On Enhancing the Image Quality of Dynamical X3D Contents in the Interne

Jong-Sung Ha[a] and Kwan-Hee Yoo[b]

[a] *Dept. of Game and Contents, Woosuk University,*
490,Hujong-ri, Samrae-up, Wanju-kun, Chonbuk, 565-701, Korea
Tel: +82-63-290-1455, Fax: +82- 63-290-1453, E-mail:jsha@woosuk.ac.kr

[b] *Dept. of Computer Education, Chungbuk National University*
San 12, Gaesin-dong, Heungduk-gu, cheongju-si, Chungbuk, 361-763, Korea
Tel: +82-43-261-2788, Fax: +82-43-273-4031, E-mail:khyoo@chungbuk.ac.kr

Abstract: *This paper presents the practice of multitexturing in the Internet for enhancing the image quality of 3D contents, where the attributes of objects such as textures are dynamically changed. We explain the empirical results of realizing the X3D nodes related with multitexturing in the recent X3D viewers, and discuss the directions for upgrading X3D viewers that satisfy the user requirements and the advanced graphics accelerators.*
Keywords: *X3D; Multitexturing; 3D contents;*

1. Introduction

X3D (eXtensible 3D) [1] is the next generation of VRML (Virtual Reality Modeling Language) that is a standard description language for 3D contents in the Internet. This paper is concerned with enhancing the image quality of 3D contents in the Internet with the multitexturing technique [2] that has been newly added to X3D. Multitexturing is able to apply two or more textures to the same primitive usually by combining the texture through the fixed or programmable hardware pipelines supporting OpenGL [3] and DirectX [4] in a single pass.

In order to speed up the rendering time and enhance the image quality, the real-time applications such as 3D games use the multitexturing technique of combining an image map and a light map that is previously computed by many shading techniques such as the Phong shading [5] or the global shading of radiosity [6] and ray tracing [7]. With the same method, it became possible to enhance the rendering speed and image quality of the dynamical X3D contents that can be widely used in the Internet.

In this paper, we present the practice of applying the multitexturing-related nodes of X3D to the existing VRML site of E-IAS system [8] for enhancing the image quality. Since the E-IAS system provides the function for navigating a virtual room while changing the textures of wallpapers and furniture, it is very important to provide more realistic images with high-level rendering. For the static objects, we generate the burned textures that are the combined images with light maps. For the dynamics objects, we generate light maps independently to the image, and they are combined with the images in the rendering process, that is just multitextured.

2. Practice in Multitexturing for Dynamical X3D Contents

In usual, 3D models are designed with a special tool such as 3D Max and they are transformed into X3D files by the plug-in softwares called X3D exporters. The X3D files are rendered in web browsers by the plug-in softwares called viewers. If there exists a complete X3D viewer that supports all nodes related with multitexturing, it is meaningless to discuss about the practice of multitexturing in X3D. Unfortunately, however, the existing viewers in current have limitations and they are not compatible to each other. Table 1 compares the recent X3D viewers. We realize the multitexturing in Cortona [9] that was adopted by E-IAS system and Flux [10] and BlaxxunContact [11] that are relatively excellent in image quality and rendering speed.

There are three nodes related with multitexturing in X3D: MultiTexture-Node, MultiTransform-Node, and MultiTextureCoordinate-Node. The following steps for realizing the X3D multitexturing assume that it is needed for editing the X3D files of an incomplete X3D exporter that cannot generate the three nodes exactly, since there are neither complete X3D exporters nor viewers, and furthermore they are incompatible to each other in current.

Step 1: In 3D Max, burned textures for static objects are generated after setting up appropriate lights, the original textures are replaced with the burned textures, and then the objects are exported into X3D files.

Step 2: In 3D Max, light maps and texture coordinates for dynamic objects are generated after setting up appropriate lights, the original textures are replaced with the light maps, and then the objects are exported into X3D files. Figure 1 illustrates the light maps of a floor and a closet in a designed room model.

Step 3: By combining Texture-Node, Transform-Node, and TextureCoordinate-Node in the two files generated in Step 1 and 2 respectively, the multitexturing nodes of MultiTexture-Node, MultiTransform-Node, and MultiTextureCoordinate-Node are implemented in a final file. We described the main codes for multitexturing in each X3D viewer in Table 2.

Table 1. Comparison of X3D viewers

	image quality	speed	updating node fields		navigation GUI	light map type	license
			JAVA EAI	ActiveX Control Interface			
Flux	very high	very fast	no	unkown	no	color	commercial
BlaxxunContact	high	fast	yes	yes	no	color	commercial
Cortona	low	very slow	yes	yes	yes	gray scale	commercial
BSContact	high	fast	yes	yes			commercial
FreeWRL	low		yes				free
Xj3D			yes				free
Octaga							commercial
OpenWorlds							commercial

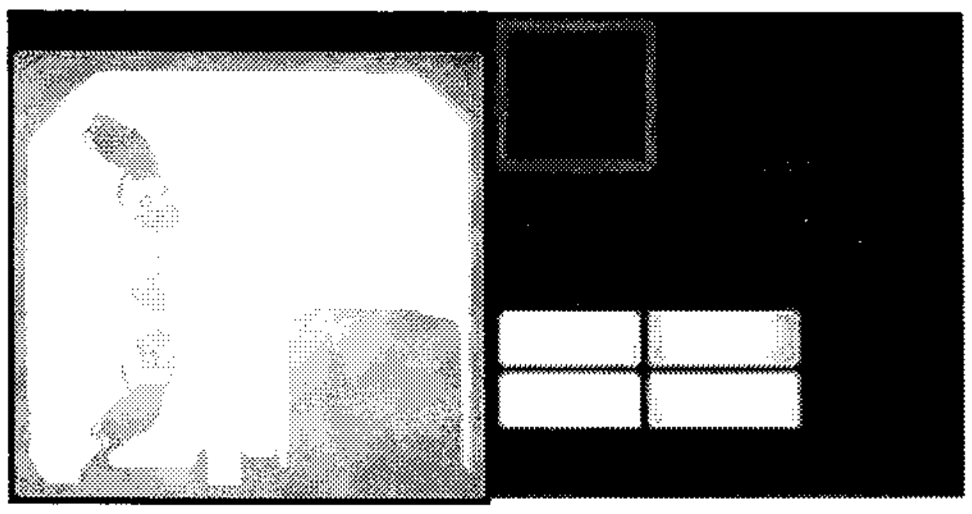

Figure 1. Light maps of a floor and a closet

Table 2. Multitexturing description in Cortona/BlaxxunContact/Flux

Cortona	BlaxxunContact	Flux

appearance **AdvancedAppearance** {
material Material {...
}
textures [
DEF Tnode ImageTexture {
url "texture.jpg"
}
ImageTexture {
url "lightmap.png"
}
]
mappingTypes
["SIMPLE","SIMPLE"]
backgroundFactor["ZERO"
"BACK_COLOR"]
foregroundFactor ["ONE"
"FORE_COLOR"]
textureTransforms [
TextureTransform{
}
TextureTransform{
}
]
}
	appearance Appearance {
	material DEF X456 Material {
	...
	}
	texture MultiTexture {
	mode ["REPLACE"
	"MODULATE"]
	texture [
	DEF Tnode ImageTexture
	{
	url "texture.jpg"
	}
	ImageTexture {
	url "lightmap.jpg"
	}
	]
	textureTransform [
	...
	]
	}
	}
		appearance Appearance {
		material DEF X456 Material {
		...
		}
		texture MultiTexture {
		mode ["REPLACE"
		"ADDSMOOTH"]
		texture [
		DEF Tnode ImageTexture {
		url "texture.jpg"
		}
		ImageTexture {
		url "lightmap.jpg"
		}
		]
		}
		textureTransform
		MultiTextureTransform {
		textureTransform [
		...
		]
		}
		}

3. Results and Concluding Remarks

This paper presented how to realize the multitexturing feature in the recent X3D viewers and exporters that have many limitations yet. Figure 2 illustrates the rendering images before and after applying the multitexturing techniques to the 3D contents of a virtual room. Even though the multitexturing could not be realized delicately due to the limitations of the existing X3D viewers, the image quality could be enhanced considerably without any sacrifice of rendering speed. It is also expected that the image quality can be enhanced in certain objects by using the bump maps [12] that are used for describing smoother or uneven surfaces with fewer data.

(a) no multitexturing in Cortona (b) multitexturing in Cortona

(c) multitexturing in BlaxxunContact (d) multitexturing in Flux

Figure 2. Rendering results in each X3D viewer

The X3D specifications [13] include many nodes for utilizing the graphic accelerators and has the flexibility to extend the nodes for adding new features. Especially, the image can be leaped to higher quality by realizing the Shader-Node that is defined for programmable pipelines, in which dedicated shading methods and advanced mapping methods such as displacement mapping [14] can be loaded. Hence, it is inevitable that X3D viewers will be evolved as the graphic accelerators do as well as they will be upgraded for supporting more nodes such the MultiTextureCoordinate-Node that is necessary for more accurate combining of an image and a light map.

References:

[1] http://www.web3d.org/x3d

[2]Watt, A. and Policarpo, F. (2001). *3D Games Real-time Rendering and Software Technology*, Addison-Wesley, New York, pp.315, 346-348.

[3] http://www.opengl.org

[4] http://www.microsoft.com/directx

[5] Hall, R.A. (1986). "Characterization of Illumination Models and Shading Techniques," *The Visual Computer*, Vol.2, pp.268-277.

[6] Willmott, A.J. and P. S. Heckbert, P.S. (1997). "An Empirical Comparison of Radiosity Algorithms," Technical Report, CMU-CS-97-115.

[7] Whitted, T. (1980). "An Improved Illumination Model for Shaded Display," *Comm. ACM*, Vol.23, No.6, pp.343-349.
[8] http://www.knowledgetech.co.kr
[9] http://www.parallelgraphics.com
[10] http://www.mediamachines.com
[11] http://developer.blaxxun.com
[12] Blinn, J.F. (1978). "Simulation of Wrinkled Surfaces," *Computer Graphics (SIGGRAPH '78 Proceedings)*, Vol.12, pp.286-292.
[13] http://www.web3d.org
[14] Cook, R.L. (1984). "Shade Trees," *Computer Graphics (SIGGRAPH '84 Proceedings)*, Vol.13, No.3, pp.223-231.