The Influence and Optimization Method of Intensive and High Precision Model on Virtual Scene

Wu Xu[1,a], Jingji Yang[1,a], Yali Gao[1,b *]

[1] School of Electronic and Information Engineering, Yunnan Minzu University, Kunming, China

[a]email:xuwu_111@sina.com, [b]email:yali75@126.com

Keywords: X3D, JavaScript, HighPrecision Model, VirtualReality

Abstract: The X3D is a virtual programming language and the target of it is to establish interactive 3D multimedia. With the emergence of X3D, the virtual reality (VR) is entering the lifestyle of the people as well as multimedia and Internet. 3D scene is described by text message in X3D, which can be transmitted on Internet, and then the explorer of X3D on the local machine can generate this 3D scene. Although the VR scene constructed by the X3D can be transmitted rapidly and real-time rendering, a model with high-precision will lead to large text message which has a strong impact on transmission and rendering speed of the virtual scene. In this paper, a real-time loading mode to a model with high-precision is presented. Feasibility of the VR scene with intensive high-precision models has been improved through optimizing the scene with high-precision model.

Introduction

Internet technology linking people around the world together has changed people's lives obviously. People are undergoing rapid changes to their lives such as working, studying, entertainment, communication. People are enjoy in various conveniences like network communication, shopping remote education and so on[1]. However, these are the 2D technology of the Internet. With the rapid development of the computer hardware, sensing equipment such as tridimensional helmet, data glove and touchscreen are gradually improved[2]. X3D technology which can simulate the seeing hearing touching is received high attentions. X3D supports not only 3D representation of data and motor process but also real-time sound effect. Users are brought into a virtual world with high quality acoustic and visual effects in which they can interact with the virtual object[3]. This advanced digital human-machine interface technology is widely used in national defense, commercial manufacture, medical treatment, etc.

As a descriptive VR modeling language based on WWW, X3D can generate 3D entitative object[4]. The X3D can improve performance of the interactivity in WWW with which users are given a 3D virtual space and their actions are taken as main part of the browsing. In order to enhance the immersed sense in the virtual scene generate by X3D, files (*.X3d) should be processed to improve performance at run time. From X3D, major factors as follows: the first is the time spending of downloading scene files due to network bandwidth, and the second is scene browsing is a time consuming work even on a high performance computer[5].

Modeling Optimization

Models in X3D have properties of physical dimension and appearance and they are defined by the node Shape, which decide the models in the virtual space[6]. Size values and appearance are controlled by the given node and selected value of the node domain. Appearance of the model is described by the Appearance node and Material node. Four original geometric models provided by X3D are Sphere, Box, Cylinder and Cone, based on which any complicated object can be generated by these four geometric models[7]. Geometric nodes, however, cannot construct special models, and the X3D-EDIT is not visual software for editing which means the modeling process is very complicated. Other modeling software such as 3DS MAX are needed to construct the model and output files in VRML and then convert to X3D format through X3D-EDIT. As shown in Figure.1.

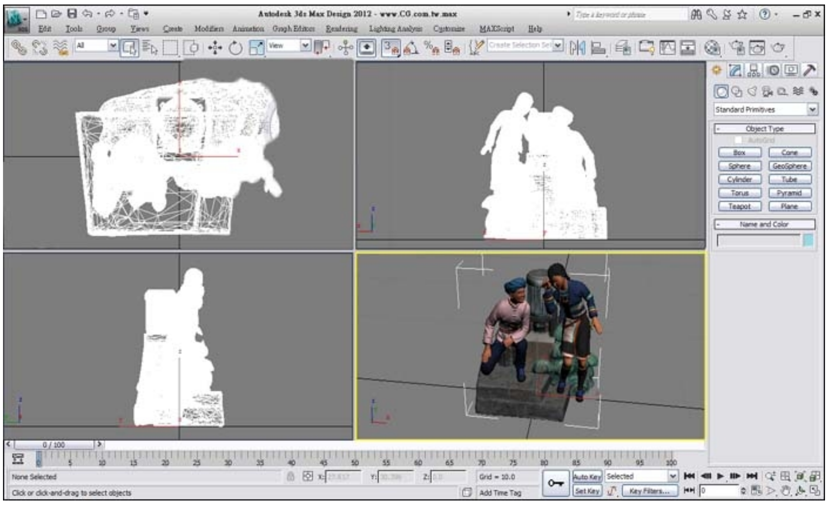


Fig 1. Model building

In order to increase the files transmission speed and provide users convenience interacting with virtual environment, orientation of the model should be modified and the visual effect should also be optimized to reduce the file size. The implementation methods are usually file compressing and optimizing.

File Compression

Editing platform of VRML is VRMLPad which can compress files. The compressed files size is 1/3 of what they were, but the visual effect remains un-changed. Compressing method: click “File“ of the VRMLPad menu and then “Save As”. Thus the files can be compressed after click the “Save compressed”.

Optimization of the Source Code

VRMLPad editor can not only to compress files, but also can direct optimizing source code .Click the “Unused Identifiers” form “tools” on the menu, and then click the “All unused identifiers” to delete the prompts never have been used. The command framework also provides basic command management functions, like ”Unused PROTO declarations”, “Unused node names”, ”Unused field declarations”, “Default field values”, “Redundant fields”, and “Duplication routes”[8].

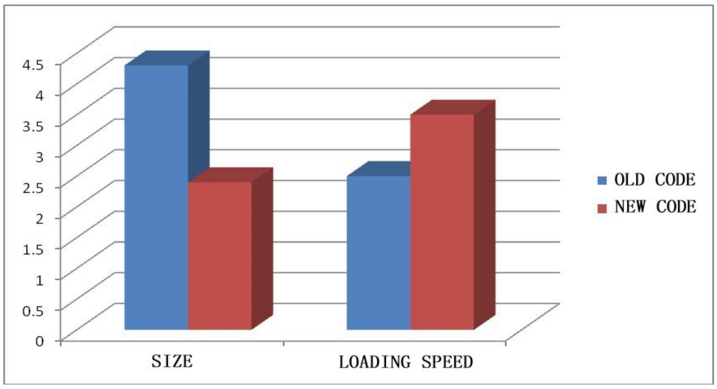
VRMLPad can also optimize the source codes using plugin tools. Download ”Comments.rar” and extract to …\VRMLPad\AddIns\, and the command plugin menu can be found at tools menu after starting VRMLPad. There are six commands in ”Command menu”. “code format” displays the structural relationships between the codes in best mode. “delete comment” can delete all comments. “delete default region” can delete default defined values contents of the nodes. “optimize codes” can optimize blank areas, breakpoints and so on automatically. “optimize codes & preserve comments” can preserve the comments when optimizing operation. ”reduce codes indentation” can delete the codes space of each row. Files size can be compressed 10% after optimizing.

Graphical Optimization

When creating scenes, as vein files, graphics files like JPEG, GIF and PNG can be stick-on the model as vein files. For JPEG, as a lossy compression, it can reach compressing ratio at 100:1 without distortion. Thus for many scenes, it can satisfy the browsing needs and this format is often adopted.

Through optimizing mentioned above, the model files can be compressed furthest. Then the VRML files are converted into X3D files using X3D-EDIT. Table 1 shows the comparison of the source codes before and after the optimizing.

TABLE I. A COMPARISON OF RESULTS BEFORE AND AFTER OPTIMIZATION



Programming Optimization

USE the node. Once a node is named, it can be called in same scene using “USE” before the node name. the node can also be regarded as a range value when there needs a whole node description. USE and cite the primary node with random times in the same file. Thus, all the instances are changed together while changing the primary node. During the modeling process, one model may be reused many times in the scene. A model node can be used repeatedly when it was defined one time in the file, which avoids using repetitive codes. The key codes are as follows:

```
<Shape DEF=”Leg”>
    <Appearance>
        <Material diffuseColor = “1 1 1”/>
    </ Appearance>
    <Cylinder height = “1” radius= “0.1”/>
</Shape>
<Transform translation=”0.5 0 -0.5” USE = “Leg”/>
```

LOD node. In order to reduce the burden of X3D browser, X3D provides a LOD node. LOD (Level Of Derail) permits browser switching automatically between different levels of the object, and describes the level relations of the same model. Since details of the model in the distance are far less than that of the near model, the virtual space can appear in different details with different observation distance through range settings. Thus, the scene rendering can be speed up, and the realistic X3D world are also increased.

With the two methods mentioned above, memory requirements of the virtual scene are decreased largely through optimizing the model, and figure.2 shows the fundamental principle.

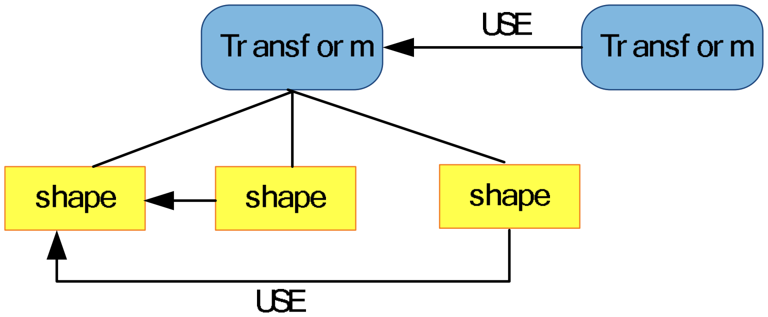


Fig 2. Code optimization

Scene Loading

Codes of the model are approximate to optima through the two optimizing method mentioned above. At the present time, all the models are loaded in the virtual scene once only. The loading time and fluency will be seriously affected using this one time loading method if there are intensive and high precision models, thus a special processing method will be needed to these models.

Models are described using text mode by X3D, of which file loading velocity of one high-precision model is very fast. Inline node is adopted to control the loading time and fluency of the high-precision model in the scene. Table 2 shows the comparison of the two controlling methods.

TABLE II. CONTRAST TO THE CONTROL MODE

Control mode	Load time	Fluency of Scenes	Network transfer time	Memory footprint
One-time loading	long	slow	long	large
Loading by Inline node	short	fast	short	small

One-time loading. It is a models loading method in the scene with-out any controller, which means models in the scene are mapped to memory only once and rendered by the plugin of the browser. This method, however, needs not only a long time to load the text massage but also slow down the rendering of the whole scene. There will be a lot of memory needed, which leads virtual scene loading velocity very slowly even. memory overflow, if there are intensive and high precision models.

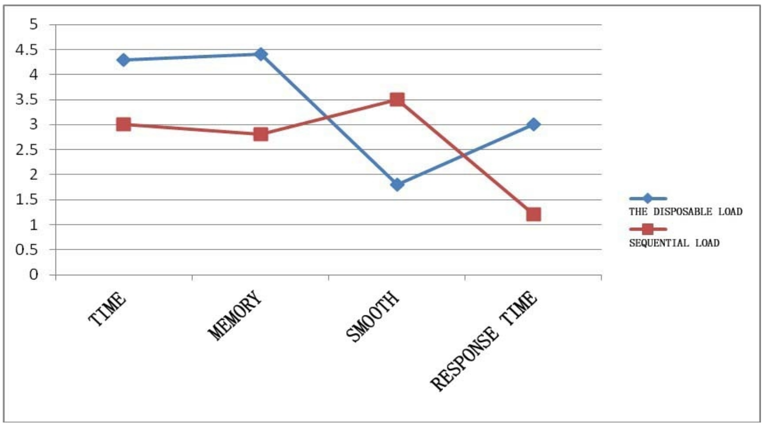
Loading by Inline node. Loading time is controlled by the Inline node, and it uses a texturing integrates all the high precision models instead of the model itself. After observed carefully by users the models can be loaded by the Inline node, and the key codes are as follows:

```
<Switch DEF="cdns2" whichChoice="-1">
    <Transform translation ="0 0 0">
    <ProximitySensor  DEF ="jjcgq37"   size ="100000 100000 100000"   center ="0 100 0"/>
    <Transform DEF ="cdzt37"   translation ="0 0 0">
    <TouchSensor DEF="touchns2"/>
    <Transform translation =".305 0.12 -.57">
```

```
        <Shape> <Appearance>
                    <ImageTexture url ="maps/cdns2.jpg"/>
                </Appearance>
            <Box size =".03 .0240 0"/>
        </Shape> </Transform> </Transform> </Transform></Switch>
 <ROUTE fromField="position_changed"  fromNode="jjcgq37"  toField="translation"
toNode="cdzt37"/>
 <ROUTE fromField="orientation_changed"   fromNode="jjcgq37"   toField="rotation"
  toNode="cdzt37"/>
   <Transform DEF="ns2"    translation='8144 138.1 -17270'    rotation='0 1 0 -2.074'>
      <Inline load="false" DEF="loadns2"       url="ns/ns2/ns2.x3d" >
         <LoadSensor DEF="isloadns2"/>
      </Inline> </Transform>
   <Script DEF="Mover">
     <field accessType="initializeOnly" name="ns2"      type="SFNode" >
        <Transform USE="ns2"/></field>
     <field name="cdns02" type="SFTime"     accessType="inputOnly" />
     <field accessType="initializeOnly"      name="loadns2" type="SFNode" >
        <Transform USE="loadns2"/></field>
   <![CDATA[ecmascript:
        function cdns02(value)
         { loadns2.load=true;  ......  }
   </Script>
<ROUTE fromNode="touchns2"    fromField="touchTime" toNode="Mover"     toField="cdns02" />
```

As the codes mentioned above, Load property of ns2 is false which means the model does not be loaded. Load property of ns2 is true while the menu but-ton of the ns2 is clicked, and the model is loaded in the scene immediately for users browsing. Table 3 shows the comparison of the performance parameters.

TABLE III. PERFORMANCE COMPARISONS



Summary

Other optimizing methods such as floating-point integer processing, compressed veins, can be used to establish virtual scene using X3D. Meanwhile, during the creating process, balance of the browsing velocity and visual effect of the virtual scene must be noted. Perfect virtual simulation effect and browsing fluency are both important for virtual scene, which must be taken into account together. Thus, the virtual simulation of X3D can reach the best effect and availability.

Acknowledgment

This work was financially supported by the Scientific Research Foundation of Yunnan Provincial Education Department (2013Y020) and the Ethnic Minorities' Traditional Culture Rescue and Protection Foundation 2013.

References

[1] Xu Wu, Shao Jian Long. Application of VRML in the virtual city scene,C, Modern Information Technology Theory and Application (Book), China University of Technology Press,2002,pp. 504–506. (In Chinese)

[2] J.J. LIN, W. WEI, and Y.B. Yang, "Content-based design and implementation of 3D model database and retrieval system, " Computer Sci-ence, vol. 35, no. 10, pp. 238-242, October 2008.

[3] Xiong jianxin, Zhilin, "modern interior design", Wuhan university of science and technology press,2005, pp.78-82

[4] Xu Wu.The application of VRML in motion of people and car, J.Journal of Higher Correspondence Education (Natual Sciences),2010, 23(1),pp. 74–76. (In Chinese)

[5] C.T. Khuan, A. Abdul-Rahman, and S. Zlatano-va, "New 3D data type and topological opera-tions for Geo-DBMS, " Proc. Conf. Urban and Regional Data Management (UDMS Annual 2007), pp. 211-222, 2008

[6] Ye Yan Qing, Shao Jian Long, Nian Xiao. Interactive Virtual Reality Modeling based on VRML, J. System Simulation, 2006,18 (10),pp. 2827–2831. (In Chinese)

[7] Ruo guoshi, " the research of display design about virtual bronze culture museum", art school of Huangshi science and technology insti-tute,2007,pp.45-50

[8] Shao Jian Long, Ye Yan Qing, Xu Wu. The implementation of Java programming in VRML virtual scene, J. Computer and Applied Chemis-try, 2004, 21 (4),pp. 641–644. (In Chinese)