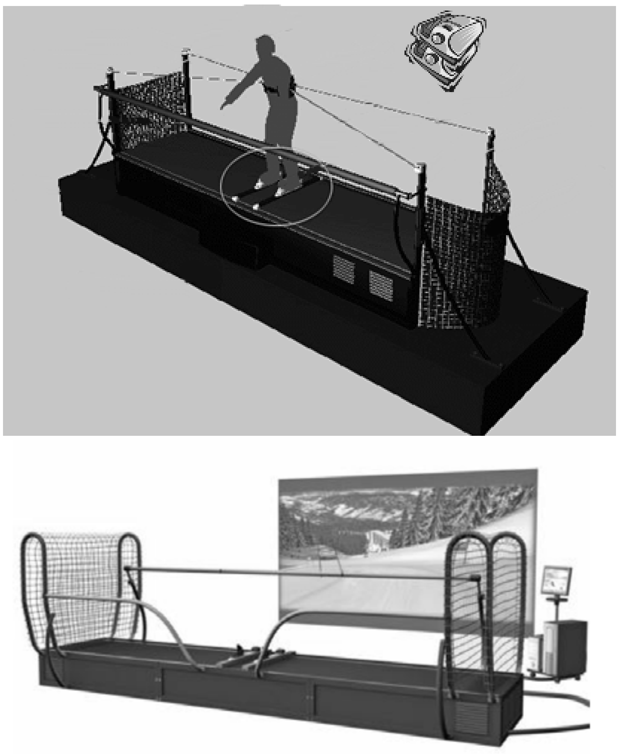

Figure 1: The training system scheme.

The designed system consists of two main parts: the 3D visualization subsystem and the sportsman motion modeling subsystem.

The 3D visualization subsystem creates a stereoimage of a skiing track, surrounding landscape, obstacles, and virtual sportsman (avatar).

The sportsman motion modeling subsystem provides interaction between the sportsman and the training system and computes physical aspects of the process: it receives data about the sportsman position on the training platform, computes his position on the skiing track, computes power loads on the sportsman, and sends this data back to the training system. This subsystem also receives signals from the head tracking system and computes the sportsman head position. After that, it sends the new position and orientation of the camera to the visualization subsystem.

Besides these two main subsystems, there is also the controlling subsystem. This subsystem is connected to the operator display. Through this subsystem, the operator can choose the skiing track and weather conditions, adjust motion parameters, control the strength of power loads, etc. The operator can observe different statistics: the sportsman velocity, his position on the track, the time elapsed from the start, etc.

2. 3D SCENE DESCRIPTION

A 3D skiing track scene consists of the territory surface, the skiing track attributes (flags, nets, ski lift), other objects (trees, houses, spectators), and the landscape in the background. On the base of this 3D scene, the visualization subsystem creates a stereoimage that the sportsman sees when training. The sportsman motion modeling subsystem also uses the 3D scene for computing the sportsman position and orientation on the track and detecting his collisions with virtual obstacles (flags, a net, trees).

We designed the 3D scene description format that can be used by both main subsystems. The terrain surface is described in one file, and the skiing track and 3D objects position in another separate file.

Our systems supports landscapes and surfaces created and described in different ways and applications.

The first way is to create the surface on the base of a regular grid of vertexes. Such surface can be created in 3D editors: 3D Max, Blender, etc. In these 3D editors, we can change the surface by changing positions of separate vertexes and observe the result at once. In this way, we can create surfaces with the most complicated geometry. But the performance suffers greatly because the motion modeling subsystem has to work with the whole set of vertexes.

The second way is to create the surface on the base of an altitude map on the fly. An altitude map is a map image where pixel brightness stands for the corresponding point altitude. The visualization subsystem creates a regular grid of vertexes where one vertex stands for one pixel in the altitude map. The subsystem defines the altitude of a pixel from its brightness and transfers this altitude value to the *Z* coordinate of the corresponding vertex in the grid. In most cases, it is easier to keep one large regular grid and work with it. However, in the case of large landscapes, it is preferably to keep a small grid connected to the camera position and dynamically update the vertexes *Z* coordinates on the camera motion. The altitude map also simplifies the process of computing the sportsman position on the track in the motion modeling subsystem [5].

An altitude map can be created manually in either graphical editor. However, there are dedicated graphical applications such as Terragen or Bryce. Altitude maps of real landscapes made from satellites can be also used.

Instead of processing altitude maps manually, we can use VirtualPlanetBuilder, a special library produced by the OpenSceneGraph developers. VirtualPlanetBuilder transforms the received altitude map into the 3D model with different detail levels. The resulting 3D model appears to be a graph that consists of different parts of the scene with different detail levels. VirtualPlanetBuilder saves this model in the native OpenSceneGraph format. During the training system operation, OpenSceneGraph automatically renders parts of the scene with the corresponding detail levels. This method leverages performance greatly. The motion modeling subsystem uses the IntersectionVisitor class built in OpenSceneGraph. Due to this class, the subsystem receives the altitude value of each point. Using the VirtualPlanetBuilder library is reasonable in the case of very large landscapes.

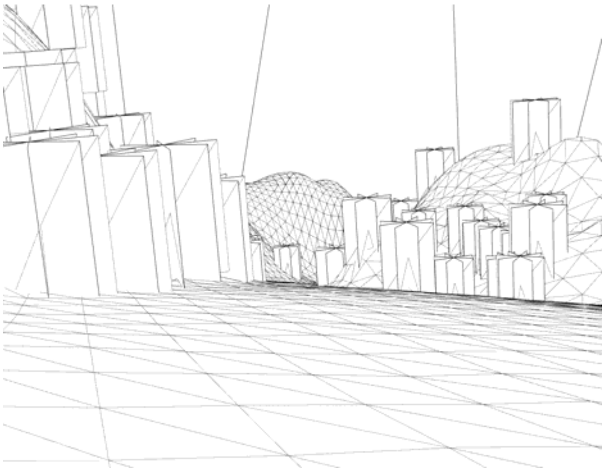

Figure 2: A polygon model.

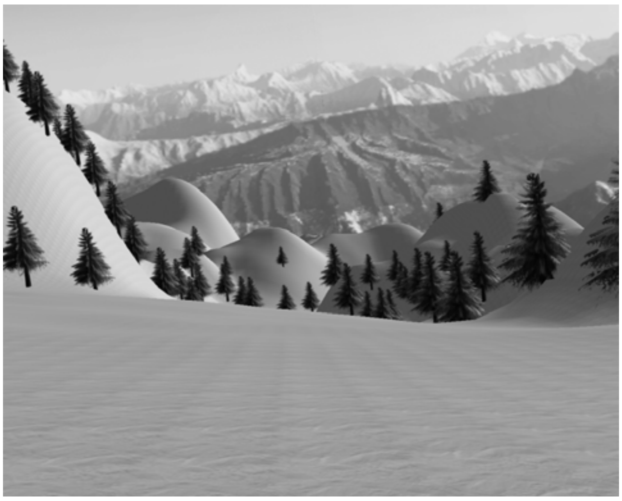

Figure 3: A 3D mountain landscape.

After the surface is created, we should create and describe a skiing track and different 3D objects position. We can use one surface for many different skiing tracks.

The skiing track description is saved as a list of checkpoints in a separate file. Each checkpoint (x,y) is written on a new line. The skiing track and 3D objects preferences are saved in another file.

The system reads these files before loading the scene. Using checkpoints position and the track width, the system calculates the track borders and flags positions. Using the 3D surface description, the system computes the *Z* coordinate of all its points. Then the system puts the 3D models of flags and nets designed in either 3D editor to the calculated coordinates. The system also adds trees, houses, and other 3D objects to the scene. These objects position can be set explicitly or using a list of rules.

For example, for setting trees, we can use two rules. First, the density of trees depends on the area altitude. The higher the area, the less trees should be placed. Second, the trees density cab be defined using a map by analogy with the altitude map. In this “tree” map, the pixel brightness stands for the trees density in this point.

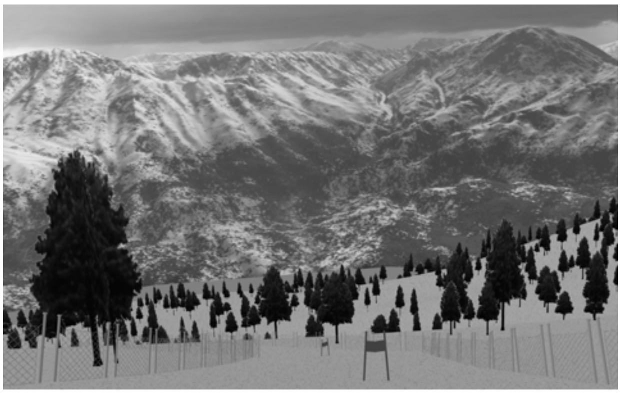

Figure 4: A skiing track with textures and obstacles.

Besides static 3D scenes, we can use dynamically generated endless scenes. In such endless scene, the surface is generated on the flight during the sportsman descent. The surface consists of a sequence of elements. Each element is created from a set pattern by applying noises to it.

3. TRAINING SYSTEM SYNHRONIZATION

The training system is connected to the computer through the special input/output cardboard. The computer receives the training system events signals from the cardboard in a separate flow. The events on the training system appear when the training system carriage changes its position or the inclination angle changes. Using the inclination angle, the current sportsman position on the track, the surface description, and some other parameters, the computer calculates the next sportsman position and the power loads on him. Then the computer sends the power loads values back to the I/O cardboard [6-8].

Before beginning to render the next frame, the visualization subsystem receives the new camera position. Thus, the sportsman experiences the training system force back-coupling and observes his sliding alone the skiing track.

When computing the next sportsman position, the motion modeling subsystem checks for collisions with other objects of the scene: flags, nets, and borders. Depending on the strength and type of a collision, the motion modeling system can either stop the training or send a message to the visualization subsystem to change the crashed obstacle position. In the later case, the visualization subsystem will start the corresponding animation in the next frame (e.g. a flag will begin to bend).

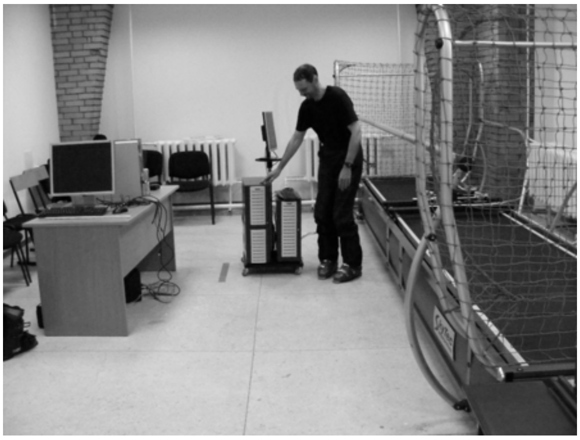

Figure 5: The experimental setup.

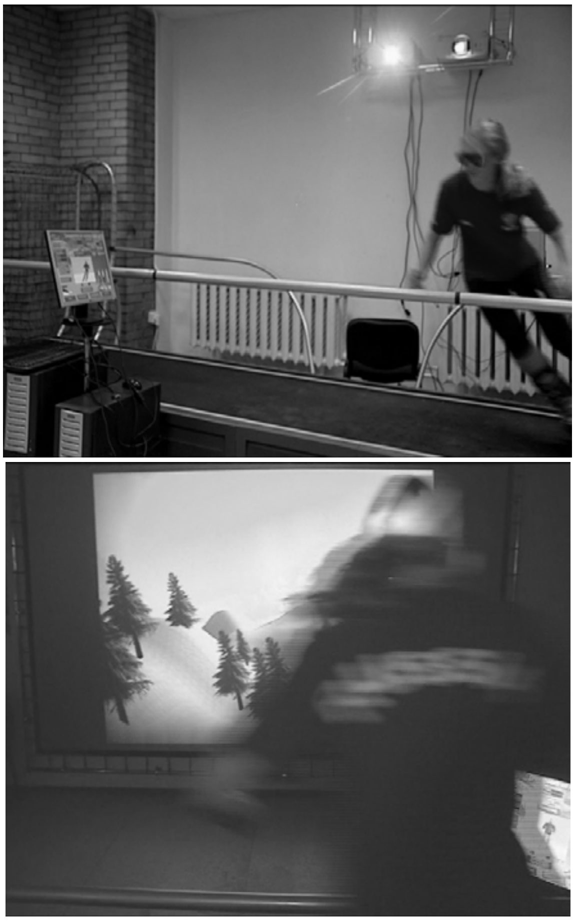

Figure 6: The training process.

The lateral shift and the edge angle are measured, the obtained data is digitized and transmitted to the virtual environment system [1,3,5]. A tracking system can be used to detect the location of the trainee's eyes. The virtual environment system models the skier's trajectory and determines the skier's position in the 3D virtual environment scene on the basis of the data obtained. The main synchronization problem is taking into account physical laws of a skier/snowboarder turning movements. The good analysis of this problem is presented in P. A. Federolf [6]. For the ideal carving turn, it is possible to use the approximation of D. Lind, S. P. Sanders [8]:

$$\tan\phi = \frac{v^2}{gR\cos\alpha} \mp \tan\alpha\cos\beta$$

where ϕ - is a inclination angle, (a tilt angle), v -is a centre mass velocity of a skier, R - is a turn radius, α - is an angle of the slope, β - is a ski traversing angle at a fixed turn point, g - is a gravitational acceleration. A turn radius R at the first approximation will be determined by the edging angle and the interaction of a ski and snow [8]. An inclination angle ϕ differs from edging angle because of the angulation. The basic problem of synchronization is connected with taking into account all foregoing parameters and with definition of the skier position in an induced virtual scene. An individual problem is formation of a signal to a training system according to the position of the skier in an induced virtual scene.

4. CONCLUSION AND RESULTS

In this article, we provided the description of the system that simulates and visualizes the sportsman sliding down a skiing track. The system based on the SkyTec Interactive training system for skiers and snowboarders. This system allows to create different 3D surfaces, scenes, objects, to set skiing tracks, and to stuff them with obstacles. Using the training system events tracking, our system computes the sportsman position on the skiing track and generates power loads on him.

Nowadays, we perform tests on our system with artificial mountain landscapes and skiing tracks. The next stage of our work is to create 3D models of real skiing resorts using stereoscopic topographical survey. Results of work were presented in Russian multimedia [9, 10].

4. REFERENCES

[1] *Aleshin V., Afanasiev V, Klimenko S., et al*. (2004) System for Visualization of Induced Virtual Environment, Proc. 14th International Conference GraphiCon-2004, Moscow, Russia, pp.12-15

[2] *Stanislav Klimenko*, Virtual Planetarium: Educational Application in Virtual Environment, talk at ED-MEDIA 2001 (World Conference on Educational Multimedia, Hypermedia & Telecommunication, June 25-30, 2001, Tampere, Finland)

[3] *Vladimir Aleshin, Stanislav Klimenko, Mikhail Manuilov, Leonid Melnikov,* Alpine Skiing And Snowboarding Training System Using Induced Virtual Environment // Proc. .of the 4th International Congress on Science and Skiing, ICSS 2007, St. Christoph, Austria, publ. 2009, ISBN:978-1-84126-255-0, pp. 137-144, Meyer&Meyer Sport.

[4] Trainer SkyTec Interactive: www.skytec-interactiv.com

[5] *Aleshin V, Klimenko S, Astakhov J, Bobkov A, Volegov D, Borodina M, Kazansky I, Novgorodcev D, Frolov P.* System for Visualization of Induced Virtual Environment fot the Skiing Training System //Proc. Of the Internal Congress "Dialog 2008", ISBN 978-5-88835-020-1, –M., 2008.

[6] *P. A. Federolf*, Finite Element Simulation Of a Carving Snow Ski, PHD Thesis, 2005

[7] *Aleshin V., Danilin V.* (2002). Basics of Alpine Skiing technique, chapter of the book "Alpine Skiing and Snowboarding", pp.35-46

[8] *David Lind, Scott P. Sanders*, The Physics of Skiing, 2004, Springer.

[9] Virtual sport, newspaper "Arguments and facts", № 27, 01.07.2009, p.48

[10] *Andrey Negru, Alpine skiing without leaving home, 10.07.2009, Utro-Russia,* www.utro-russia.ru/news.html?id=44786

About the author

Vladimir Aleshin is a docent at Moscow Institute of Physics and Technology, Department of General and Applied Physics. His contact email is aleshin_vl@mail.ru.

Stanislav Klimenko is a professor at Moscow Institute of Physics and Technology, Department of General and Applied Physics. His contact email is stanislav.klimenko@gmail.com.

Astskhov Juriy is a post-graduate student at Moscow Institute of Physics and Technology, Department of General and Applied Physics.

Bobkov Alexander is a student at Moscow Institute of Physics and Technology, Department of General and Applied Physics. His contact email is archimag@dgap.mipt.ru

Borodina Maria is a student at Moscow Institute of Physics and Technology, Department of General and Applied Physics. Her contact email is mari.borodina@gmail.com

Volegov Dmitriy is a post-graduate student at Moscow Institute of Physics and Technology, Department of General and Applied Physics.

Kazansky Ilia is a post-graduate student at Moscow Institute of Physics and Technology, Department of General and Applied Physics. His contact email is railway_station@mail.ru.

Novgorodcev Dmitry is a post-graduate student at Moscow Institute of Physics and Technology, Department of General and Applied Physics. His contact email is nov-dmitrij@yandex.ru.

Frolov Pavel is a post-graduate student at Moscow Institute of Physics and Technology, Department of General and Applied Physics.