Development of a Software Renderer for utilizing 3D Contents on a 2D-based Mobile System

Sungkwan Kang[1], Joonseub Cha[2], Jimin Lee[1] and Jongan Park[1],

[1] Dept. of Information & Communications Engineering, Chosun University,
309 Pilmun-daero Dong-gu Gwangju, South Korea [2]
Dept. of Internet Contents, Honam University,
417 Eodeung-daero Gwangsan-gu Gwangju, South Korea
{Sungkwan Kang, Joonseub Cha, Jimin Lee, Jongan Park, arome}@chosun.ac.kr

Abstract. This paper presents a software renderer which makes 3D contents available on a 2D-based mobile system. The developed 3D renderer is composed of many software libraries, which is composed of a scene section, a math section, and a base. In this system, the 3D model data can be used without changes to the other settings, the environment, or the data, and the developed 3D contents can be used in 2D mobile systems the same way as existing 2D contents.

Keywords: Renderer, 3D contents, 2D mobile system, Heterogeneous interconnection

1 Realization of a 2D based 3D Renderer

For displaying 3D contents, mobile systems cannot work smoothly because of differences and limitations in systems in comparison with PC based systems, and mobile systems usually function only through the use of 3D hardware chipsets and expensive 3D software engines. Therefore, 3D engine technology which can support 3D contents in cheap terminals and work with contents in other systems such as mobile terminals or PCs is necessary. Also, along with the decreased differences in mobile platforms, increasing demand for 3D mobile games, the interconnection of online games with 3D contents and a more competitive market environment due to the expansion of mobile markets and the convergence of platforms, 3D contents has become an essential issue for future mobile markets.[1]-[5].

This study realized a 3D software renderer which can utilize 3D contents on a 2D based mobile systems in order to overcome the problem of having a 3D mobile system to provide 3D contents in systems such as mobile phones and MP3 players. Figure 1 shows a diagram of the renderer developed. It is a library-type application which is composed of the following: The scene section which supports 3D composition, animation, and texture; The math section which supports math functions and geometric transformations; and The Base which supports sound, images, text, and communication.

The scene section is composed of a Mesh to support 3D rendering, Texture to support surface treatments, a Camera to support changes in and varying angles of scenes, Bones to support bone composition, Animation to support animation functions, and Materials to support surface textures. The math section supports Matrices, Vectors, Scaling to support size changes, Rotation to support turning, and Translations to support transformations as well as other math functions.
The base is composed of Sound support, Image support for image rendering, Fonts to support text, a File section to support file input and output, a Memory section to support memory functions and a Network section to support communication.
It called on the library developed for the mobile platform to realize the 3D contents and the 3D contents realized was rendered on a 2D mobile system.

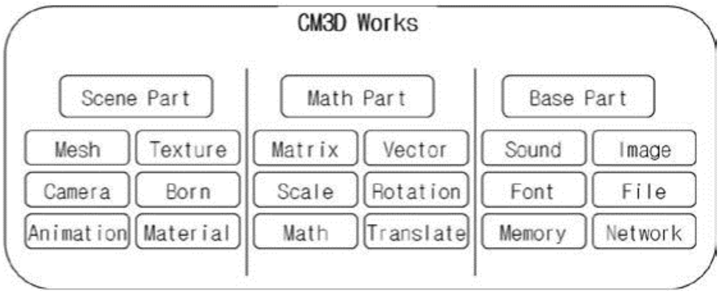

Fig. 1. Composition of mobile 3D renderer

Figure 2 is the diagram of a view flow of a mesh object in the 3D renderer which supports 3D contents in a 2D mobile system. It is composed of a mesh section, texture control, and their integration. For Mesh rendering, when the mesh file of a 3D object was loaded into to the buffer, a polygon list was stored in the vertex buffer, and the stored polygons were indexed. Then, the UV list was stored in the UV buffer and the contents was indexed and sorted.
For texture control, when the texture file of a 3D object was loaded into the buffer, it was stored in the image buffer and the texture list was indexed. The mesh and texture integration worked to connect the texture list which corresponded to the UV list. Through a projection matrix, a view matrix, and world calculation, a texture image was given to each polygon of the 3D objects processed and transformed into a 3D image. After each process was done for the whole polygon of 3D objects, each image was stored in the screen buffer.
Figure 3 is the diagram to manage bone objects in making animations, which is a central part of the 3D renderer. It is composed of a bone structure list, a mesh list, and their integration for animation.

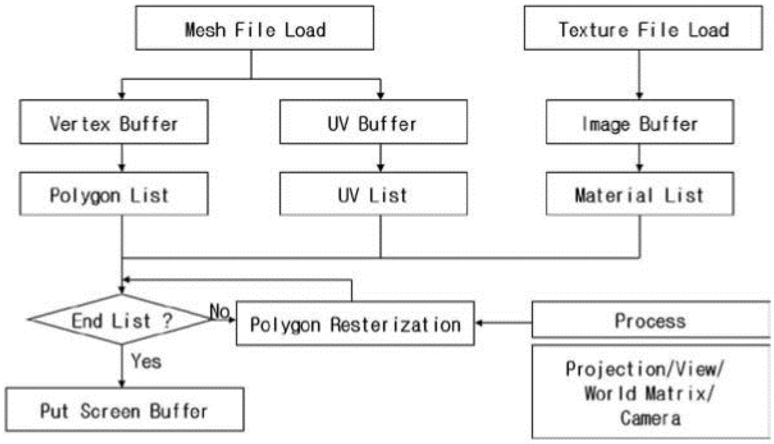

Fig. 2. View flow of a mesh object

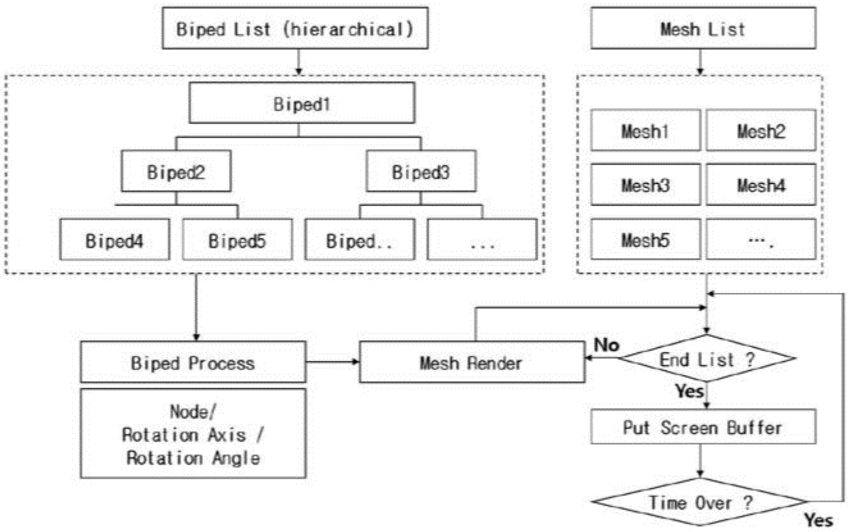

Fig. 3. Control of bone objects

Figure 4 shows how the 3D rendering developed in this study was applied to making and controlling real 3D contents. In Step 1, the same method was used as that in making existing 3D contents. In Step 2, 3D contents made was embedded into a 2D/3D terminal. In Step 3, users executed the corresponding contents in 2D or 3D systems.

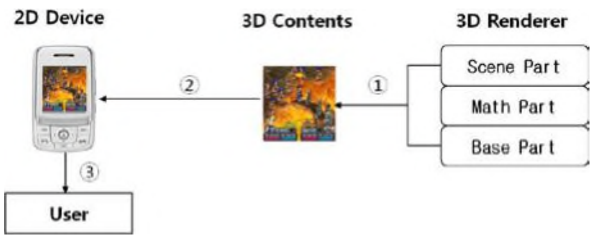

Fig. 4. Application to real contents production and execution

The developed 3D renderer has the following qualities and advantages in comparison with previous engine technologies: First, it is a pure software rendering engine for easy programming work, for which the API was realized into the standard OpenGL. Second, because of the extendibility of the application development, other engines support only full screen mode, but this application can adjust the size of the 3D area and location according to the use of the application as well as full screen mode. Also, through adjustment of the 3D areas, memory control and speed can be improved.

2 Conclusion

This study developed a software renderer which makes 3D contents available on a 2D-based mobile system. The developed 3D renderer was composed of about 200 software libraries and each library was composed of a scene section, a math section, and a base. In this system, the 3D model data can be used without changes to the other settings, the environment, or the data, and the developed 3D contents can be used in 2D mobile systems the same way as existing 2D contents.

These results provided a base on which existing 2D-based mobile contents are easi-ly converted into 3D-based contents, and the function to access 3D contents on a 2D mobile system using existing 3D data.

References

1. http://www.etnews.co.kr
2. http://kr.khronos.ort
3. http://www.jcp.ort/en/jsr/detail?id=184
4. J.K. Cho, Y.H. Park, J.M. Kim, "Design and Implementation of Mobile 3D Engine using JSR-184 on J2ME, KIISE 2005 Fall Conference Proceeding, vol. 32, no. 2, pp. 673--675(2005)
5. H. C. Kim, D. K. Shin, D. I. Shin, "A Design of Context-Awareness Architecture and Game Physics Engine for Mobile 3D Racing Game", KIISE 2009 Korea Computer Conference Proceeding, vol. 35, no. 1, pp. 46--50(2009)