

동적 X3D 콘텐츠의 영상 품질 향상을 위한 다중텍스처링 기법의 적용

Applying Multitexturing Techniques to Increasing the Image Quality of Dynamical X3D Contents

하종성*, 유관희**

우석대학교 게임콘텐츠학과*, 충북대학교 컴퓨터교육과**

Jong-Sung Ha(jsha@woosuk.ac.kr)*, Kwan-Hee Yoo(khyoo@chungbuk.ac.kr)**

요약

본 논문은 텍스처와 같은 물체의 속성이 동적으로 변하는 인터넷 3D 콘텐츠에서 영상 품질을 향상시키기 위한 다중텍스처링 기법에 관한 것이다. 다중텍스처링과 관련된 X3D 노드들을 최근의 X3D 뷰어를 중심으로 실험하여 이들의 활용 방법과 제약점을 알아보고 이를 바탕으로 향후 사용자 요구 및 그래픽가속기의 발전에 따른 차세대 X3D 뷰어의 개발 방향에 대하여 논의한다.

■ 중심어 : | 3D 콘텐츠 | 가상현실 | X3D | 다중텍스처링 |

Abstract

This paper is concerned with the techniques of multitexturing for increasing the image quality of 3D contents in the Internet, where the attributes of objects such as textures are dynamically changed. First we explain the empirical results of implementing the X3D nodes related with the multitexturing in the recent X3D viewers. Next we discuss the directions for developing the next-generation of X3D viewers that satisfy the user requirements and the advanced graphics accelerators.

■ keyword : | 3D Contents | Virtual Reality | X3D | Multitexturing |

I. 개요

X3D(eXtensible 3D)[1]는 웹브라우저에서 뷰어(viewer)를 사용해 3D 콘텐츠를 볼 수 있도록 표현하는 표준언어로서 기존의 인터넷 가상현실 표현 언어인 VRML(Virtual Reality Modeling Language)의 여러 문제점을 보완하고 장점을 발전시킨 것이다.

1994년 제1회 WWW 컨퍼런스에서 처음 언급된 VRML 기술은 3D 장면을 그래프 노드(node) 구조로

표현하는 실리콘그래픽스사의 오픈인벤터(Open Inventor) 파일 포맷에 기초하여 제정되었다. 이어 1996년에는 실리콘그래픽스사의 무빙월드(Moving World)에 기초하여 VRML2.0 규약이 탄생하였다. 2.0은 인터넷에서 단순히 3D 콘텐츠를 보는 것뿐만 아니라 사용자와의 상호 작용 기능들과 키프레임 애니메이션을 가능하게 한 것이다. 1997년의 VRML97 규약의 최종판은 기술적으로 명문화하고 일부 기능을 수정하여 ISO에서 승인된 것이다. VRML을 보완하고 발전시켜 2004년 국제표준으로

* 본 연구는 (주)케이테크의 산학협동 및 자문활동 지원으로 수행되었습니다.

접수번호 : #060313-003

접수일자 : 2006년 03월 13일

심사완료일 : 2006년 03월 22일

교신저자 : 하종성, e-mail : jsha@woosuk.ac.kr

로 승인된 X3D는 새로운 기능이 추가될 수 있는 확장성과 유연성을 가지고 있으며 기존의 노드 외에 확장된 다양한 노드들이 추가되어 있다. X3D는 애니메이션 및 CAD 등 기존에 독립형(stand-alone) 컴퓨터상에서 가능하던 3D 그래픽스 응용들을 인터넷 브라우저를 통하여 볼 수 있는 구현을 가능하게 해준다.

본 고에서는 기존에 구축된 VRML 콘텐츠의 영상 품질(이하 화질)을 향상시키기 위하여 컴퓨터그래픽스 실시간 렌더링(rendering)에서 속도 및 화질 향상을 위하여 많이 사용되고 있는 다중텍스처링(multitexturing) 기법[2]을 X3D의 다중텍스처링 관련 노드들과 최근의 X3D 뷰어들을 이용하여 실제 실험해본 사례를 설명한다. 이를 통하여 웹브라우저의 플러그인(plugin)으로 사용될 수 있는 최근의 X3D 뷰어들의 활용방법과 그 제한점 그리고 향후의 뷰어 개발 방향 등에 대하여 논의한다.

II. 실시간 화질 향상을 위한 다중텍스처링

1. 텍스처링과 셰이딩

컴퓨터그래픽스에서 주어진 3D 모델 자료를 렌더링하여 얻어지는 결과의 영상 품질에 가장 큰 영향을 주는 과정은 물체의 표면에 무늬를 입히는 텍스처링 기법과 조명효과를 주기 위하여 물체 표면의 광도를 계산하는 셰이딩(shading) 기법이다. 각 화소마다 계산되어야 하는 이들 연산은 시간이 많이 걸리므로 3D 그래픽 라이브러리의 실제적인 표준인 OpenGL[3]과 DirectX[4]의 사양을 지원하도록 그래픽스카드의 렌더링 파이프라인(pipe-line)으로 구현되기도 한다.

텍스처링은 보통 2D 영상을 3D 모델의 표면에 매핑하는 과정인데 모델의 표면의 복잡할수록 그 매핑방법도 어려워진다. 현실적이고 자연스러운 매핑을 위하여 여러 매핑연산들이 연구되어 왔다[5]. 조명에 의한 물체 표면의 광도를 계산하는 방법인 셰이딩은 각 물체를 독립적으로 처리하는지 또는 통합적으로 처리하는지에 따라 국부적 셰이딩(local shading)과 전역적 셰이딩(global shading)으로 나눌 수 있다. 계산시간이 비교

적 빠른 국부적 셰이딩은 실시간렌더링에서 사용될 수 있는 방법으로 Goraud 셰이딩 및 Phong 셰이딩[6]이 있다. 보통 3D 그래픽가속기는 Goraud 셰이딩을 지원하므로 X3D 뷰어도 이 방법을 사용하고 있다. 그러나 화질과 시간은 서로 교환관계가 있으므로 radiosity[7]나 ray tracing[8]과 같이 한 화면을 계산하는데 많은 시간을 소요하는 전역적 셰이딩은 영화와 같은 오프라인의 고품질 생성에 사용되며 X3D 뷰어와 같은 실시간 렌더링에서 적용할 수는 없다.

2. 동적 X3D 콘텐츠의 다중텍스처링

텍스처링과 셰이딩은 각각 그래픽가속기에서 독립적으로 처리되지만 미리 전역적셰이딩 결과의 광도를 적용한 텍스처를 만들 수 있으므로 서로 관련이 있는 기능이다. 즉 전역적 셰이딩을 미리 계산하여 텍스처링으로 대신할 수 있는 것이다. 따라서 렌더링 속도가 매우 빨라야 하는 게임 제작과 같은 분야에서는 국부적 셰이딩조차도 미리 계산하는 방법을 주로 사용하여 실제감을 얻으면서도 시간을 단축하는 효과를 낸다. 움직이지 않는 정적인 물체의 경우 기존 텍스처 위에 조명의 위치에 따라 미리 계산된 광도를 반영한 것을 burned texture 또는 baked texture라 부르기도 한다.

움직이는 동적인 물체의 경우 위와 같은 방법을 적용할 수는 없으므로 동영상의 프레임이 바뀔 때마다 셰이딩 처리를 계산해주어야 한다. 만약 손전등과 같이 조명이 움직이는 경우라면 이 조명의 계산된 광도만을 따로 저장하여 매 프레임마다 다른 텍스처와 결합시키는 다중텍스처링 기법을 사용한다.

이 다중텍스처링은 인터넷상에서 물체의 표면 텍스처를 동적으로 변경해야 하는 경우가 많은 동적인 X3D 콘텐츠에서 화질을 향상시키기 위하여 적용될 수 있으며 인터넷상에서 신제품 소개와 같은 분야에 널리 활용될 수 있다. 즉 텍스처를 동적으로 바꾸어야 하는 경우 계산된 광도만 독립적으로 저장한 이 라이트맵(lightmap)을 동적으로 바뀌는 텍스처와 결합만 함으로써 실시간에 고품질을 얻을 수 있는 것이다.

III. 동적 X3D 콘텐츠의 화질 향상 실험



그림 1. E-IAS 시스템

본 고에서 다중텍스처링 기법을 적용하여 동적 X3D 콘텐츠의 화질을 향상시키기 위한 연구의 대상은 E-IAS 시스템[9]이다. 이 시스템은 [그림 1]과 같이 다량의 벽지, 바닥재 텍스처들을 자신의 기호에 맞게 감성 기반으로 실시간에 검색하고 선택하여 가상공간에서 입체보고 항해(navigation)하는 기능을 제공하고 있다. 따라서 제품에 대한 충분한 검증에 위하여 보다 사실적인 영상을 제공하는 것이 매우 중요하다.

기존의 VRML97을 그대로 사용하여 이 콘텐츠의 화질을 향상시키기 위해서는 대표적인 3D 모델링 툴의 일종인 3D MAX 등에서 모든 텍스처의 조명을 계산하여 각 물체에 대한 burned texture를 만들면 되나 텍스처가 동적으로 변하는 물체의 경우 대량의 텍스처 때문에 현실적으로 불가능하다. 따라서 정적인 물체는 burned texture를 사용하고 동적인 물체는 II.2절의 다중텍스처링 기법을 적용한 X3D 콘텐츠로 재구성하기 위하여 텍스처 자료의 양과 관계없이 각 물체에 대하여 하나의 라이트맵만 미리 만들어 독립적으로 저장하고 동적으로 변하는 텍스처와 이 라이트맵을 결합하는 방법을 사용하고자 한다.

1. X3D 뷰어의 선택을 위한 비교 분석

3D 모델은 3D MAX 등과 같은 모델링 툴에서 디자인되지만 게임 등 각 특정 응용에서는 각자 사용하는 고유의 파일 형식을 가지게 되므로 보통 그 파일의 형

식에 맞게 3D 모델 자료를 변환하여 저장하는 플러그인도 그 응용별로 개발하게 된다. 그러한 플러그인을 exproter라 하며 X3D 콘텐츠의 파일 형식을 얻어내는 X3D exporter도 필요하다. 만약 사용 가능한 완전한 X3D exporter와 다중텍스처에 관련된 노드들을 완전히 구현한 X3D 뷰어가 있다면 더 이상의 논의는 필요 없다. 그러나 최근까지 구현되어 있는 X3D 뷰어들은 각 뷰어마다 기능의 제한과 차이뿐만 아니라 다중텍스처링 표현 방식에서도 서로 호환성이 없다. 따라서 새로운 뷰어의 구현을 연구하거나 제한된 뷰어에서 콘텐츠의 화질을 향상시키는 연구가 필요하며 본 고는 후자와 같이 최근 사용 가능한 뷰어들에서 동적 콘텐츠의 화질 향상 방법을 더 논의하고 향후 뷰어의 개발 방향에 대해서도 언급하고자 한다.

지금까지 알려진 X3D 뷰어는 [표 1]과 같이 정리할 수 있다. 이 표에서는 뷰어의 선택에 고려될 사항 중에 가장 중요한 것만을 비교한 것이다. 본 고에서는 기존의 E-IAS 시스템에서 사용하고 있는 Cortona[10]와 화질과 속도가 상대적으로 우수한 Flux[11]와 BlaxxunContact[12]에서 다중텍스처링의 구현 방법을 보인다.

뷰어의 화질과 속도 외에 동적 콘텐츠의 제어를 위해서는 노드 필드의 변경 방법 즉 텍스처 데이터베이스 검색 모듈과의 연동 방법과 항해 GUI(Graphical User Interface)의 제공여부도 중요하다.

표 1. X3D 뷰어의 비교

특징 종류	특징		노드 필드 변경 방법		항해 GUI	라이트 맵 형식	서 작 권
	화질	속도	JAVA EAI	ActiveX Control Interface			
Flux	가장 높음	가장 빠름	없음	확인 필요	없음	color	상용
Blaxxun Contact	높음	빠름	지원	지원	없음	color	상용
Cortona	낮음	매우 느림	지원	지원	제공	gray scale	상용
BSContact	높음	빠름	지원	지원			상용
FreeWRL	낮음		지원				무료
Xj3D			지원				무료
Octaga							상용
OpenWorlds							상용

노드 필드를 변경하는 VRML EAI(External Authoring Interface)를 자바 애플릿 안에 DB모듈과 통합하고 웹 페이지 자바스크립트(webpage JavaScript)로 그 애플릿 함수를 호출하는 방법은 기존의 E-IAS 시스템에서 택했던 방법으로 기존의 시스템 변경 없이 그대로 사용할 수 있는 장점이 있으나 화질과 속도가 상대적으로 가장 우수한 Flux에서 이 방법을 사용하기 위해서는 역시 Flux를 구동하는 VRML EAI가 개발되어야 한다. 뷰어 ActiveXControl 인터페이스를 웹페이지 자바스크립트로 호출하는 방법은 E-IAS 시스템의 기존의 DB 모듈을 변경하여야 하는 번거로움이 있으나 논리 모듈과 사용자인터페이스 모듈의 독립성을 유지할 수 있는 장점이 있다. 또한 Flux SDK에서도 ActiveXControl 인터페이스는 제공되고 있다.

Cortona와 달리 속도가 빠른 Flux와 Blaxxun은 항해를 위한 GUI를 제공하지 않는 단점이 있으나 GUI는 각 응용에 맞게 비교적 간단히 구성이 가능하므로 융통성을 제공하는 장점이 될 수도 있다. GUI를 구성하는 방법은 첫째 웹페이지에서 뷰어 밑에 버튼 및 슬라이드 바(slide bar) 등으로 인터페이스를 만들고 ActiveXControl 인터페이스를 직접 호출하여 제어하거나, 둘째 뷰어의 ActiveXControl을 상속 받아 인터페이스를 추가한 새로운 ActiveXControl을 작성하는 방법이 있다.

2. 라이트맵 생성과 다중텍스처 노드 구성 과정

다음은 3D MAX 모델링 툴과 Cortona, BlaxxunContact, Flux 뷰어들로 간단하게 실험한 과정을 설명한다. 다중텍스처링에 관련된 X3D의 다중텍스처 노드, 다중변환 노드 및 다중텍스처좌표 노드를 하나라도 export할 수 없는 X3D exporter를 사용할 때 X3D 파일의 편집 작업이 필요한 경우를 가정한다.

- 단계 1 : [그림 2]와 같이 텍스처가 입혀지고 조명이 설치된 원래의 3D MAX 파일에서 속성이 정적인 물체는 burned texture를 만들어 본래의 텍스처를 대신한 후 X3D 파일로 export한다. 이 때 상세한 조명보다는 그림자 등을 위주로 명암을 강조한 조명이 실시간이라는 제약으로 오는 뷰어의 화질 표현

한계 안에서는 더 효과적일 것으로 보인다.

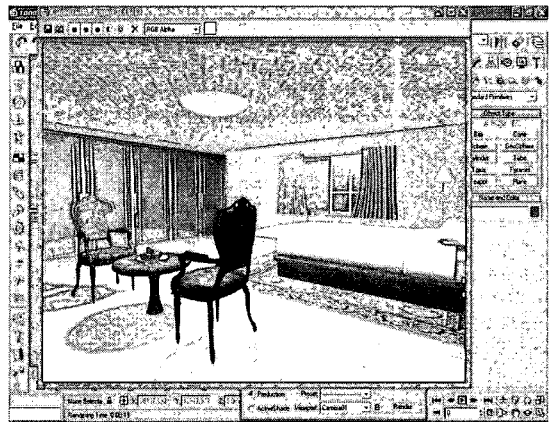


그림 2. 3D. MAX에서 디자인된 실내 내부

- 단계 2 : 속성이 동적인 모든 물체별로 라이트맵과 그 텍스처좌표값을 생성하고 각 물체마다 본래의 텍스처를 없애고 생성된 라이트맵을 생성된 텍스처좌표값을 이용하여 입힌 후 파일로 export한다. 이것은 3D MAX에서 결합된 텍스처와 라이트맵을 다중 텍스처 노드로 export할 수 없는 경우 라이트맵에 대한 텍스처좌표값을 언고자 함이다. [그림 3]은 바닥 옷장의 라이트맵 예이다.

표 2. Cortona/BlaxxunContact/Flux에서 다중텍스처링 표현

	다중텍스처 노드 및 다중변환 노드의 구조
Cortona	<pre> appearance AdvancedAppearance { material Material { ... } textures [DEF Tnode ImageTexture { url "texture.jpg" } ImageTexture { url "light.png" }] mappingTypes ["SIMPLE", "SIMPLE"] backgroundFactor ["ZERO" "BACK_COLOR"] foregroundFactor ["ONE" "FORE_COLOR"] textureTransforms [TextureTransform() TextureTransform()] } </pre>

BlaxxunContact	<pre> appearance Appearance { material DEF X456 Material { ... } textureMultiTexture { mode ["REPLACE" "MODULATE"] texture [DEF Tnode ImageTexture { url "texture.jpg" } ImageTexture { url "light.jpg" }] textureTransform [...] } } </pre>
Flux	<pre> appearance Appearance { material DEF X456 Material { ... } textureMultiTexture { mode ["REPLACE" "ADDSMOOTH"] texture [DEF Tnode ImageTexture { url "texture.jpg" } ImageTexture { url "light.jpg" }] } textureTransform MultiTextureTransform { textureTransform [...] } } </pre>

- 단계 3 : 단계 1과 2에서 생성된 파일들에서 각 물체에 표현되어 있는 텍스처 노드, 변환 노드 및 텍스처 좌표 노드를 결합하여 각각 다중텍스처 노드, 다중변환 노드 및 다중텍스처좌표 노드로 만들어 최종 파일을 만든다. 각 뷰어별로 만들어진 최종 파일의 예는 표2에 표현되어 있으며 지면 관계상 바닥 부분의 다중텍스처 및 다중변환 노드의 주요 부분만 보여주고 있다.

[그림 4]는 위와 같은 단계를 거쳐 실내 내부를 나타내는 VRML 콘텐츠의 화질을 향상시킨 실험 예를 보여주고 있다. (a)는 기존의 방식으로 다중텍스처링을 하지 않고 뷰어에서 렌더링된 결과이며 (b)~(d)는 각각의 뷰어에서 다중텍스처링을 적용하여 렌더링된 결과로서 바닥 외에는 완전한 다중텍스처링을 적용하지 않은 실험적 수준이지만 더 좋은 화질을 얻을 수 있었다.

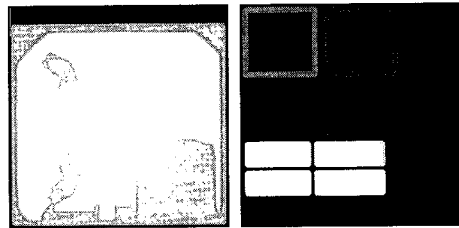
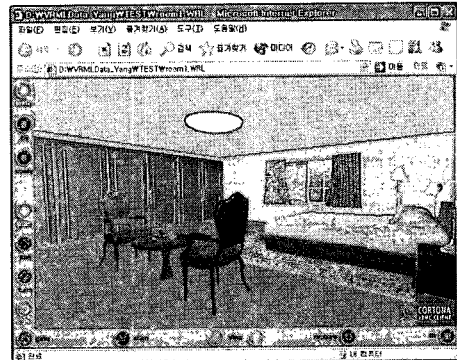
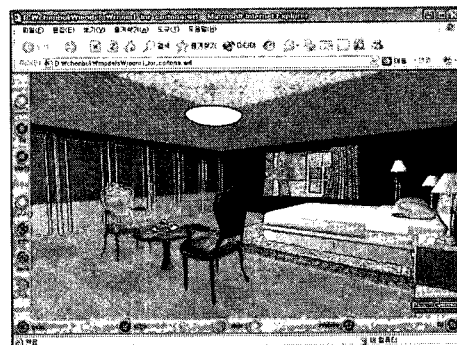


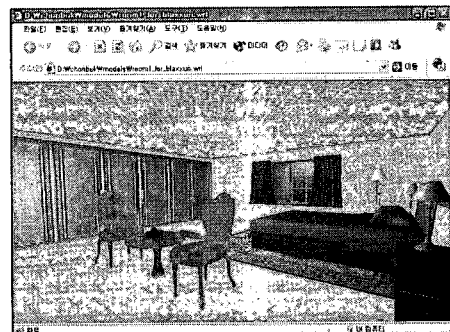
그림 3. 바닥과 옷장의 라이트맵



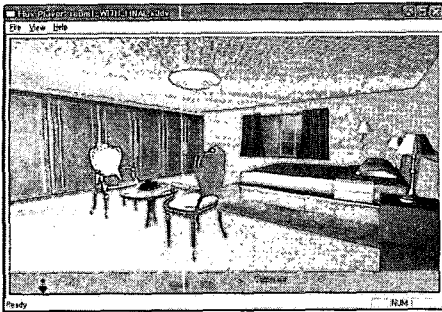
(a) Cortona에서의 기존 방식



(b) Cortona에서의 멀티텍스처링



(c) BlaxxunContact에서의 멀티텍스처링



(d) Flux에서의 멀티텍스처링

그림 4. 각 뷰어에서의 렌더링 결과

IV. 논의 및 향후 과제

본 논문에서는 셰이딩과 다중텍스처링의 관계를 이용하여 텍스처를 동적으로 변경하여야 하는 실시간 X3D 콘텐츠의 화질을 향상시키기 위하여 다중텍스처링 기법을 소개하고 X3D의 다중텍스처 노드를 지원하는 최근의 X3D 뷰어들에 적용하여 실험함으로써 이들 뷰어의 활용 방법을 보였다. 다중텍스처링은 라이트맵뿐만 아니라 적은 양의 3D 기하적 자료를 가지고 표면의 굴곡을 더 부드럽거나 울퉁불퉁하게 보이게 하기 위하여 표면의 수직벡터를 표현한 범프맵(bump map)[13]의 결합에도 많이 이용되는데 실험에 포함되지 않았다. 최근의 DirectX를 지원하는 그래픽카드의 경우 동시에 8개의 다중텍스처링이 가능하므로 표면의 굴곡을 조절해야 하는 물체가 있는 경우에 라이트맵뿐만 아니라 범프맵과 같이 다른 맵을 활용하여 속도를 저하시키지 않으면서 X3D 콘텐츠의 화질을 더 향상시키는 것도 필요하다.

실험 결과 최근의 X3D 뷰어들은 다중텍스처링의 구현에서 호환성이 없고 서로 다른 제약사항을 가지고 있었다. 특히 다중텍스처 노드와 쌍으로 사용되는 다중변환 노드와 다중텍스처좌표 노드는 정확한 다중텍스처링을 위하여 매우 중요한 것이나 이들 뷰어에서는 다중텍스처좌표 노드가 지원되지 못하고 있다. 즉 III.2 절의 단계 2에서 얻어지는 라이트맵의 텍스처좌표값은 이들 뷰어에서 활용할 수 없고 바닥과 같은 간단한 물체 의

에 복잡한 물체는 라이트맵을 정확히 결합시키기 매우 어렵다. 따라서 이들 뷰어에서 멀티텍스처링 기법을 활용할 때 복잡한 물체의 경우 라이트맵을 사용하기 위해서는 디자인 단계에서 평면같이 간단한 물체로 분리하는 작업이 추가적으로 필요하게 된다. 따라서 궁극적으로는 라이트맵의 정확한 결합을 위해서 다중텍스처좌표 노드를 지원하는 뷰어를 개발하여야 한다.

또한 보통 실내디자인과 같이 전문적인 조명에서는 VRay와 같은 플러그인 또는 LightScape와 같은 툴을 연동하기도 하지만 이와 같은 초고급 조명이 이들 뷰어의 화질 한계 안에서는 의미가 없다. 이러한 한계는 기술적인 면뿐만 아니라 3D 애니메이션과 게임 등과 같이 실시간으로 렌더링이 매우 중요한 뷰어의 선택적인 면에서 오는 사항도 있다. 예를 들어 뷰어의 개발 후에 DirectX나 OpenGL 사양에 따라 그래픽카드의 가속 파이프라인이 업그레이드되었지만 이를 충분히 활용하고 있지 못하거나 자주 사용되지 않는 X3D의 수많은 다른 기능도 구현에 포함시킴으로써 오버헤드가 발생할 수도 있는 것이다. 특히 X3D는 기존의 VRML에 이미 많은 노드를 추가하였고 새로운 기능을 위하여 노드를 확장할 수 있는 융통성이 있다. 향후에도 업그레이드되는 그래픽가속기를 활용하고 새로운 노드를 지원하기 위해서는 X3D 뷰어도 같이 업그레이드되어야만 한다. 특히 파이프라인 안에 직접 다른 텍스처링과 셰이딩 프로그램을 적재하여 수행시킬 수 있는 사양이 그래픽스 라이브러리 및 카드에서도 지원되고 있으며 X3D에서도 이를 위한 노드를 사양에 포함시켜 놓고 있다. 적재될 수 있는 대표적인 기능이 물체 표면의 상세 표현을 위하여 표면 법선을 따라 스칼라 변위를 복원하는 변위 매핑(displacement mapping)[14]이다.

따라서 다중텍스처링과 관련된 노드들을 모두 지원하고 다른 오버헤드를 줄이기 위하여 응용에 맞게 주문 제작된 전용 뷰어를 개발함으로써 보다 더 고급스런 라이트맵을 활용하고 상세하게 렌더링하는 방법의 연구가 필요하다. 또한 앞으로 프로그램이 가능한 파이프라인을 가지는 그래픽카드의 대중화가 이루어지고 X3D에서 셰이더 노드라 불리는 이 기능을 지원하는 뷰어를 개발하면 콘텐츠마다 고유의 셰이딩이 가능하고 본 고

에서 다른 X3D 콘텐츠의 화질 향상에도 도약을 가져올 수 있을 것으로 여겨진다.

참고 문헌

[1] <http://www.web3d.org/x3d>
 [2] A. Watt and F. Policarpo, 3D Games Real-time Rendering and Software Technology, Addison-Wesley, New York, p.315, pp.346-348, 2001.
 [3] <http://www.opengl.org>
 [4] <http://www.microsoft.com/directx>
 [5] P. S. Heckbert, "Survey of Texture Mapping," IEEE Computer Graphics and Applications, Vol.6, No.11, pp.56-67, 1986.
 [6] R. A. Hall, "Characterization of Illumination Models and Shading Techniques," The Visual Computer, Vol.2, pp.268-277, 1986.
 [7] A. J. Willmott and P. S. Heckbert, "An Empirical Comparison of Radiosity Algorithms," Technical Report, CMU-CS-97-115, 1997.
 [8] T. Whitted, "An Improved Illumination Model for Shaded Display," Comm. ACM, Vol.23, No.6, pp.343-349, 1980.
 [9] <http://www.knowledgetech.co.kr>
 [10] <http://www.parallelgraphics.com>
 [11] <http://www.mediamachines.com>
 [12] <http://developer.blaxxun.com>
 [13] J. F. Blinn, "Simulation of Wrinkled Surfaces," Computer Graphics(SIGGRAPH '78 Proceedings), Vol.12, pp.286-292, 1978.
 [14] R. L. Cook, "Shade Trees," Computer Graphics (SIGGRAPH '84 Proceedings), Vol.13, No.3, pp.223-231, 1984.

저자 소개

하 중 성(Jong-Sung Ha)

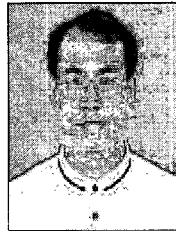
정회원



- 1984년 2월 : 서울대학교 컴퓨터 공학과(공학사)
 - 1986년 2월 : 한국과학기술원 전산학과(공학석사)
 - 1996년 8월 : 한국과학기술원 전산학과(공학박사)
 - 1986년 3월~1989년 2월 : (주)현대전자산업 근무
 - 1990년 3월~현재 : 우석대학교 게임콘텐츠학과 교수
 - 2001년 6월~2002년 2월 : 미국 조지워싱턴대학교 방문교수
- <관심분야> : 응용계산기하학, 컴퓨터그래픽스, 3D 콘텐츠 등임

유 관 희(Kwan-Hee Yoo)

정회원



- 1985년 8월 : 전북대학교 전산통계학과 (이학사)
 - 1988년 2월 : 한국과학기술원 전산학과 (공학석사)
 - 1995년 8월 : 한국과학기술원 전산학과 (공학박사)
 - 1988년 1월~1997년 8월 : (주)데이콤 종합연구소 선임연구원
 - 1997년 8월~현재 : 충북대학교 컴퓨터교육과, 정보산업공학과 및 컴퓨터·정보통신연구소 교수
 - 2003년 7월~2005년 2월 : 미국 카네기멜론대학교 로보틱스연구소 방문교수
- <관심분야> : 컴퓨터그래픽스, 인공지능모델링, 3차원게임 등임