3D Modeling Using Stereo Projection

TOMAS POPEK, MARTIN NEMEC, MICHAL KRUMNIKL, RADOSLAV FASUGA
VSB-Technical University of Ostrava
Department of Computer Science
17. Listopadu 15/2172 70030 Ostrava
Czech Republic
{pop101, martin.nemec, michal.krumnikl, radoslav.fasuga}@vsb.cz

Abstract: The aim of this paper is to describe a modification of an open source tool Blender, and its improvements to support the primary types of stereoscopic projections. The proposed stereoscopic mode can be used for design and modeling. Even though the game engine, an integral part of Blender, already contains a support for stereoscopic projection, it is limited to display only the complete applications but does not support the stereoscopic mode in the design phase of three-dimensional projects or scenes. The introduction of the stereoscopic projection in Blender improves the user experience during the scene modeling.

Key–Words: Blender, Stereo projection, Anaglyph, Quad-Buffer, Modeling

1 Introduction

The advancements in information technology are providing new opportunities for the development of 3D technologies. The 3D stereoscopic projection systems are commonly used in cinemas for the presentation of popular movies. The similar 3D projections can also be used while working on computers (on condition that this technology is supported). Nowadays, there are even modern television sets that allow the stereoscopic viewing of certain movies and channels.

To achieve the stereoscopic depth perception, a particular image projection is used with individual images being displayed for each eye separately in a certain horizontal displacement. If we think about how human depth perception works, we realize that it is mainly due to the fact that our eyes are set apart and space is viewed from two, slightly dissimilar viewpoints, which result in depth perception [8]. There are many ways how to project images for each eye independently, but generally they can be divided into passive and active methods [7].

In case of the passive methods, a visual information is displayed for both, the right and the left eye, simultaneously. The projected images can be separated by several methods. An exemplar passive method is an anaglyph, which achieves stereoscopic projection through setting a color mask for each eye. Another passive method is usage of polarized glasses. This method separates images using different polarized filters.

With the active method, images are usually displayed with double frequency and separated by Liquid Crystal Shutter Glasses. Active glasses can display individual images for the particular eye, after being synchronized with a projector or a television set.

However, the 3D stereoscopic perception occasionally causes headache and eye strain due to a fixed projection plane in the 3D projection. There are other perspective principles like digital holography. Holographic displays generates generally an optical fields so all the users can have a correct perception according to their spatial position [5].

2 Blender

Blender is an open source application for a 3D modeling, animation production and scene or model rendering. It uses OpenGL library for displaying its interface. By using OpenGL and other modules, Blender is not bound to any particular platform or operating system [1]. One of its part is called a Game Engine, this engine allows the creation of various interactive applications, such as games, virtual tours, etc. The Game Engine, as a separate part of Blender, has already the stereoscopic projection built-in, although it can only be used by activating the "game mode". The Game Engine provides several types of stereo rendering.

The idea of extending Blender is not new, there is already an experimental, altered version of an older release of Blender, which was created by the University of West Bohemia [6]. The alteration was an extension of the classic Blender application. Its aim was to create a projection which could be viewed in the 3D with a stereoscopic laptop Sharp. The stereoscopic effect itself was accomplished by using interlaced lines, as well as an adjustment of the display buffer, which projects the rendered image on the monitor. However, due to the parallax barrier used in the Sharp AD3L, user's spatial position was very limited, as the cross-talk grows significantly if the user moves from the optimal position. Also the screen resolution was reduced to a half due to the used principle [4]. The display buffer itself was adjusted in such way that two offset images were produced, and were merged using a shader. Merged image was projected on the monitor as a final image. When a newer version of Blender was released, this altered version was no longer usable, since the newer version significantly differs in source codes.

The main section of this paper presents Blender's extension enabling the view of 3D modeling segments (3D viewport) using the stereo projection. We are currently working with the latest SVN version of Blender, i.e. 2.6.2.0, which has, compared to the previous version (2.4.0), undergone a major code and structure reconstruction. The changes have positive effects on its functionality and significantly improved the user interface and program behavior.

As mentioned before, Blender does not contain stereo projection support as such, but the Game Engine, one of its part, does. The changes we have made in Blender were partly inspired by the stereoscopic projection built in the Game Engine.

3 Stereoscopic Projection

As opposed to the Game Engine, our goal is to create a stereoscopic projection in a way that would enable the usage of the stereo projection in the editor during the designing and modeling of a 3D scene. The main issues we have faced include a lack of consistency between the 3D scene, 2D control panels, cursor and other elements in the scene.

The fundamental idea of acquiring horizontally shifted images lies in the work with cameras. By using slightly adjusted camera, it is possible to obtain a stereopair (separated images for the right and the left eye) [2]. These images are used for creating an anaglyph, or for any other type of interlaced method. If the Quad-Buffer method is set to be used, it must be supported by the graphic card. In our case, the NVIDIA Quadro FX 2000 was used for the testing purposes. However, a device that facilitates viewing also has to be present, such as the 3D LCD monitor BenQ XL2410T. This monitor successively displays images for the right and the left eye at a total frequency of 120 Hz. Therefore, each eye views the image at 60 Hz. For further and wider testing additional hardware was used. The quad buffered projection was tested even on a projector. The tests were performed on the stereoscopic projector DQ 3120-X2.

4 Practical Implementation

As said before, all essential parts of the scene redrawing are based on OpenGL calls. The part of the OpenGL source code was modified to enable the stereoscopic projection [3]. As a part of practical implementation, it was necessary to analyze the source code and search for the parts where the drawing of the individual objects of the scene takes place. The term "scene objects" refers to the 3D models, as well as various auxiliary entities within the scene. Among these entities is the grid, frame, 3D cursor, camera, and so on. All these objects are parts of the scene and have to be converted into the stereo projection, in order to eliminate the eventuality that one object without the stereo effect would appear in front of another object with the stereo effect enabled. If such situation happened, an unpleasant optical phenomenon would occur: A spatial object, which seems to be closer to the viewer, is covered by another object, which seems to be farther, behind it. There are several physiological and psychological cues we need to observe. The rendering of all scene objects is located in the one unified class. This class contains various help methods and two functions that are used for the drawing. One of them is used for drawing objects mentioned above; the second one is used for drawing texts and manipulators which are located within the scene (window name, frame index, Blender cursor etc.).

As mentioned in the introduction, the perception of space and depth is caused by a small horizontal offset of our eyes and consequently of the images which each eye perceives. It was therefore necessary to adjust the individual drawing functions so that the result would be two horizontally shifted pictures which would be projected for the designated eye.

The main drawing function is composed of several parts. We are most interested in the matrix preparation part and the drawing part. During the preparation part, we need to copy the prepared matrix, so that we can achieve desired translation in the drawing part. The translation span is defined via a keyboard shortcut and users can adjust it according their individual needs, with regards to the distance between the user and the projection area.

There are two main changes in the drawing part of Blender. The first change is a creation of two methods that can adjust projection matrices. These methods serve to achieve horizontal translation and use a transformation matrix with an added offset which alters its values in such a way that all the objects are translated while the perspective is preserved. The projection of the entire scene takes place after this translation. The resulting objects will be slightly moved for the human eye.

The second change is the creation of the main drawing loop which can redraw objects for each eye with different transformation matrices. During the iterating through the loop, it is determined which translations are to be used (for which eye). In this part of the code, it is also decided which stereo method is used and based on that what modifications should be applied. After rendering, it is necessary to reset all changed values (color mask, buffer etc.) into their original states, so to not influence further rendering.

4.1 Rendering Using the Anaglyph Method

All rendering processes including this one are based on the aforementioned principle of the stereo projection. After locating all rendering functions in the source code, it was necessary to adjust them, so that the result would not be the rendering of a single object but the rendering of two mutually translated objects, which are horizontally shifted from each other and color-encoded.

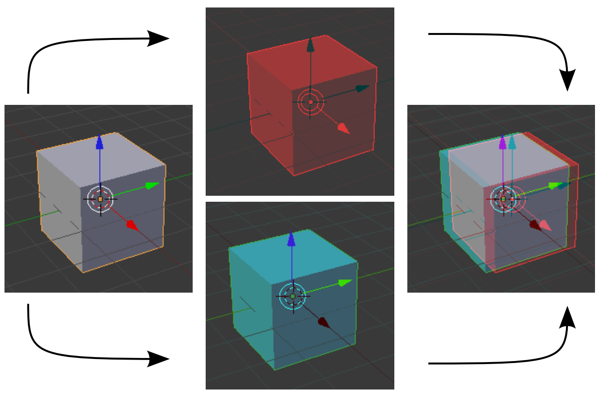

Figure 1: Processing anaglyph method

The anaglyph color effect (red/cyan) is easily achieved with the color mask *glColorMask(...)*, which allows to activate or deactivate certain color channels in the displaying buffer. A different mask is used for each eye. For the left eye, we use a red-encoded mask, therefore information about green and blue channels is left out. The filter for the right eye is set so that the red color is left out and the resulting image is in the green and blue spectrum (cyan). The color mask is set this way for every object in the scene.

In this case it is necessary to display both color-encoded images simultaneously, so both of them must be combined to form a single final image. This is accomplished by the buffer not being erased after the scene rendering for the left eye. The information for the right eye is rendered into it. After the rendering, it is necessary to reset the color mask back into its original state, so as not to influence further rendering.

4.2 The Quad-Buffer method display

When compared to the other methods, the main difference is the use of two double-buffers, which enable the rendering for both eyes into each buffer separately. Thanks to the fact that we can render images for each eye into two separate double-buffers without the need for blending them, we can achieve full resolution (based on the used hardware equipment).

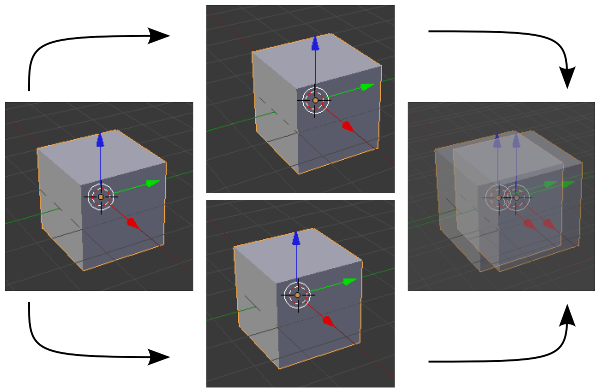

Figure 2: Processing quad-buffered method

Instead of rendering all of the content into the buffer with the command *glDrawBuffer(GL_BACK)*, we need to render two images separately with *glDrawBuffer(GL_BACK_LEFT)* for the left eye and *glDrawBuffer(GL_BACK_RIGHT)* for the right eye.

There was a little adjustment in the code, where viewport window was declared. It was necessary to set the stereo support On. Of course, this method can be only used if we have necessary hardware such as the graphic card with Quad-Buffer support and the projection device that can be used with the frequency of 120 Hz.

The most interesting and valuable attribute of this method is, as said before, possibility to achieve full resolution of the projection screen. This quality provides us very comfortable conditions for the work.

4.3 Interlaced display method

This method is using an interlaced mask which is made on a special plane laid over the screen. This plane cannot be directly seen by the eye, but it overlaps rendered images and makes specific parts invisible. We can choose from two types of interlaced drawing – the horizontal and vertical interlacing. The

ISBN: 978-1-61804-109-8

principle of the horizontal or vertical interlacing is the same with the only difference in the stipple mask.

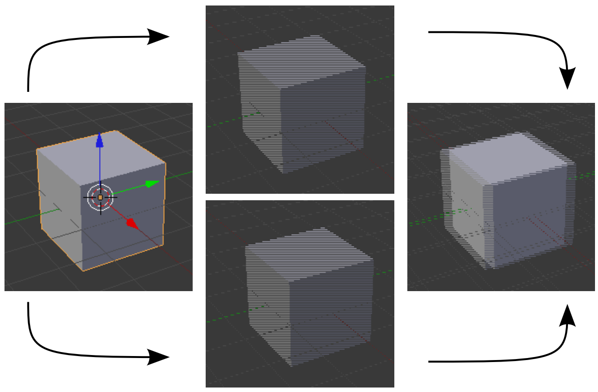

Figure 3: Processing interlaced method

We used a *glPolygonStipple(...)* command to transform the overlapping plane to the interlace mask. The mask was turned invisible by setting its color and alpha channels to zero using OpenGL command *glColorMask(...)*. Depending on the target eye, we use one or the other interlace mask to switch currently shown lines.

This interlaced method has got the same positives and negatives as its older equivalent from the version made by the University of West Bohemia. The biggest disadvantage of this method is the lack of the possibility of achieving full resolution of the projection screen.

5 Testing

At the moment, the tests of the practical use of the stereo projection in Blender are taking place. In these tests, we are trying to find out how different users react to the changes we have made. Users are mainly university students divided into two groups, one having no prior experience with Blender, and the second, where students have been modeling with Blender for many years. We have also included an analysis of further necessary modifications, which would increase the user-friendliness of the application, provide a problem-free running and extend application features.

Test subjects (users) were asked to answer short questionnaire after the work with Blender. Subjects were asked about their skills, former experience with the stereo projection, environment and used hardware. The main part of the questionnaire was devoted to questions about how subjects managed to work with the stereo projection. For better comparison, subjects were also asked about the difficulties and their satisfaction with the original Blender.

Subjects were divided into two groups. One group was testing the stereo projection using the anaglyph method. The second group was using the Quad-buffered method. From the gathered results and interviews we were able to identify some specific problems and fix them. We have to mentioned that these interviews were very helpful. Thanks to them we have identified topics and ideas we can implement to the future versions. For example, we noticed that using the anaglyph method is more eye-friendly if subjects are using only wireframe models. There were also situations when users were zooming very close to the objects. In situations like this, the zooming objects were producing stereopairs with higher disparity which lead to the ghosting effect and eye divergency. This was improved by moving near clipping plane forward to avoid such unsuitable situations.

Other finding was that user satisfaction with our modification depends on used hardware. Most test subjects preferred monitor than projector because of better resolution and contrast.

When using the anaglyph method most of the subjects were complaining about ghosting around objects or color distortion. These problems with the anaglyph method were solved by one of the test subjects which activated one of the Blender displaying modes - Wireframe. When using this method, every object on the scene is drawn as the wireframe model without surfaces and therefore without apparent ghosting. In contrast, the Quad-Buffer method was received very well. The only complains concerning this method were because of the difficulties with focusing on the proper depth or occasionally eye pain due to the unusual frequency rate. The strength of this method is mostly in obtaining the depth in the scene and better recognizing of the spatial shapes.

Most of the subjects were complaining about the lack of the stereo-cursor. There was an unpleasant effect with the cursor seen twice when users tried to choose certain vertex, line or polygon. This was caused because the non-stereo mouse cursor lies on the projection screen whereas our object of interest is in the front (or behind) the screen. This problem was easily solved by hidingthe mouse cursor and redrawing it only for one eye (e.g. into the right back buffer using quad-buffered method). Because of this adjustment, there is no possibility for seeing the cursor twice.

To increase user-friendliness of our application, we have created a simple GUI panel. It was necessary to create an interface through which the user can control and adjust available stereo properties. The GUI itself was created using the scripting language

ISBN: 978-1-61804-109-8

Python. Blender is using Python for definition its own interface and even some functions and scripts are written in this language. Resulting GUI contains several control elements and confirmation buttons. User can choose the type of stereoscopic projection and define the offset between the images in the stereopair. There is also possible to activate stereo-cursor for interlaced and quad-buffered method. It should be noted that the stereoscopy itself is fully operational without this panel. The interface is created only as an extension. Alternatively, it is also possible to control stereoscopic properties via keyboard shortcuts and startup parameters with no need of using additional settings in panel.

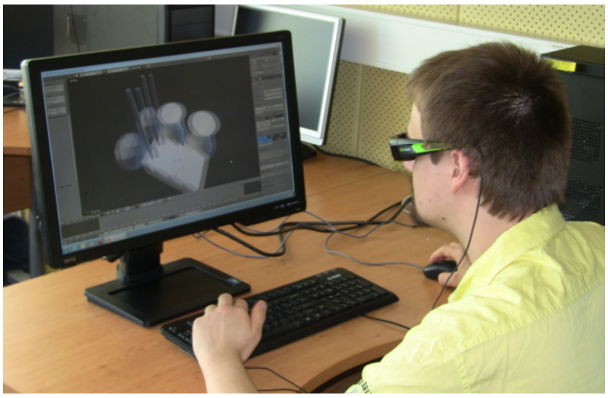

Figure 4: Practical testing

6 Conclusion

Recent deployment of the stereoscopic projection in computer graphics, particularly in modeling, brings up new possibilities for more flexible applications. With the implementation of stereo imaging in the 3D design and modeling tools, artists and users have better opportunities to view the modeled scene from a completely new perspective.

This paper has shown that the usage of the stereoscopy is not only for the projection purposes, but it also provides help for graphic designers. Almost all of the people interviewed during our survey were able to see the difference in projected depth of the scene. Only a small part of users were not able to work with our modified Blender. We suppose, this was mainly caused by the fact that by using the stereoscopic projection we have changed their internal projection standards and habits. Currently there are several changes being developed, e.g. a stereo-cursor. We are trying to address users needs, and problems raised from the practical testing.

The next step in our work is to test these options in a practical environment, as well as better analyze users reactions. Even though there is a certain percentage of people who do not have developed spatial perception and therefore will never make use of the final 3D stereoscopic projection, the others might regard stereoscopic projection as a very interesting extension and an opportunity for the 3D modeling and much more.

Based on the users feedback, we have calculated over thirteen percent increase in the satisfaction with the active quad buffered stereo projection method among young users (with no significant experiences in this field).

This work and data gathered during the experiments could be the starting point for more sophisticated approach to this topic, e.g. by using digital holography [5]. We have shown that working and modeling in the stereoscopic environment is not only possible, but even relatively easy.

Acknowledgements: This work was partially supported by the SGS in VSB Technical University of Ostrava, Czech Republic, under the grant No. SP2012/58. Main impetus for this work came from the University of West Bohemia and their first version of Blender 3D.

References:

[1] Blender, *Blender*, blender.org. WWW: <http://www.blender.org/>

[2] P. Bourke, *Stereographics*, paul-bourke.net, 1999.

[3] A. Edward, *Interactive Computer Graphics*, United States of America : Pearson Education, 2009.

[4] M. Halle, *Autostereoscopic Displays and Computer Graphics*, Massachusetts, 1997.

[5] V. Skala, *Holography, Stereoscopy and Blender 3D*, submitted for publications, 2012.

[6] V. Skala, J. Stefany and V. Bystricky et al., *Blender 3D Modelling - stereoscopic version*, University of West Bohemia, 2010. WWW: <http://www.kiv.zcu.cz/cz/vyzkum/software/>

[7] Stereographics Corporation, *StereoGraphics Developers Handbook*, StereoGraphics Corporation, 1997.

[8] Uppsala Universitet, *Optics of the eye*, Uppsala University, 2004.