THE CREATING AND PREPARING 3D GRAPHICS MODELS TO USE THEM IN THE REAL-TIME APPLICATION

POKORNÝ, Pav[el]

Abstract: The important part of the modern 3D applications is the visual look. The bad graphics design can depreciate many users. This graphics design especially depends on the quality of the 3D graphics models and their surface - used materials and textures. But the high poly models and high quality textures are very heavy for the computer, whereas is this application running. This paper describes my experience with the possibilities of optimizations of 3D models, which can help to make real time applications with the high quality graphics design.

Key words: computers, graphics, modelling, visualization

1. INTRODUCTION

The modern real-time applications are mostly complicated and large applications with many functions and possibilities. The software developers also implement into their software comfortable user interface and suitable graphics design, which makes the best communication between the application and the user. The very important part of the graphics design is own graphics output. Many of application (like graphics programs or computer games) offer the three dimension visualization. But the best quality of visualization graphics brings much more system and hardware requirements of the computer, on which is this application running. The most difficult situation arises in applications that visualize real world. One example could be a computer game. The actual story takes place in any environment. The virtual environment (world) can be formed by the ocean, mountains or natural landscapes. (Lander, 2003) Next, in this environment, there are exist hundreds or thousands objects (trees, stones, buildings, characters, etc.). And we want to visualize all of these objects and fluently animate in the real-time. The modern computers are very powerful computer, but they are not still capable to compute and render so much of data in the high quality in the real-time. Also we need to seek the ways how to optimize the 3D models and visualization algorithms.

2. 3D MODELS REPREZENTATION

The first problem is the 3D object representation. There are many types of representation, but currently the boundary representation is most used. The borders of objects are most represented by the analytic or by the polygons. (Zara, 1998)

The analytic boundary representation is based on theory of curves and surfaces. The parametric and implicit surfaces are very often used in the sphere of computer graphics. The greatest advantages of this representation are the accuracy and low memory economy. This representation is also often used in the CAD/CAM systems or architecture.

The polygonal boundary representation (fig. 1) is based on polygons. The base polygon is a triangle, but many APIs like OpenGL or Direct3D also supports quadrilaterals or general polygons. (Shreiner et al., 2006) (Walnum, 2006) This representation does not have the advantages like analytic representation, but the polygons are very suitable for visualization (render) algorithms like ray-tracing. The second advantage is the polygonal boundary representation. This is the standard supported by the hardware devices (graphics cards). This means, most of the operations with the polygons are accelerated. Also in the final phase, the analytic boundary representation is transferred into the polygonal boundary representation. (***, 1997)

Fig. 1. The polygonal object representation (wire frame)

3. MODELLING

To create 3D models, we need to choose a suitable graphics program. There are many programs in this area, commercial and non-commercial, which have a greater or lesser number of tools for creating graphics models. The most sophisticated programs of non-commercial sector includes program Blender, with which I have much experiences.

Blender is the free open source 3D content creation suite, available for all major operating systems under the GNU General Public License. Blender has a large number of graphics tools, like modelling, shading, UV unwrapping, imaging and compositing and works with physics and particles, rendering, simple and complex animation. (***, 1998)

Blender contains a huge number of modelling tools for the analytic and polygonal representation. In the field of analytic representation here we can add and modify various 3D objects begins with simple curve or a circle and ending with a complex 3D surface. Manipulation of these objects is simple and it can easily connect individual curves or surfaces. Upon completion of this modelling is the representation usually converted into a polygonal representation, as stated in the previous paragraph.

For the models defined by a polygonal representation is available to an even greater number of modelling tools. In Blender, these objects are generally named as the mesh and being characteristic, that consists of vertices, edges and faces. In the simplest elements can freely manipulate and apply them
After creating 3D models, it is continued with the setting of the material properties, texturing (there is usually used UV unwrapping for precise application of textures), and possibly animation. (Pokorný, 2009)

5. IMPLEMENTING INTO THE APPLICATIONS

Very important factor in the whole process of rendering complex 3D scenes in real-time is not only to optimize the 3D model itself and its texture, but also the appropriate organization of the entire program, especially the part, that deals with the visualization of the whole scene. The biggest impacts have the rendering algorithms and the administration of created 3D graphics models including textures. The entire organization and connecting between algorithms are very complicated, and everything depends on the skills and experience and the programmers, who created the particular application. In some cases, we can help out and finished the existing libraries and engines that may have already programmed most of the used algorithms.

An example of such an engine can be Irrlicht, with which I have very good experience. Irrlicht is a cross-platform high performance real-time 3D engine written in C++. It features a powerful high level API for creating complete 3D and 2D applications such as games or scientific visualizations. It comes with an excellent documentation and integrates all state-of-the-art features for visual representation such as dynamic shadows, particle systems, character animation and collision detection. All this is accessible through a well designed C++ interface, which is extremely easy to use. (***, 2003)

6. CONCLUSION

The modern real-time applications with the 3D graphics visualization can be more exacting for computers, on which they are running. By default, the frequent requirement of very good graphics needs more system and hardware computer resources. This is, why we are looking for the ways, how to get the best graphics quality in the real-time together with the minimal load of the computer. This paper describes my own experiences, possibilities, how to do it in two phases - the 3D models creating and exporting them into the own applications. My experience shows, that the software optimization is mostly limited by the possibilities of computer hardware.

7. ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education of the Czech Republic in the range of research projects No. MSM 708352102.

8. REFERENCES