Eye-Catcher : Real-time 2D/3D Mixed Contents Display System

Jin-Wook Chang, Kyoung-Il Lee, Tae-Soo Park*
Digital Display Research Lab., LG Electronics, 16 Woomyeon-Dong, Seocho-Gu, Seoul 137-724, Korea
TEL:82-2-526-4791, e-mail: tyrann7@lge.com.

Keywords : ***Real-time 2D/3D mixing, True3D effect, True3D content creation, interactive True3D content***

Abstract

In this paper, we propose a practical method for displaying 2D/True3D mixed contents in real-time. Many companies released their 3D display recently, but the costs of producing True3D contents are still very expensive. Since there are already a lot of 2D contents in the world and it is more effective to mix True3D objects into the 2D contents than making True3D contents directly, people became interested in mixing 2D/True3D contents. Moreover, real-time 2D/True3D mixing is helpful for 3D displays because the scenario of the contents can be easily changed on playback-time by adjusting the 3D effects and the motion of the True3D object interactively. In our system, True3D objects are rendered into multiple view-point images, which are composed with 2D contents by using depth information, and then they are multiplexed with pre-generated view masks. All the processes are performed on a graphics processor. We were still able to play a 2D/True3D mixed contents with Full HD resolution in real-time using a normal graphics processor.

1. Introduction

Recently, the 3D display technology has been developing drastically and many companies have released their products for various applications such as advertisement, games, the medical field, and others. However, one of the most serious problems of spreading 3D display to the market is a lack of True3D content. Until now, many ways of generating True3D content has been developed, such as True3D scene rendering by 3D CG, capturing the real scene by synchronized multiple cameras and 2D-to-3D conversion. But, there are still not enough True3D contents for use and 2D-to-True3D conversion techniques are not sufficiently satisfactory yet. Moreover, making new True3D contents is very expensive because multi-view type 3D displays need many different view-point images. For example, LG's 3D display needs 10~25 view-point images for displaying full True3D contents. On the other hand, since there are already a lot of 2D contents in the world, many people became interested in using 2D contents on 3D display by mixing them with True3D objects, which is easier than making True3D contents directly. If 2D/True3D mixed rendering is performed in real-time, not only can a user adjust the 3D effects of the True3D object on playback-time, but he can also change the scenario of the contents easily.

In this research, we developed a novel method for mixing 2D/True3D in real-time to help make 2D/True3D contents more easily.

2. Real-time 2D/True3D Mixed Contents Display System

As shown in Figure 1, 2D/True3D mixed contents display system is composed of four main parts: the 2D movie library, the 3D data library, the 2D/True3D mixing renderer, and the 3D display.

The 2D movie library includes various 2D movies and images which were already made by someone. The 3D data library also includes plenty of 3D objects with its own animation. The 3D object's modeling and animation data can be easily and inexpensively made from commercial CG tools like 3DSMax™, Maya™, Softimage™, and others. The 2D movie and 3D objects can be selected automatically by a pre-defined program schedule or interactively by the user. These two libraries don't have to be at the same machine with the other parts. In other words, they can be at a remote machine and accessed through a network so that contents in them can be transferred at a necessary time.

In the real-time 2D/True3D mixing renderer, the selected 2D movie and 3D object will be rendered into True3D multiplexed images, which can be directly represented on 3D display. The format of the True3D multiplexed image can be changed depending on the stereoscopic method of the 3D display. Any type of 3D display can be adapted into this system because the above mentioned renderer can render out the multiplexed image appropriate to the 3D display.

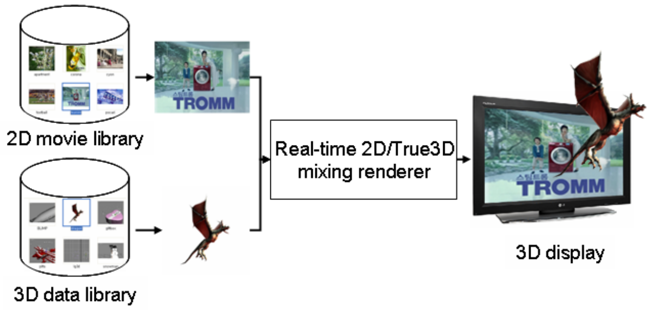

Fig. 1. System Overview.

Figure.2 shows the structure of the 2D/True3D mixing renderer. Real-time 2D/True3D mixing renderer is composed of three sub-parts: a multi-view renderer, a multi-view blender, and a multiplexer.

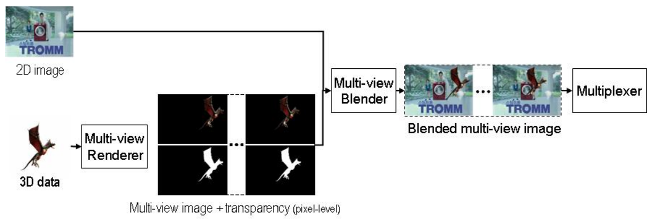

Fig. 2. 2D/True3D Mixing Renderer.

The multi-view renderer renders the 3D object into a multi-view image with transparency information. A multi-view image is a set of perspective view images captured by each virtual camera, which is located at evenly spaced position in the virtual world. The perceived depth of the 3D object is determined by the value of the camera separation, which means the distance between the virtual cameras, and convergence distance, which means the distance from virtual camera to virtual image plane. The values of the two parameters can be changed in playback-time automatically by a pre-defined schedule or interactively by the user. The more separation and convergence distance, the more a popping-up effect of the 3D object is generated. The transparency information is an image which indicates that a corresponding pixel is opaque or transparent. It can be easily calculated from the depth information, which is generated when rendering a 3D object. If the depth value of a pixel is smaller than zero, which means the depth value of the 2D image plane, its transparency is opaque. Otherwise, it should be fully transparent. This transparency information will be used for blending with the 2D image. The transparency value can be multi-level to represent a translucent 3D object.

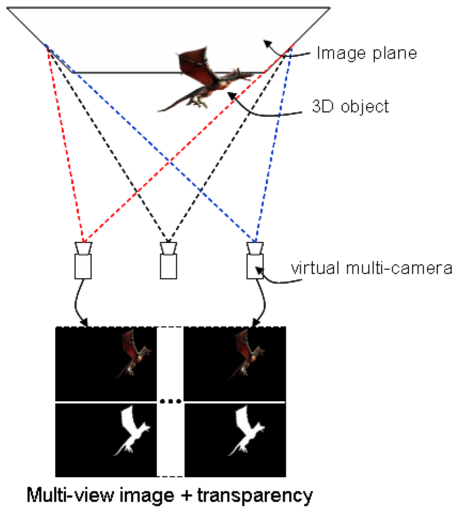

Fig. 3. Multi-view Renderer.

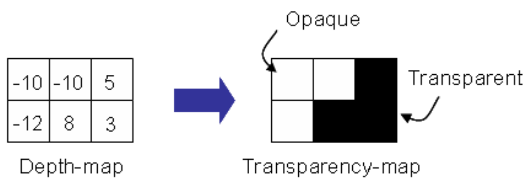

Fig. 4. Calculating Transparency from depth.

The multi-view blender blends each view images of the 3D object individually with 2D images using the transparency information generated by the multi-view renderer. A pixel in the output image plane will be filled with the value, which is calculated by blending the corresponding pixel's color value in 2D image and 3D object's view image. If the transparency is opaque, the value will come from the 3D object's view image, if not then from the 2D image. In case of having multi-level transparency values, two color values will be alpha-blended with the transparency value.

Fig. 5. Blending.

In the multiplexer, the blended multi-view images are merged into one unit of the frame, which should be displayed at the same time in a barrier type 3D display or one after another in a time-sequential type of 3D display. The multiplexing method will be decided by the type of 3D display's representation method. In case of the barrier type 3D display, the multi-view images are merged into one image by sampling from each view images using a pre-defined view addressing table. Fig. 6 shows the example of multiplexing for the barrier-type 3D display. View addressing table contains the view index, from which each pixel in the output multiplexed image should be sampled, and it is pre-defined according to the 3D display.

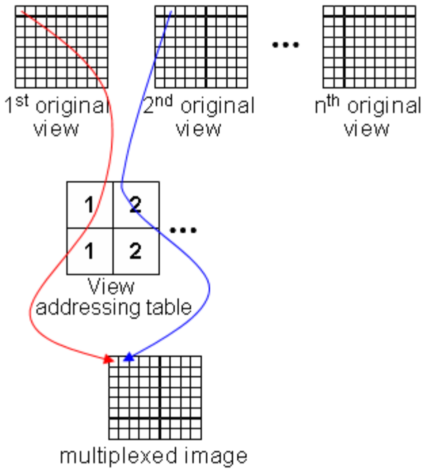

Fig. 6. Example of Multiplexing.

3. Implementation and Results

The proposed system was implemented in a PC environment with a Pentium IV 3.0 GHz, 2GB RAM, and Nvidia Geforce 7800GT. Our implementation of our system was based especially on the DirectShow that uses a filter diagram for representing the 2D movie streams. Moreover we implemented real-time 2D/True3D mixing rendering functions as a plug-in to the new version of Video Mixing Renderer (VMR9) because we can use Direct3D function for getting GPU acceleration in the VMR9. For the display, we used a 42" LG 3D monitor, which is an auto-stereoscopic type and has Full HD resolution.

Depth effect was implemented by modifying the view and projection matrix of the reference virtual camera. The modification could be easily done in real-time due to simple multiplication of the matrices in the Direct3D engine. Rendering a True3D object and blending it with the 2D image was implemented by using a Z-buffer test function in the Direct3D engine. Multiplexing was also implemented by pre-generating the view masks and blending each view image with them through multi-texturing functions and fragment shaders.

Figure 7 shows 2D/True3D mixed contents that are playing on the 42" LG 3D monitor. We have played a 2D/True3D mixed content in Full HD resolution at a speed over 25Hz.

Fig. 7. 42" LG 3D monitor playing 2D/True3D mixed contents.

4. Conclusion

Playing 2D/True3D mixed contents was possible in real-time due to the high performance of modern graphics processors. We found that the performance of the normal graphics processors have no problem in rendering 2D/True3D mixed contents in real-time. Using an Nvidia Geforce 7800GT and 3.0GHz CPU, we have played a 2D/True3D mixed content on a 42" LG 3D monitor in Full HD resolution at a speed over 25Hz.

5. Summary

This paper addresses real-time 2D/True3D mixed contents display systems. Due to the rapidly developed graphics hardware, 2D/True3D mixed rendering is possible in real-time. Displaying

2D/True3D mixed contents in real-time will make 3D displays more popular because using this scheme is more productive for 3D displays than making True3D contents directly. Also, when applied to digital signage, our technology seems to be effective in catching the eyes of passersby.

6. References

1. G. Wolberg. "Digital Image Warping. ", *IEEE Computer Society Press*, 1990..
2. T. Okoshi, "Three-dimensional imaging techniques.", *Academic Press*, NY, 1976.
3. David F. McAllister, Ed. "Stereo Computer Graphics and Other True 3D Technologies.", *Princeton U. Press*, Princeton, NJ, 1993.
4. J. Shade, S. Gortler, L.-W. He, and R. Szeliski. "Layered depth images. In *Computer Graphics (SIGGRAPH'98) Proceedings*, pp. 231-242, Orlando, July 1998. ACM SIGGRAPH.
5. T. Wong, P. Heng, S. Or, and W. Ng. Image-based rendering with controllable illumination. In *Proceedings of the 8-th Eurographics Workshop on Rendering*. pp. 13-22, St. Etienne, France, June 1997.
6. S. B. Kang, R. Szeliski, and P. Anandan. The geometry-image representation tradeoff for rendering. In *International Conference on Image Processing*, Vancouver, Canada, Sept. 2000.