

# Java3D 및 GL4Java로 구현한 인터넷 3D 기술 응용

김병수<sup>\*</sup> · 강병익<sup>\*\*</sup>

## 요 약

다양한 3D 구현 기법들이 웹 상에서 구현되고 있다. 본 논문에서는 웹에서의 애플릿 구현에 필수적인 Java 언어를 근간으로 하는 2개의 API - Java3D 와 GL4Java - 를 이용하여 웹상에서의 3D 구현 방법에 대하여 논한다. 그리고 3차원쇼핑몰 내비게이션이나 원하는 물체의 실시간 3차원 렌더링 등을 각각 Java3D 및 GL4Java를 이용하여 구현한다. 특히 Java3D에서의 객체의 구현은 널리 이용되고 있는 VRML97, OBJ, 3DS 등의 화일 포맷을 Java3d 로더를 이용한 방식으로 구현하여 다른 방식과의 차별화를 꾀한다. 또한 Java3D 과 GL4Java의 장단점에 대한 상호 비교 연구 결과를 제시한다.

## The Application of Internet 3D Technologies using Java3D and GL4Java

Byungsoo Kim<sup>\*</sup> and Byung-ik Kang<sup>\*\*</sup>

## ABSTRACT

Various virtual reality methods are examined on the internet. In this paper, 3D implementations on the internet are discussed using Java3D and GL4Java which are based on the Java. 3D shoppingmall, real time 3D renderings, and various objects are implemented using Java3D and GL4Java, respectively. Especially with Java3D, the file formats of VRML97, OBJ, and 3DS are implemented using Java3D loader, which shows that object implementation in Java3D is better than other methods. Also, the comparisons between Java3D and GL4Java are discussed.

**Key words:** Java3D, GL4Java, 인터넷, 3D

## 1. 서 론

최근들어 컴퓨터 응용 소프트웨어에서 널리 쓰이는 가상현실(virtual reality) 기법은 기존의 오프라인(off line)에서 수행되는 프로그램들뿐만 아니라 인터넷의 발달로 온라인(on line)상에서도 널리 적용되고 있다. 가상현실은 컴퓨터가 제공하는 3차원의 컴퓨터 그래픽 기술을 이용하여 사용자와 온라인으로 상호대화를 하는 기법이라고 간단히 정의할 수 있다 [1,2]. 이 중에서 전자상거래를 위한 현존하는 인터넷 쇼핑몰은 대부분 평면적인 2차원으로 운영되고 있다. 따라서 3차원 세계에서 살고있는 일반 사용자

는 생생한 현실감을 느낄 수 없으며 이는 판매부진의 중요한 요인으로 작용할 수 있다.

실제 매장과 동일한 현실감 및 입체감을 경험하기 위하여는 플래시(Flash)와 같은 플러그인(plug-in) 및 동영상 도입하여 동적인 콘텐츠를 제공하는 방법이 있다[3]. 그러나 이들 방식들은 진정한 의미의 가상현실 기법이라고 볼 수 없다. 따라서 최근 들어 컴퓨터가 제공하는 3차원의 컴퓨터 그래픽 기술을 이용하여 사용자와 온라인으로 상호대화를 하는 기법들이 여러 사이트에서 선보이고 있다[4-7].

본 논문에서는 우선 이러한 기법들의 종류 및 적용방식에 대하여 분류 설명한다. 그런 다음 최근 브라우저상의 애플릿(applet) 표현 방법으로 플랫폼에 구애받지 않고 널리 쓰이는 Java의 3차원 표준 확장

<sup>\*</sup> 정희원, 건양대학교 IT학부 정보전산공학전공 부교수

<sup>\*\*</sup> 건양대학교 IT학부 컴퓨터공학전공 부교수

API(application Programming Interface)인 Java3D를 설명하고 3차원 쇼핑몰로의 적용을 논한다. 선택한 물체에 사진 이미지 기반 방식과 3차원 모델링 방식을 모두 적용하여 보다 현실감 있고 모든 방향으로의 접근이 가능하도록 하였다.

그런 다음 OpenGL 라이브러리를 이용하는 GL4Java에 대한 설명 및 구현을 하여 Java3D와의 비교 분석이 있다.

마지막으로 결론 및 향후 연구과제에 대하여 논한다.

## 2. 웹 3d 기술

웹사이트내의 3차원 그래픽 기법을 적용한 부분은 크게 여러 장의 사진을 촬영한 이미지를 기반으로 한 방법과 3차원 폴리곤(Polygon) 그림을 기반으로 하는 3차원 모델링 기술의 두 가지로 구분 지을 수 있다. 이중 사진이미지 방식은 서로 다른 방향에서 찍은 사진을 이용하여 실사 이미지에 매핑하여 돌려보는 형식으로서 제작과정이 간단하고 질감이 우수하다는 장점이 있다. 그러나 사진이 많아질수록 로딩하는 속도가 느리며 완전한 360도의 3차원 입체를 구현하지 못하는 단점이 있다. 반면에 3차원 모델링 기법은 구현이 어렵지만 실제 3차원 모델을 생성하여 실시간 렌더링으로 구현하므로 상호작용이 요구되는 게임이나 쇼핑몰등에 훨씬 효과적이다[2].

사진 이미지 방식은 회전하는 객체가 사용자인지 물체인지에 따라 파노라마(Panorama)VR과 오브젝트(Object)VR의 2가지 방식으로 나눌 수 있다[2,3]. 파노라마VR은 아바타를 기준으로 주위를 360도 회전하여 촬영한 사진이미지를 편집하여 사용자가 마우스나 키보드 조작 등을 통하여 아바타(avatar) 주위를 회전하면서 살펴볼 수 있는 기법이다. 사진으로 이미지를 만들기 때문에 선명하고 질감이 우수하며 따라서 현실감이 우수하지만 확대 축소 및 특히 y축 이외의 축을 기준으로 한 회전은 사실상 불가능하므로 상당히 제한적인 기법이라고 할 수 있다. 오브젝트VR은 물체 주위로 카메라를 회전하면서 일정 각도마다 찍은 다음 편집하여 돌려볼 수 있게 하는 기법이다. 파노라마VR과 같이 질감이 우수한 반면 한정된 수의 사진 이미지로 인한 회전의 불연속성 및 대개의 경우 y축만의 회전을 보여주고 있다. 물론 x

또는 z축으로의 회전도 가능하지만 추가적인 비용 및 이미지 로딩시간이 상당한 부담으로 남는다.

3차원 모델링 기법은 N-VR, 3D Polygon, Mesh 방식이라고도 하며 사진 이미지가 아닌 3차원 그림 데이터를 이용한다. 3D MAX나 Maya등을 이용하여 물체의 3차원 모델링을 완성한 후 네트워크상에서의 실시간 렌더링을 구현한다. 사진 이미지 방식만큼의 선명한 질감은 제공 못하지만 확대 축소, 재질 및 질감 변화, 원하는 방향 및 각도로의 구현이 자유자재로 이루어질 수 있는 완전한 3차원 방식이며 현재 많은 사이트에서 이 방식을 선호하고 있다. 현재 VRML[12,13], cult 3D[7], Pulse3D, 3DML[14] 등과 같은 플러그인 3차원 모델링 기법이 개발되어 국내 외적으로 여러 사이트에서 널리 쓰이고 있다. 그림 1에서는 Cult3D로 구현한 3차원 브라이언틀링 시계를 보여주고 있다[7].

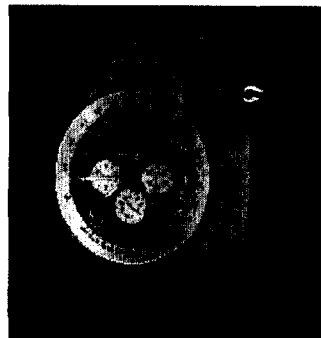


그림 1. cult3d로 표현한 3차원 시계

## 3. Java3D

선(Sun)사에서 플랫폼에 구애받지 않는 인터넷 언어인 Java를 발표한 이후 Java는 지속적인 발전을 해 온 결과 Java2/JDK 1.2 버전을 내놓게 되었다(2001년 10월 현재 JDK1.3). 또한 멀티미디어 프로그래밍 지원을 위한 Java Media와 통신 API들을 개발하였으며 그중의 한부분인 Java3D는 Java2/JDK1.2 플랫폼에 대한 3차원 그래픽 지원을 제공하는 확장 API(application programming interface)이다[8].

Java3D는 3차원 그래픽 프로그래밍을 위한 Java 언어의 표준확장 패키지인 javax.media.j3d 및 javax.vecmath로 제공되는 저수준 3차원 장면 그래프(scene-graph)에 기초한 프로그래밍 API이다. 그림

2에서 장면그래프를 보여주고 있다.

Java3d는 Java 프로그래머에게 인터랙티브한 3차원 그래픽 애플리케이션 및 애플릿을 쉽게 구현할 수 있게 한다. 프로그래머는 그래픽 물체, 빛, 소리 및 다른 객체들과 상호작용하는 객체들을 포함하는 장면 그래프를 만든다. 그런 다음 장면 그래프를 Java3D에게 전달하여 실행시킨다. 그러면 Java3D는 장면 그래프내에 있는 객체들을 렌더링하기 시작한다[9].

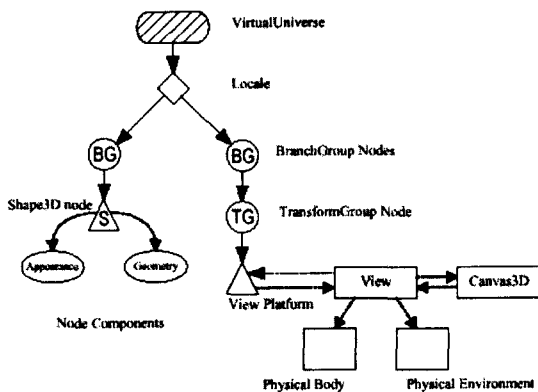


그림 2. Java3D의 장면 그래프

### 3.1 Java3D의 장점

Java3D의 장점을 살펴보면 다음과 같다[15,16].

- (1) Java3D는 장면 그래프에 근간을 둔 3차원 그래픽 모델을 사용하여 고수준의 객체지향 뷰(view)를 제공한다.
- (2) VRML97, 3DS(3d studio), OBJ(Wavefront)를 비롯한 수많은 3D 로더(loader)가 제공되기 때문에 다양한 파일 포맷을 수용할 수 있다.
- (3) Java3D는 주로 네트워크상에서 수행을 위한 3D 그래픽 API 이며 따라서 네트워크상에서 3D 모델과 데이터간의 협조가 가능하다.
- (4) OpenGL이나 Direct3D와 같은 저수준 API를 이용하여 사용자 개별 시스템의 그래픽 하드웨어를 최대한 활용하며, 또한 소프트웨어 그래픽 라이브러리를 이용하여 특정 그래픽 하드웨어없이 Java3D를 수행시킬 수도 있다.
- (5) 통합된 3차원 사운드와 그래픽을 지원하여 가상세계의 현실감을 증대시킨다.
- (6) 복수의 LOD(level of detail)를 제공하여 사용

자로 하여금 가까이 있거나 중요한 물체를 보다 선명하게 관측할 수 있게 한다.

(7) 대부분의 시스템에서 수행이 가능하기 때문에 특정 시스템을 위하여 다시 프로그램을 수정할 필요가 없다. - 플랫폼 프리

(8) 전세계적으로 인터넷언어로서 널리 쓰이는 Java의 3D 그래픽 API이기 때문에 프로그래머의 입장에서 볼 때 매우 친숙하며 쉽고 빠른 3D 프로그래밍을 할 수 있다.

### 3.2 Java3D를 이용한 삼차원 공간구축

Java는 현재 플랫폼에 구애받지 않는 인터넷 언어로 가장 각광받고 있다. 또한 Java Media Api의 하나인 Java 3D는 OpenGL, Direct3D, DirectX와 같은 고급 그래픽 디바이스 라이브러리뿐만 아니라 하위 수준의 디바이스 인터페이스도 지원하여 기존 기술인 VRML등과 달리 보다 일반적인 3차원 환경을 구현할 수 있으며 애플릿과 스윙(Swing)과의 통신이나 3D간의 통신 등 상호간의 높은 인터랙티브를 구현할 수 있는 장점이 있다. 또한 기존의 3차원 웹 구축의 주가 되는 VRML과 연동이 가능하며, 다른 Java Media API, 즉 Sound, Java 2D, JMF, Speech, Advanced Image등 향상된 기술들과의 혼용이 가능하다. 특히 Java 3D에서 지원되는 3D 사운드는 방향과 거리감이 존재하는 보다 현실감 나는 3차원 공간의 구축을 가능케 한다. 특히 새로이 Netscape 6.0과 같은 브라우저에서는 Java 1.3을 기본으로 내장하기 때문에 Java 3D는 인터넷 3차원 구현의 효용성을 높일 것이다.

따라서 본 논문에서는 사진 이미지가 필요한 부분은 자바를 이용하여 구현했으며, 3차원쇼핑몰 내비게이션이나 원하는 물체의 실시간 3차원 렌더링 구현 등은 Java 3D를 이용하였다. 사용자는 Java 3D로 구현된 가상공간내에서 원하는 물체를 클릭하여 오브젝트 VR 및 3차원 모델링으로 구현된 2가지 방식으로 살펴볼 수 있다. 특히 3차원 모델링 방식으로 구현된 제품은 VRML97, OBJ, 3DS 등의 화일 포맷을 Java3d 로더를 이용하여 다양한 방식으로 구현하여 기존 방식과의 차별화 및 성능 향상을 꾀하였다.

### 3.3 객체생성 및 텍스처 기법

본 3차원 가상공간 구현을 위한 Java 3D는 Simple

Universe 가상공간을 장면그래프로 구현한다. 뷰는 기본적인 사용자의 시선을 기준으로 설정하여 가상공간이 구축되었으며 가상공간은 상자(Box)라는 객체에 이미지를 텍스처 매핑하여 구현되었다. 텍스처 기법은 3D객체에 매핑하는 방식과 이미지를 2D로서면에 붙이는 두가지를 사용했다.

절차적 순서는 다음 알고리즘과 같다.

표 1. 객체생성 및 텍스처 매핑 알고리즘

1. SceneGraph를 생성한다
2. Appearance 객체를 생성한다.
3. TextureLoader를 사용해 이미지를 불러와 texture를 생성한다.
4. 이전에 생성된 Appearance에 하나의 컴포넌트로 texture를 세팅한다.
5. Texture된 Appearance 객체 속성으로 생성된 shape를 scene에 삽입한다.

### 3.4 마우스 구현

3차원 공간에서의 내비게이션을 위해 마우스와 키보드를 사용하였는데 Java 3D에서의 마우스와 키보드를 위한 이벤트처리는 Java 1.2에서의 이벤트처리와 많은 차이가 있다.

우선 마우스의 왼쪽버튼으로 드래그되면 좌우의 시선방향이 바뀐다. 그에 대한 실제적 구현은 다음 표 2와 같다.

표 2. 마우스 로테이션 구현 알고리즘

```

MouseRotateY myMouseRotate =
    new MouseRotateY(MouseBehavior.
        INVERT_INPUT);
myMouseRotate.setTransformGroup(vpTrans);
myMouseRotate.setSchedulingBounds(mouseBounds);
objRoot.addChild(myMouseRotate);
    
```

마우스 중앙버튼으로 드래그되면 전후의 이동이 이루어진다. 그에 대한 알고리즘은 표 3과 같다.

### 3.4 상품 상세정보 및 로더

전시된 상품에 대한 상세정보는 또 다른 창으로 보여진다. 이를 구현하기 위해서는 여러 객체에 대해 개별적인 이벤트 처리와 정보에 대한 입출력이 필요하다.

표 3. 마우스 줌 구현 알고리즘

```

MouseZoom myMouseZoom =
    new MouseZoom(MouseBehavior.
        INVERT_INPUT);
myMouseZoom.setTransformGroup(vpTrans);
myMouseZoom.setSchedulingBounds(mouseBounds);
objRoot.addChild(myMouseZoom);
    
```

상세정보의 내용은 두가지로 보여진다. 첫 번째는 오브젝트VR을 사용하여 사진으로 구성된 이미지를 돌려보는 것이고, 두 번째는 3차원 모델링을 이용해 물체를 임의의 각도 및 방향으로 살펴볼 수 있게 한다. Java3D는 특히 3차원 모델링을 이용하는 방법은 여러 파일포맷에 대한 로더가 다양하게 준비되어 있다는 장점을 가지고 있다. 유용한 로더들을 표 4에서 보여주고 있다.

표 4. 현재 유용한 로더 종류

File Format	Description
3DS	3D-Studio
LWS	Lightwave
OBJ	Wavefront
VTK	Visual Toolkit
VRT	Superscape
WRL	VRML

다음은 로더를 이용하는 알고리즘 및 OBJ 파일포맷에 대한 Java3D 그림을 보여준다[10].

표 5. Java3D 로더 알고리즘

1. import the loader class for your file format
2. import other necessary classes
3. declare a scene variable
4. create a loader object
5. load the file in a try block, assigning the result to the scene variable
6. Insert the scene into the scene graph

## 4. GL4Java

SGI가 최초로 만든 OpenGL은 주로 assembler, C, C++ 등으로 구현되는 3D 응용 프로그래밍의 표준 3D API이다. 그러나 운영체제 및 하드웨어에 의존적



그림 3. OBJ 파일 로더로 표현한 Java3D

이며 따라서 웹상에서의 3차원 구현에는 부적절한 API로 인식되었다. Java3D API를 이용하여 웹에서 Java로 구현이 가능하지만 Java3D는 명확하지도 강력하지도 않은 API이며 게임이나 인터랙티브 그래픽스 같은 실세계의 작업에는 부적절한 측면이 많고 아직은 연구수준에 머무르고 있는 실정이다[11]. 반면에 OpenGL은 수많은 응용 프로그램이 시중에 나와 있으며 사용자로부터 이미 철저하게 검증받은 상태이다. 따라서 플랫폼에 구현받지 않는 Java 응용 프로그램에 OpenGL의 뛰어난 성능을 이용할 수 있게 하는 인터페이스가 요구되었으며 Java로 쓰여진 OpenGL 라이브러리인 GL4Java는 이러한 역할을 만족시키는 3D API이다. GtkGLArea, JavaGL, Java OpenGL, Jogl 등과 같은 유사한 OpenGL 라이브러리가 있으나 개발이 중단되었거나 GL4Java 만큼 OpenGL에 대한 완벽한 지원을 하지 못하고 있다 [17].

#### 4.1 GL4Java의 특성

GL4Java의 특성은 다음과 같다.

- (1) GL4Java는 OpenGL 네이티브 라이브러리와 자바가상기계(Java Virtual Machine)에서 수행되는 응용프로그램들간의 인터페이스 역할을 한다.
- (2) Windows, Solaris, Macintosh, Linux 등 다양한 운영체제에서 수행될 수 있다.
- (3) GL4Java API 및 API로 구현된 소스들은 무료이다.
- (4) GL4Java 애플릿들은 MS-IE 및 Netscape에서 구현 가능하다.
- (5) OpenGL 1.2 와 GLU 네이티브 라이브러리의

완벽한 지원을 한다.

(6) OpenGL/glut 로 쓰여진 자바를 지원한다.

(7) C 언어로 쓰여진 OpenGL/glu API로의 액세스가 가능하다.

#### 4.2 객체 생성 및 텍스처 기법

GL4Java로 구현한 3차원 쇼퍼물은 기존의 Java 3D에서 박스(Box)객체를 연결하여 그 객체에 텍스처를 사용하여 꾸며 만든 것과는 조금 다르다. Java3D에서는 박스의 폭, 넓이, 그리고 높이로서 조절하여 벽을 생성하여 만들고 그 객체 전부에 이미지가 입혀졌지만, GL4Java에서는 하나의 점들을 결정하여 다각형을 만들고 그것을 벽의 이미지에 부합되도록 박스형으로 만들어 연결을 한 뒤 단지 하나의 면(다각형)에만 이미지를 입혀 만든 것이다. 그만큼 GL4Java가 Java3D보다 추상화가 덜 되어 있다는 것을 알 수 있지만 반면에 GL4Java가 좀 더 자세한 꾸미기 기법을 추가할 수 있다는 장점이 있다. 예를 들어 Java3D에서는 볼 수 없었던 필터효과가 그것이다. GL4Java에서는 박스전체가 프로그래머가 의도한 색이 들어가지만 하나의 꼭지점마다 색을 넣음으로서 섞여서 나오는 효과도 볼 수가 있는 것이다. 또한 이미지와 색을 같이 씬으로 해서 복합 효과도 볼 수가 있다. 객체 생성 및 텍스처 구현 알고리즘은 다음 표 6과 같다.

표 6. GL4Java에서의 객체 생성 및 텍스처 구현 알고리즘

<ol style="list-style-type: none"> <li>1. 애플릿 창에 Canvas를 생성한다.</li> <li>2. display()메소드안에 객체를 만들거나 메소드 밖에서 원하는 객체를 만들어 불러들인다.</li> <li>3. LoadGLTexture() 메소드에 원하는 이미지와 효과를 부여한다.</li> <li>4. LoadGLTexture() 메소드를 원하는 객체에 호출하여 텍스처를 세팅한다.</li> </ol>
---

#### 4.3 내비게이션 기능 추가

Java3D에서도 움직임을 컨트롤하는 것은 마우스와 키보드였다. 이것들을 사용하여 움직임을 컨트롤하는 것에는 자신을 중심으로 회전하는 것, Z축을 이용한 원거리와 근거리 조절, X축을 이용한 좌우 움직임, Y축을 이용한 위아래 움직임, 그리고 사용자를 중심으로 물체가 회전을 하는 것이다. 여기에서는

GL4Java에서 사용되는 gluLookAt()을 사용하여 보는 시점을 달리하여 원하는 곳을 볼 수 있게 만들었다. 즉, 물체가 움직이는 것이 아니라 보는 아바타(avatar)의 시점을 달리 하는 것이다.

#### 4.4 GL4Java로 구현한 쇼핑물

위의 기법들을 이용한 3차원 골프 쇼핑물을 구현한 것을 그림 4에서 보여준다.

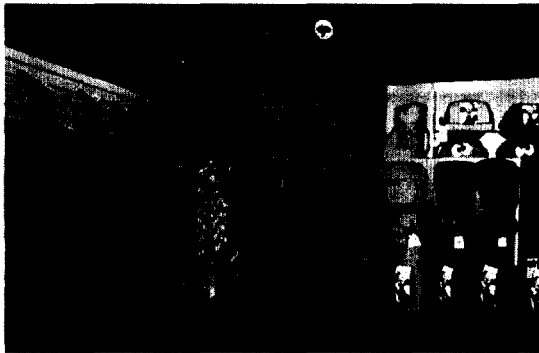


그림 4. GL4Java로 구현한 3차원 골프 쇼핑물

### 5. Java3D와 GL4Java의 분석 및 비교

#### 5.1 간단한 다각형 만들기

두 언어에서 모두 점을 이용하여 다각형을 그릴 수 있다는 것은 비슷하나 Java3D에서는 장면 그래프가 들어간다는 것과 GL4Java에서는 기존의 애플릿 형태를 기본 바탕으로 자신에게 필요한 클래스를 참조하여 프로그램이 이루어진다는 것이 차이점이라 할 수 있다.

다음 표 7에서는 다각형에 색을 넣는 알고리즘의 핵심 코드를 보여준다.

색을 삽입하는 과정에서 Java3D에서는 색을 삽입할 수 있도록 허가하는 코드가 추가되는 것 외엔 형식면에서 두 언어에서 커다란 차이점이 없다. 또한 두 언어에서 색을 삽입할 때 각 점의 색을 달리할 수도 있으며 gradiation의 효과를 만들 수 있다.

#### 5.2 다면체에 대한 event 처리와 texture 기법

Texture부분에서 Java3D와 GL4Java가 많은 다른 점을 보이고 있다. Java3D에서는 단순히 이미지

표 7. 다각형에 색을 넣는 알고리즘

```

- Java3D
public Shape3D makeTriangle() {
    Shape3D result를 생성
    TriangleArray 클래스의 객체 ta를 생성
    ta에 세 개의 Point3d를 생성하여 각각 상속
    ta에 세 개의 Color3f를 생성하여 각 점마다
    색을 삽입하고 상속 result에 ta를 상속
}

- GL4Java
public void display() {
    glTranslatef를 사용하여 위치 설정
    glVertex3fmg 사용하여 box 생성
    glColor3f를 사용하여 각 점마다 색을 삽입
}
    
```

를 Loading하여 Appearance라는 외관을 담당하는 클래스로써 가시적인 효과만을 나타내었다. 하지만 GL4Java에서는 필터(Filter)라는 부분이 더해져서 이미지가 걸러진 모습을 단계적으로 나타낼 수 있으며 그림 5에서 필터링 효과를 보여준다. Cube를 만들 때 Java3D에서는 box라는 클래스가 있어 그것을 이용하여 만드는 반면, GL4Java에서는 GL\_QUADS라는 것으로 일일이 사각형을 붙여서 box를 생성해 낸다. 물론 Java3D에서는 box라는 하나의 클래스에 이미지를 Loading하여 Texture하면 되지만 GL4Java에서는 하나 하나의 사각형에 모두 이미지를 Texture해야 한다.

또한 event부분에서 Java3D는 mouse behavior클래스와 keyboard behavior클래스가 있어 그것을 이용한 몇 줄의 코딩으로 행동이 주어지지만 GL4Java에서는 행동 하나 하나를 계산에 의해 프로그래머가 직접 관찰해야 하는 번거로움이 있다. 이밖에도 마우스나 키보드를 이용한 것이 아닌 연속적인 움직임을 주고 싶다면 Java3D에서는 TranslationInterpolator와 RotationInterpolator를 이용하여 나타낼 수가 있을 것이다.

#### 5.3 공간을 중심으로 회전하는 Navigation

Navigation에서 Java3D는 이미 지원이 되는 클래스인 KeyNavigatorBehavior라는 것이 있다. 따라서 그것을 사용하여 공간을 이동하는 듯한 효과를 줄 수 있는 것이다. 그러나 GL4Java에서는 조금 다르다. Java3D에서 공간의 어떤 점을 중심으로 물체가



그림 5. GL4Java 에서의 필터링 효과

회전하는 것을 이용한 것과 다르게 GL4Java에서는 gluLookAt이라는 클래스를 사용하여 시선이 바뀌는 것을 이용하여 마치 물체가 공간의 어떤 점을 중심으로 회전하는 듯한 효과를 준 것이다. 표 8에서 이에 대한 알고리즘을 보여준다.

표 8. 공간중심 회전 알고리즘

```

- Java3D
public BranchGroup
createSceneGraph(SimpleUniverse su) {
    KeyNavigatorBehavior keyNavBeh = new
        KeyNavigatorBehavior(vpTrans);
    keyNavBeh.setSchedulingBounds(new
        BoundingSphere(new Point3d(),100.0);
    moveGroup.addChild(keyNavBeh);
}
- GL4Java
public void display() {
    glu.gluLookAt(posx, posy, posz, posx+lookx,
    posy+looky, posz+lookz, 0.0f, 1.0f, 0.0f);
}
    
```

5.4 구현된 쇼핑물 비교

이와 같이 Java3D와 GL4Java로써 쇼핑물을 구현한 예제 결과를 그림 6에서 보인다. GL4Java에서는 Texture를 Java3D처럼 많이 사용하면 Loading 시간 뿐만 아니라 Navigation이 원활히 이루어지지 못하는 것은 물론 시스템에도 많은 어려움을 준다. GL4Java에서 이미지를 Loading하는 것이 아닌 Texture 그 자체를 만드는 것이 있는데 구현은 어렵지만 넓은 공간을 만들고 거기서 빠른 이동을 하고 싶을 때 그것을 사용하면 보다 좋은 효과를 얻을 수 있다. 반면에 Java3D는 초기에 Loading되는 시간이 상대적으로

로 더 길다는 것을 알 수 있으나 상대적으로 많은 Texture를 사용할 수 있으며 다양한 로더를 이용하여 물체를 모델링 할 수 있다는 장점이 있다.

6. 결 론

본 논문에서는 다양한 플랫폼에서 적용 가능한 Java 언어를 근간으로, 설계 및 구현에서 뛰어난 장점을 가지는 Java3D 및 실세계의 표준 3d API인 OpenGL을 이용한 GL4Java 프로그래밍을 통하여 3차원 가상공간을 구현하였다. 편리하고도 직관적인 내비게이션을 할 수 있고 사실적인 이미지 부착등을 통한 가상 현실에의 완성도를 증대시켰으며 또한 3차원 모델링 기법을 사용하여 가상제품에 대한 현실감을 극대화하였다.

Java3D는 전세계적으로 인터넷언어로서 널리 쓰이는 Java의 3D 그래픽 API이다. 장면 그래프에 근간을 둔 3차원 그래픽 모델을 사용하며 다양한 파일 포맷에 대한 현존하는 로더를 사용할 수 있는 장점이 있는 반면, GL4Java는 Java3D와 달리 이미 업계에서 검증된 3D 그래픽의 표준 API인 OpenGL 라이브러리를 지원하는 장점이 있다. 간단한 다각형이나 소수개의 객체에 대한 구현에는 GL4Java가 초기 로딩 및 내비게이션에 대하여 조금 나은 효과를 보이지만, 다량의 텍스처 사용이나 복잡한 객체의 구현등에 있어서는 다양한 로더를 활용할 수 있는 Java3D가 더 나은 결과를 보여준다.

추후 과제로는 가상공간을 현실화하여 여러 분야에 적용할 수 있도록 다양한 기능들을 추가하여 보다 뛰어난 사용자 인터페이스를 제공해야 하겠다. 또한 가상공간등에 국한되지 않는 다른 응용분야에의 연구도 병행되어야 하겠다.

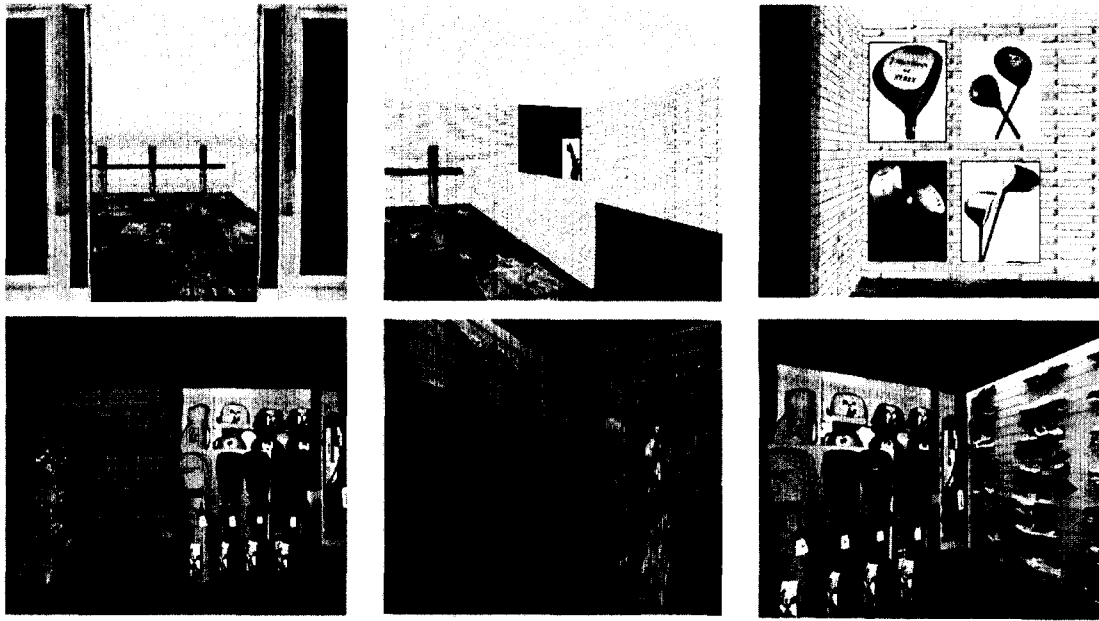


그림 6. Java3D(위) 및 GL4Java로 구현한 쇼핑몰

참 고 문 헌

[ 1 ] [www.dreamscape.co.kr/newhome.htm](http://www.dreamscape.co.kr/newhome.htm)  
 [ 2 ] [www.web3d.co.kr/x3d/faq/index.html](http://www.web3d.co.kr/x3d/faq/index.html)  
 [ 3 ] [www.shockwave.com/sw/downloads](http://www.shockwave.com/sw/downloads)  
 [ 4 ] [www.imoveinc.com/home-spherical/00Introduction.asp](http://www.imoveinc.com/home-spherical/00Introduction.asp)  
 [ 5 ] [www.apple.com/software](http://www.apple.com/software)  
 [ 6 ] [www.ipix.com/products/products.shtml](http://www.ipix.com/products/products.shtml)  
 [ 7 ] [www.cult3d.com/news/default.asp](http://www.cult3d.com/news/default.asp)  
 [ 8 ] [www.gamasutra.com/features/programming/19990115/java3d\\_01.htm](http://www.gamasutra.com/features/programming/19990115/java3d_01.htm)  
 [ 9 ] L. Rosenblum, M. Macedonia, "The Java 3D API and Virtual Reality", IEEE Computer Graphics and Applications, May/June 1999.  
 [10] [java.sun.com/products/java-media/3D/collateral](http://java.sun.com/products/java-media/3D/collateral)  
 [11] [3d.linart.krakow.pl/OfficinaArtificialis/OpenGL](http://3d.linart.krakow.pl/OfficinaArtificialis/OpenGL)  
 [12] H. Baerten and F. Reeth, "Using VRML and Java to visualize 3D Algorithm in Computer Graphics Education," Computer Networks and ISDN Systems, Vol.30, pp.1833-1839, 1998.  
 [13] N. El-Khalili and K. Brodli, "Surgical Training on the Web," Future Generation Computer Systems, Vol.17, pp.147-158, 2000  
 [14] Tim Bray, "An Introduction to 3DML," [www.xml.com/pub/1991/01/3dml](http://www.xml.com/pub/1991/01/3dml)  
 [15] [www.geocities.com/SiliconValley/Horizon/6481/Java3D.html](http://www.geocities.com/SiliconValley/Horizon/6481/Java3D.html)  
 [16] [www.gamasutra.com/features/programming/19990115/java3d\\_01.htm](http://www.gamasutra.com/features/programming/19990115/java3d_01.htm)  
 [17] <http://html.com/cgi-bin/directory?dir=/Computers/Programming/Graphics/Libraries/OpenGL/>  
 [18] [www.jausoft.com/gl4java](http://www.jausoft.com/gl4java)  
 [19] <http://gl4java.sourceforge.net/docs/overview>





김 병 수

1981년 서울대학교 수학과(이학사)  
1987년 미 오클라호마 대학교 대학원  
전산학과(이학석사)  
1993년 고려대학교 대학원 전산학  
전공(이학박사)  
1999년 미 UC Irvine 인지과학과  
객원교수

2001년~현재 건양대학교 학술정보지원센터장  
1987년~1991년 한국과학기술연구원 시스템공학센터 연구원  
1991년~현재 건양대학교 IT학부 정보전산공학전공 부교수  
관심분야 : 병렬처리, 정보검색, 가상현실, 멀티미디어



강 병 익

1984년 한양대학교 전자공학과  
(공학사)  
1986년 한양대학교 대학원 전자  
공학과 (공학석사)  
1992년 한양대학교 대학원 전자  
공학과 VLSI CAD전공  
(공학박사)

1998년 미 San Jose State University 방문교수  
2001년~현재 건양대학교 창업보육센터장  
1991년~현재 건양대학교 IT학부 컴퓨터공학전공 부교수  
관심분야 : VLSI CAD, FPGA, GIS, 인터넷 기술