4 S − 9

How to Render 3D Objects in Distributed Virtual Reality Environment
- A Way to Handle Hierarchical Scene Graph Structure in 3D Graphics -

Eric Young-Sang Shim Nobuhito MIYAUCHI Hiroshi KOZUKA
Hisao FUKUOKA Yoshiki SHIMOTSUMA
Information Technology R&D Center, Mitsubishi Electric Corporation

1. Introduction

It is necessary to have a 3D renderer to display the world model in distributed virtual reality environment. SpVisual, a 3D graphics renderer, was developed to render 3D world for a distributed virtual reality environment API called Spline, a Scaleable Platform for Interactive Network Environment, that was developed in Mitsubishi Electric Corp. Spline's target is a session level multimedia communication infrastructure exporting an API on top of which networked virtual environments can be built by an application writer who can just focus on the content of the application without worrying about communication and synchronisation among users. In this research, I will introduce a relationship between spVisual and Spline, and more details of how spVisual handles 3D objects in distributed virtual reality environment such as how spVisual handles multiple VRML files, and how scene graph is differ from the ordinary scene graph. Also, some enhancements how spVisual make its rendering performance faster.

2. Scene Graph

Scene graph is an arranged hierarchical internal structure of 3D graphics nodes, which are more than just a collection of nodes. 3D graphics renderer can render its 3D world model by traversing, modifying, and transforming this scene graph. Since scene graph is very closed related to how to render 3D world model, it is necessary to carefully create a scene graph, and need to traverse it as efficient as possible so that traversing the scene graph does not effect the performance of renderer.

2.1. Multiple VRML File Support

As a 3D graphics renderer, spVisual needed to support multiple VRML files so that user can have as many model as they want to have in a virtual 3D world model without having a huge VRML file, which allows modularity and flexibility of VRML files. 3D renderers which are used in real world these days are supporting only a single VRML file which could lead application developers have lâck of flexibility of their application. To overcome these constrains, spVisual is designed to handle multiple VRML files by using sub scene graph concept.

2.2. Scene Graph

To support multiple VRML file, spVisual constructs sub scene graph, which has hierarchy of each node in a VRML file, within a main scene graph that is handled a little different than what usual OpenGL scene graph. SpVisual constructs a display list for each VRML file, which is described as a sub scene graph in above subsection. Then, spVisual connects this sub scene graph into the scene graph that is the internal tree structure as its scene graph. Using this internal tree structure of Spline, there are some advantages that can be achieved.

There are kinds of advantages as described following. First, since scene graph has the same structure as virtual world model in real time, spVisual can display virtual world without having its own scene graph, and this allows spVisual very reliable performance. Second, there are no overheads, and this will allow spVisual to save time and performance to construct, to maintain, and to remove scene graph. SpVisual loads a VRML file, and convert it to a sub scene graph,

---

How to Render 3D Objects in Distributed Virtual Reality Environment − A way to Handle Hierarchical Scene Graph Structure in 3D Graphics −
Eric Young-Sang Shim, Nobuhito MIYAUCHI, Hiroshi KOZUKA, Hisao FUKUOKA, Yoshiki SHIMOTSUMA
Information Technology R&D Center, Mitsubishi Electric Corp.

called display list. This sub scene graph contains all hierarchy relationships, transform information, material information, and texture information in a VRML file. By connecting this sub scene graph into Spline's internal tree structure, spVisual can have the main scene graph. Since Spline's internal tree structure contains hierarchy relationship and transform information, spVisual can easily retain scene graph. Also, newer version of Spline requires locale concept, and spVisual can be very easily adapted to locale concept by using Spline's own structure. Using this concept of multiple scene graph, I was able to achieve reliable performance, and very easy to understand structure of virtual world model. However, there is a disadvantage of using this concept that is it might take considerable time and effort to port this spVisual to other 3D graphics renderer since the scene graph is very dependent to Spline's internal tree structure.

3. Enhancements

There are number of techniques which can make spVisual fast enough to qualify as a 3D graphics renderer. One good example could be pre clipping. OpenGL does the clipping on every 3D graphics node that was sent to graphics pipeline, however it takes much calculation cost to clip it using OpenGL's method. To avoid this situation, I tried to clip 3D graphics nodes even before I send into pipeline so that I could enhance much calculation cost. Also, tried to use simplified the matrix calculation, cross matrix multiplication, for specific purpose so that I could dramatically reduce the number of calculations for transformation of each node that could be many thousands in some 3D world model cases.

3.1. Pre Clip

As described above, OpenGL does the clipping on every 3D graphics node that was sent to graphics pipeline only when the option is on. However, it takes much calculation cost. To avoid this problem, spVisual was needed to find a way to do pre clipping before it send the 3D graphics node into pipeline. It was accomplished by checking the volume of the node. Simply check if a node is within the rectangular box that contains view frustum of the camera, and if it is inside of the rectangular box, called pre clip box, the node is sent to pipeline to let OpenGL handles the rest of rendering part. In other hands, if it is not inside of the pre clip box, just ignore the graphics node, and keep traversing the scene graph to handle other graphics nodes. The reason that the rectangular box for the pre clip box is used is that it is much efficient than using the pyramid shape view frustum to pre clip. Since this method is called pre clip, it does not need to clip every vertex or every point it is calculating, but using this volume information, just check if the node is within the pre clip box or not. After using this concept, I could accomplish much faster performance, around 20 to 30 percent.

3.2. Calculation

In 3D graphics, a lot of matrix calculations are used, and those calculation costs are very expensive. To reduce the number of calculations, a dedicated matrix multiplication for transform was used in spVisual. To perform the transformation, only 3 values out of 16 need to be multiplied and others are not concerned at all. Using this fact, spVisual uses the specified transform multiplication function so that it could reduce number of calculation dramatically, around 80 percent of calculation cost. This concept enhanced the performance of spVisual dramatically as well as pre clip.

4. Conclusion

Among 3D graphics renderer, it is very core thing to consider if it is reliable, flexible, and high performance renderer especially for virtual reality environment. Using multiple scene graph concepts, as described in Scene Graph section, it was possible to give application developers the reliability, and flexibility. Also, using some of enhancement concepts such as pre clipping and dedicated calculation method, spVisual was fast enough to be a 3D graphics renderer for virtual reality environment, not only reliable and flexibility.

References

[1] Paul M. Summitt, Mary J. Summitt, "Creating Cool 3D Web Worlds with VRML", IDG Books, 1996.

[2] Mason Woo, Jackie Neider, Tom Davis, "OpenGL Programming Guide Second Edition", OpenGL Architecture Review Board, 1997.

[3] Jed Hartman, Josie Wernecke, "The VRML 2.0 Handbook – Building Moving Worlds on the Web", Silicon Graphics, Inc., 1996.

[4] Alan Watt, "3D Computer Graphics – Second Edition", Addison – Wesley Publishing Company, 1995.