

碩 士 學 位 論 文

指導教授 李 箕 遠

모바일과 웹에서의 3D 공간자료
연계활용에 관한 연구

Interoperable Application of
3D Geo-spatial features on Web3D and Mobile3D

2008年 12月 日

漢城大學校 大學院

情報시스템工學科

情報시스템工學專攻

董 雨 哲

- 목차 -

| | |
|---|----|
| 제 1 장 서론 | 1 |
| 제 1 절 연구 배경 | 1 |
| 제 2 절 연구 목적 | 4 |
| 제 2 장 적용 기술 | 7 |
| 제 1 절 eXtensible 3 Dimension (X3D) | 7 |
| 제 2 절 Mobile 3D Graphics (M3G) | 11 |
| 제 3 절 Keyhole Markup Language (KML) | 19 |
| 제 3 장 시스템 설계 및 구현 | 22 |
| 제 1 절 GIS 3D 가시화 모델링을 통한 기초 연구 | 22 |
| 1. 기초 연구 시스템 설계 | 22 |
| 2. 기초연구 시스템 시험 구현 | 23 |
| 제 2 절 시스템 설계 | 26 |
| 1. 시스템 구현 | 26 |
| 2. m3g 파일 처리 | 27 |
| 3. 연구 처리 과정 및 결과 | 30 |
| 제 4 장 결론 및 향후 연구 | 34 |
| 참고문헌 | 36 |
| Abstract | 38 |

<표 차례>

| | |
|------------------------------------|----|
| 표 1. The main classes in M3G | 16 |
| 표 2. Children of <PlaceMark> | 20 |
| 표 3. 개발환경 및 운영환경 | 27 |
| 표 4. KML-X3D 태그 매칭 | 30 |

<그림 차례>

| | |
|---|----|
| 그림 1. 현재 모바일 환경에서의 공간정보처리 | 3 |
| 그림 2. 복합형 3D 공간객체의 저장 및 가시화 시스템 | 5 |
| 그림 3. X3D File Structure | 8 |
| 그림 4. X3D Profile | 10 |
| 그림 5. 모바일 3D 적용을 위한 M3G Scene Graph 모델 | 13 |
| 그림 6. M3G의 동작 개요 | 14 |
| 그림 7. Geometric classes for Mesh | 18 |
| 그림 8. 데스크탑의 구글 어스와 모바일의 구글 맵스 실행 예시 화면 | 19 |
| 그림 9. 기본 3D 객체 자료구조 | 22 |
| 그림 10. m3g 파일 내부구조 | 24 |
| 그림 11. 모바일 에뮬레이터에서의 3D 객체 가시화 결과 | 25 |
| 그림 12. m3g 파일 내부구조 예제 : Section 0 | 28 |
| 그림 13. m3g 정보 처리 프로세스 소스코드 | 31 |
| 그림 14. kml/kmz과 x3d 매칭 처리 화면 | 32 |
| 그림 15. 3D file 변환: KMZ (Google Earth)-M3G (Mobile3D:JSR184) | 33 |

제 1 장 서론

제 1 절 연구 배경

웹 2.0 패러다임의 등장은 웹을 중심으로 하는 컨버전스를 가속화시켰다. 웹의 개념이 기존의 어플리케이션 등의 다양한 서비스를 사용자가 선택하여 새롭게 만들고 사용할 수 있는 플랫폼을 제공하는 것, 즉 플랫폼으로서의 웹으로 인식되고 있는 것이다(O. Tim, 2006). 웹 2.0은 사용자의 참여와 개방, 공유를 전제로 하고 있다. 따라서 일방적인 제공이 아닌 누구라도 자신이 필요로 하는 정보에 대해서 웹을 통해서 얻을 수 있어야 하고 공유해야 한다. 물론 사용자뿐만 아니라 다양한 플랫폼에 대한 통합을 위한 기술도 있어야 한다. 웹2.0은 기술이라기보다는 패러다임이다. 공간정보를 다양한 웹 환경에서 보길 원하고, 보여주는 것 또한 같은 맥락으로 볼 수 있겠다.

모바일 분야에서도 언제 어디서나 사용가능한 모바일의 특징과 기존의 웹2.0이 결합된 모바일 웹 2.0의 개념이 등장했는데 이런 배경에는 모바일 단말의 성능이 발전하면서 과거의 단말들이 가지던 성능의 한계들을 줄여가고, 무선 인프라의 발달, 와이브로(Wibro)나 고속하향패킷접속(HSDPA) 등의 이동통신 기술의 발달을 들 수 있으며, 이로 인해 사용자들의 모바일에 기반을 둔 새로운 서비스에 대한 요구 또한 증가 하고 있기 때문이다. 이런 경향 속에서 기존 웹상에서 제공되던 공간정보 또한 모바일을 통한 공간정보로의 수요로 이어 질 수 있으며, 특히 이런 공간정보의 가시화를 위한 모바일 3차원 그래픽 분야는 모바일 단말기기에 기반을 둔 응용 분야가 증가하면서 향후 중요한 기술 분야로 발전할 것으로 예상된다(Malizia, 2007; Pulli, 2008). 모바일 3차원 그래픽 분야는 GIS 분야에서도 웹 3차원 그래픽 분야와 함께 향후 시장에서의 발전 가능성이 높으며, 데스크탑과 모바일 단말을 통한 웹

기반 공간정보의 사실감 있는 영상을 제공할 수 있는 영역이다. 이러한 웹과 모바일의 기술 발전은 실생활이 이루어지는 공간을 모델링하여 컴퓨터를 통해 공간데이터를 구현하려는 사용자들의 요구를 증가시켰고, GIS 분야에서도 공개 표준 및 Open API 등을 가지고 웹 브라우저를 통해 사용자가 쉽고 편하게 위치정보 등을 활용할 수 있는 새로운 GIS의 패러다임인 Where 2.0 이 출현하게 되는 계기가 되었다. 사용자는 자신의 목적에 맞게 지리정보를 활용할 수 있는 환경이 필요해지고 있는 것이다. 웹과 모바일을 통한 3차원 그래픽은 이런 사용자의 수요와 지리정보를 활용하여 공간정보를 필요로 하는 다양한 사용자 중심의 서비스로의 창출이 가능해지게 해준다.

이런 공개 표준 및 공개 API들 중 웹과 모바일에서의 3차원 그래픽의 가시화를 위한 공개 표준 및 공개 API는 데스크탑 환경에서는 W3C(Web3D Consortium)에서 개발된 eXtensible 3 Dimension(X3D)와 OGC(Open Geospatial Consortium)에서 공표한 구글(Google)의 Keyhole Markup Language(KML), 모바일 환경에서는 JSR184/297 3D 그래픽 라이브러리 API인 Mobile 3D Graphics API(M3G) for Java ME(Micro Edition)등이 대표적이다. 그리고 이런 표준안들을 적용함으로써 GIS 정보 및 기술을 일반 사용자 환경에서 용이하게 처리 할 수 있게 되며, 다양한 사용자와 사용 환경에서의 상호운용이 용이하다. 또한 이를 기반으로 하는 다중 환경에서의 공간자료 처리를 위한 표준안들의 상호운영의 관한 연구는 거의 보고된 바 없으며, 현재는 주로 모바일 환경에서의 공간정보 처리는 2D 공간객체나 영상 콘텐츠 처리와 가시화가 주요 동향이다[그림 1]. 이는 단말 환경의 조건이 3D 공간 객체 가시화에 적합하지 않거나 실무적인 적용성의 한계에 기인한다.



그림 1. 현재 모바일 환경에서의 공간정보 처리

제 2 절 연구 목적

모바일과 웹에서의 3차원 그래픽스는 각각의 목적에 따른 산업계의 요구도 다르고 다양한 표준안들이 존재한다. 3차원 GIS 분야의 경우에도 예외는 아니다. 그래서 본 연구에서는 이런 다양한 표준안들 중 모바일 그래픽스 API 중 다른 API와는 다르게 .m3g 형태의 파일 포맷을 제공하는 M3G API와 최초의 웹 3차원 구현을 위한 VRML(Virtual Reality Markup Language)에 XML이 결합된 X3D와 웹상에서 구글의 맵핑 서비스를 가능하게 해주는 XML 기반의 .kml/.kmz 파일 포맷 형태의 KML을 활용하여, 데스크탑 환경 및 모바일 디바이스 환경 즉 서로 다른 환경에서의 동일한 3차원 공간정보를 처리할 수 있게 하는 3차원 공간정보 연계활용에 관한 기초 연구를 진행하였다.

관련 연구로 그림 2의 Lee and Kim(2006)에서는 PDA 환경에서 OpenGL|ES를 기반으로 하여 3차원 공간 객체 처리와 가시화를 위한 기본 환경을 구축하고 처리 프로세스를 체계화한 바 있으나, 모바일 휴대폰에서 M3G 기반의 3차원 공간객체를 가시화 할 수 있는 시스템은 구체적으로 연구된 경우가 거의 없다.



그림 2. 복합형 3D 공간객체의 저작 및 가시화 시스템

그리고 최근 관심이 높아지고 있는 CityGML은 Virtual 3D City 모델의 저장과 교환을 위한 XML 기반의 데이터 모델로, 공간자료의 변환을 위한 표준으로 OGC에 의해 표준안으로 채택되었는데, CityGML의 주된 목적은 사용자에게 적합한 공간정보를 효율적으로 제공해주기 위하여 처리 수준이 다른 분야의 정보를 통합 처리함으로써 통합 및 연계를 제공한다.

CityGML은 도시의 3차원 객체의 표현을 위한 일반정보모델(Common Information Model)로 정의할 수 있는데, 도시에 관련된 지형과 지역에 대한 객체간의 관계나 특성들에 대해 그래픽의 교환 형식을 떠나 다른 응용프로그램에서도 분석이나 시뮬레이션 등의 작업이 가능하도록 3차원 모델을 사용할 수 있다. 기존의 3차원 모델들이 가시화만을 위한 목적으로 사용되었다면 CityGML은 3차원 모델의 분석 작업이나 공간자료의 마이닝 등이 가능하며 다른 응용프로그램과의 공유를 위하여 개발된, 진행 중인 공간모델의 패러다임 변화라 할 수 있다 (Thomas, 2007). 그리고 현재 진행 중인 WFS상의 CityGML을 W3DS상의 X3D와 KML /COLLADA 으로의 변환은 현재 진행하려는 연구와 XML을 기반 하는 처리 부분이 일부분 동일하기 때문에 향후 진행될 연구에서 CityGML의 활용부분도 검토하여 보다 안정적이고 확장 가능한 모바일과의 호환을 가능하게 진행할 수 있다.

모바일과 웹에서의 이런 파일 교환 및 각대상의 변환에 대한 기술들의 활용을 위한 검토를 통하여 3차원 GIS의 사용가능한 분석 자료로 활용하고자 한다. 먼저 기존의 M3G API를 활용한 3차원 공간 정보 처리 시스템을 위한 선행 연구는 없었기 때문에 M3G API를 활용한 GIS 3차원 가시화 모델링 생성 및 활용을 위한 기초 연구를 진행하였고 또한 향후 모바일에서의 3차원 웹 응용프로그램을 위한 각 표준안의 모바일과 웹 환경에서의 3차원 공간자료의 연계활용에 관한 연구 및 시험을 수행하였다.

제 2 장 적용 기술

본 연구에서는 모바일과 웹에서의 3차원 공간자료 연계활용에 관한 연구를 위해서 3차원 공간정보의 가시화를 위한 각 플랫폼에서의 3차원 그래픽 표준을 이용하였다. 웹 환경에서는 KML과 X3D를 이용하였고, 모바일 환경에서는 M3G API를 사용하여 처리 하였다. 그러나 이런 표준들은 각각의 환경에서만 가시화가 가능하기 때문에 각 플랫폼에서 생성된 공간정보에 대해서 다른 환경에서의 가시화 하기는 쉽지 않다. 그래서 서로 다른 환경에서 데이터를 공유하기 위해서는 추가적으로 각각의 표준에 맞게 호환이 가능하도록 별도의 변환작업이 요구된다. 본 연구에서는 각각의 목적에 맞는 데이터들 Keyhole Markup Language(KML), eXtensible 3D(X3D), Mobile 3D Graphic(M3G) 간의 시험 변환을 통하여 서로 다른 환경에서의 동일한 가시화의 가능성을 검토 하였고 이를 활용하여 3차원 공간자료의 연계활용에 대하여 연구를 진행 하였다. 그리고 시험 변환이 적용되어진 각각의 포맷들(kml, x3d, m3g)은 전용 브라우저를 통해서 보여 지게 된다.

제 1 절 eXtensible 3 Dimension (X3D)

X3D(eXtensible 3 Dimension)는 웹에서의 3차원 구현을 위한 개방형 표준안이다. 기존의 웹 3차원 구현을 위한 표준 언어인 VRML97을 대체하는 새로운 표준안이며 기본적으로 XML(eXtensible Markup Language)의 형식으로 정의 되어 있다. XML을 기반으로 웹상에서 3차원 물체를 모델링하고 정보를 제공하는 것을 목적으로 하고 구성되어 있기 때문에 다른 콘텐츠들과의 호환성이 높으며, 기존의 VRML이 가지던 하나의 스펙과는 다르게 프로파일과 컴포넌트 별로 분리 구현이 가능하여, 사용자에게는 메모리의 부담을 줄여주며, 필요한 기능은

컴포넌트에 추가해서 사용 하면 되는 효율성을 제공한다. X3D의 기본적인 파일 구조는 그림 3과 같이 표현할 수 있다.

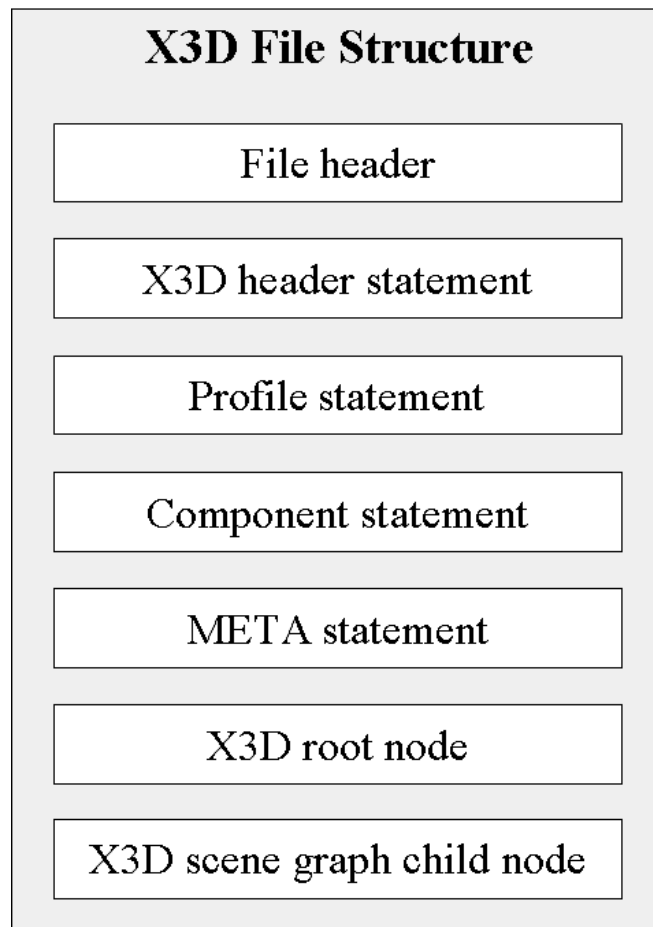


그림 3. X3D File Structure (web3D Consortium)

먼저 “File header”는 X3D scene의 관한 기본 정보들을 포함하고 있는데, 실제 렌더링에 해당하는 노드는 포함하지 않으며, XML 과 X3D의 header들과 프로파일(Profile)과 컴포넌트 등의 정의로 구성되어 있다. “X3D header statement”는 X3D 파일을 식별하는 역할을 하는데, 예를 들면 X3D의 버전과 텍스트 인코딩 방식(X3D의 경우에는 universal text format: UTF-8을 사용) 등이다. X3D 문서만을 작성하고자 할 때는 생략 가능하지만 기본적으로 기존의 버전과의 호환이나 새로운 버전을 적용하는 등에 경우에는 사용해야 한다. X3D는 여러 프로파일을 목적에 맞게 레벨 단위로 적용하여 사용할 수 있는데, 그림 4 는 X3D의 기본 프로파일의 구조와 몇 개의 컴포넌트를 보여준다. “Profile statement”에서는 이런 프로파일을 파일이 실행될 때 해당 노드들을 최소로 지원하기 위한 기준과 포함해야할 컴포넌트와 실행 수준을 정의 한다. 그림에 나타나 있지는 않지만 X3D에는 기능에 따라 6단계의 프로파일로 구분되어지며, 실제적으로 가장 최소단위의 프로파일은 “Core” 이다. 장면을 표현하기 위한 최소단위의 프로파일은 “Interchange” 이며, 최소한의 3차원 물체 렌더링과 장면을 만들기 위해 사용 한다. “Interactive” 프로파일은 일정수준의 그래픽과 간단한 상호작용을 지원한다. 즉 한정된 가상공간의 탐색과 제한된 센서제어를 요구하는 장면을 연출한다. “Immersive” 프로파일은 거의 완벽한 수준의 기능을 제공하는데, 기존의 VRML97의 모든 노드가 이 범위에 속한다. “Full” 프로파일은 X3D의 모든 표준노드를 포함한다. X3D 컴포넌트는 기능별로 분리된 프로파일에 맞게 유사한 기능을 가지는 노드들의 집합이다. 기본적으로 제공되는 컴포넌트를 바로 적용 가능하며 새로운 컴포넌트를 추가함으로 새로운 기능을 확장할 수 있다.

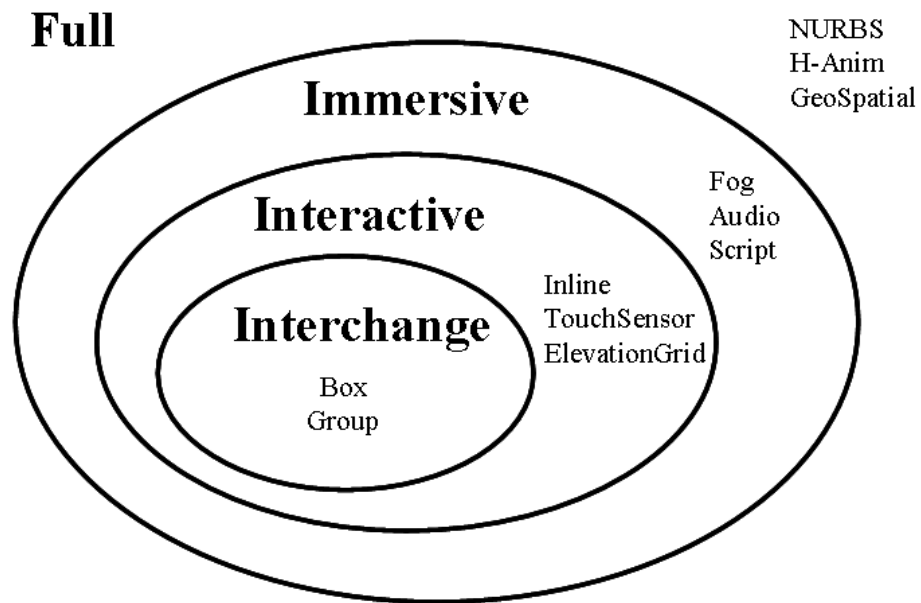


그림 4. X3D Profile

프로파일은 컴포넌트(Component)의 집합이라고 생각하면 되는데 헤더에 정의되는 프로파일에 속하지 않는 기능은 “Component statements”를 통해 지원되며 헤더에 따로 정의할 필요는 없다. 실제 Full 프로파일을 적용할 경우에 이론적으로 별개의 컴포넌트를 정의해서 사용해야 하는 것이 불필요해보이지만, 사실 Full 프로파일의 경우에는 전용 브라우저에서의 사용이 제한적일 수도 있으며, 성능을 위해서도 필요한 기능만 사용하는 것이 유리하다. “Meta statement”는 파일의 대한 정보를 나타낸다. “X3D root node”는 X3D 파일의 가장 상단에 위치하며 반드시 적용되어야 하고, 앞서 언급한 프로파일이나 X3D header statement 등이 포함된다. “X3D scene graph child node”는 실제로 X3D 파일의 렌더링을 위한 여러 기능, X3D의 필드와 노드들로 구성되어 지는 Scene graph 형태의 처리 구조이다.

제 2 절 Mobile 3D Graphics (M3G)

M3G (Mobile 3D Graphics API for Java 2 Micro Edition)는 모바일 환경에서의 3차원 가시화를 위한 자바 기반의 3차원 그래픽 라이브러리 API 이다 [5][8]. JSR(Java Specification Request) - 184 로도 알려져 있으며, 모바일 장치를 위한 프로파일이다. CLDC/MIDP 기반 환경에서의 3차원 그래픽을 위한 경량의 패키지이며, CLDC 기반(부동 소수점 지원)에서 동작하고 MIDP(Mobile Information Device Profile)의 옵션 패키지이다.

M3G는 현재 산업계 모바일 3D API 표준 중 하나이며, 다른 모바일 3차원 API 표준들인 OpenGL ES(Embedded System), Direct3D mobile 과 함께 널리 쓰이고 있다. M3G는 기존의 저수준 형태의 절차적 방식으로 표현되는 OpenGL ES의 모든 기능을 구현할 수 있으며 구조적, 기능적 개선이 가능하다. M3G는 즉시모드(Immediate mode), 보유모드(Retained mode)의 2가지 그래픽 표현방식을 제공하고 있는데 즉시모드는 저수준 형태로 제공되어 기존의 OpenGL ES 등의 그래픽 표현 절차를 제공함으로써 호환이 가능하게 하였고,

Scene Graph구조의 보유모드를 활용함으로써 각 부분의 참조 및 재사용이 가능하여 효율성을 높인 것이 장점이다. M3G는 Scene Graph 구성, 카메라 및 애니메이션 요소 구성이 모두 Java API로 구성되어 있으며, 실제 Scene Graph의 최상위 World 노드의 이벤트 처리, 렌더링 부분도 역시 Java API로 표현되어 있다. 이런 API 형태는 VRML/X3D 와 비슷한 형태로 볼 수 있다. 그리고 이런 3D 표현을 위한 제어 함수를 자체적으로 제공하나 모두 Scene Graph의 구조 내에서 처리되어야만 한다. 모바일 3D 적용을 위한 M3G Scene Graph 모델의 예시는 그림 5와 같고 그림 6은 M3G의 동작 개요를 나타낸다.

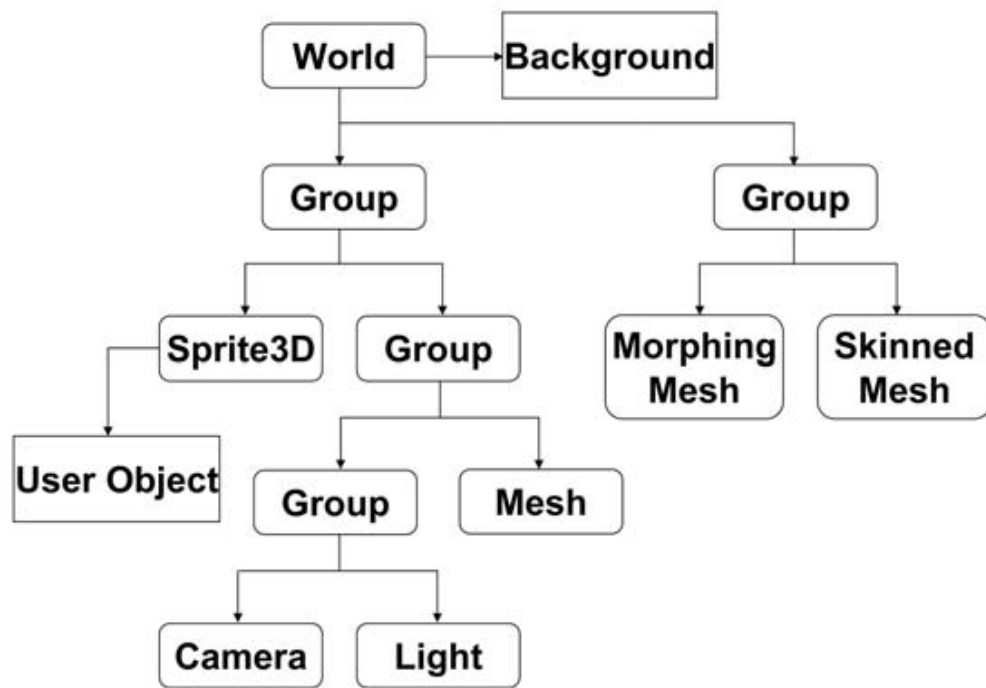


그림 5. 모바일 3D 적용을 위한 M3G Scene Graph 모델
(Malizia, A, 2006)

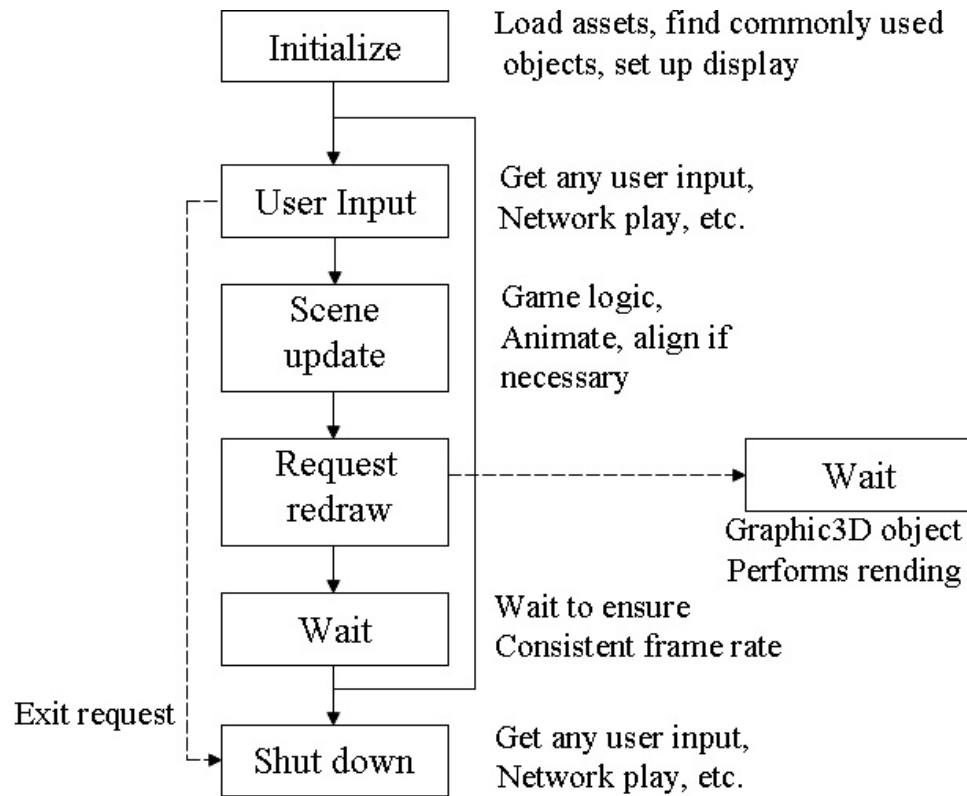


그림 6. M3G의 동작 개요 (M3G API Spec. 2005)

M3G에서의 World, Graphics3D, Loader 노드는 핵심적인 클래스이다. World는 Scene Graph의 최상위 노드이며, Graphic3D는 3차원 그래픽 컨텍스트를 포함하는 상태: frame buffer, depth buffer, viewpoint, hints 등의 렌더링을 위한 것이고, Loader는 .m3g 확장자를 가지는 파일 포맷에서 각 오브젝트와 Scene Graph를 불러온다(Pulli 2005,2008). 이 .m3g 파일은 모든 데이터가 주어진 Scene Graph를 포함한다. Scene Graph에서의 중요한 각 클래스는 표 1로 정리하였다.

⌘ 1. The main classes in M3G (Morales and Nelson, 2007).

| Class | Description |
|---------------|---|
| Background | Defines a background image or color for the render |
| Camera | A node that defines a camera. It is pre programmed with two projection matrices: parallel and perspective. Can be used for custom projections |
| Graphics3D | The main render class for M3G. Capable of retained or immediate rendering. |
| Group | A scene graph node capable of storing other nodes. |
| Light | A scene graph node that can define ambient, directional, omnidirectional, and spotlights. |
| Loader | Imports a .m3g file. |
| Material | Subcomponent for Appearance that defines the material for an object. |
| Mesh | Defines a polygonal mesh. |
| Morphing Mesh | Manipulates meshes that have morph targets. |
| Node | An abstract class shared by all nodes. |
| Object3D | An abstract class from which all nodes capable of existing in the 3D world are derived. |
| Skinned Mesh | A scene graph node that represents a mesh with skeletal animation. |
| Sprite3D | A scene graph node used to define 2D billboards. |
| Texture2D | A subcomponent of Texture2D used to map Image2D objects to Sprite3D and Mesh3D. |
| Transform | A generic class for manipulating transformations. |
| Transformable | An abstract class that gives Sprite3D and Node the capability to store and manipulate transformations. |
| World | A top-level node for scene graphs. |

기본적으로 M3G는 3차원 그래픽 툴을 위한 기본 포맷이기 때문에, 오브젝트 타입과 클래스 간의 1대1의 매핑이 되며 툴을 이용해 만든 데이터를 내보내거나 응용프로그램에서 불러올 수도 있으며, 전용뷰어를 통해 파일을 볼 수도 있다. M3G 파일은 “file identifier”와 하나 또는 여러 개의 섹션으로 구성되어지는데, 모든 M3G 파일의 처음의 12바이트는 file identifier를 나타낸다(<<JSR 184>>). 첫 번째 섹션, section 0은 반드시 존재해야 하며, 반드시 header object를 포함해야 한다. 나머지 섹션들은 모델이 실제로 그려지는 역할을 한다. 그림 7은 실제로 모델이 그려질 때 객체의 모든 정보를 제공하는 역할을 하는 Mesh 노드의 Geometric class들을 도식화 한 그림이다. IndexBuffer는 M3G가 triangle strip을 사용할 수 있는 TriangleStripArray 정보를 제공하며, VertexBuffer는 VertexArray의 정보를 참조하는데 보통 하나의 VertexBuffer는 하나의 VertexArray를 참조하며 이 벡터에 배열의 position, normal, color, texture coordinate의 정보를 참조한다. 최소한으로 참조할 경우에도 반드시 position의 정보는 포함해야 한다.

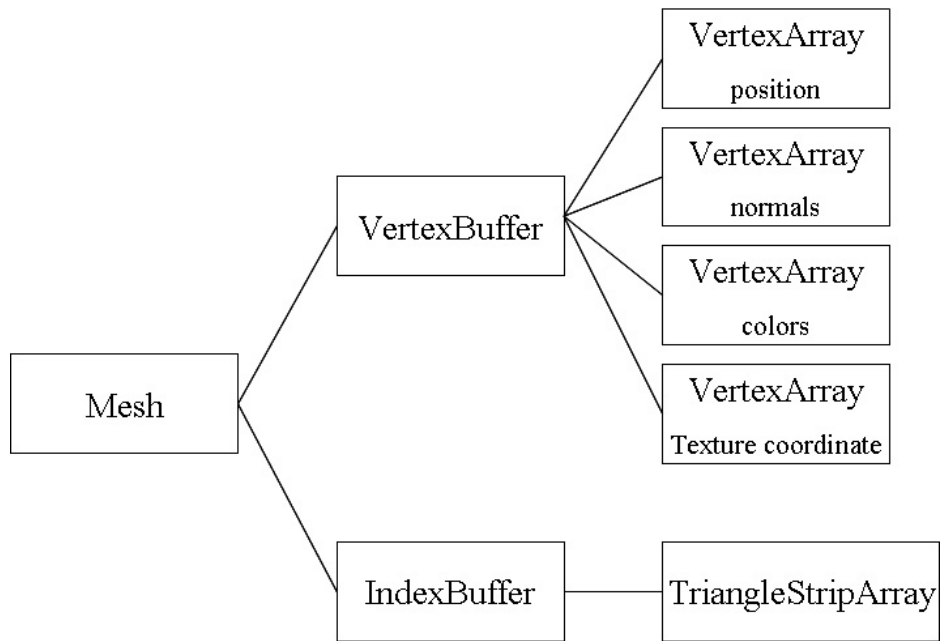


그림 7. Geometric classes for Mesh (Morales and Nelson, 2007)

제 3 절 Keyhole Markup Language (KML)

KML은 XML 기반의 3차원 지리정보데이터의 표현을 위한 언어로 구글 어스, 구글 맵스, 구글 모바일 등 어스 브라우저에서 지리정보를 표시하기 위해 사용되는 파일 포맷이다. KML은 웹 브라우저가 HTML 또는 XML을 처리하는 방식과 유사한 방식으로 처리되며, KML은 화면에 표현되는 목적에 맞는 이름과 속성이 있는 태그 기반 구조이다. 구글 어스와 구글 맵은 KML 파일에 브라우저 역할을 한다 (J.Wernecke, 2008). OpenGIS KML 2.2 Encoding Standard (OGC KML) 라는 이름으로 OpenGIS Consortium에서 표준으로 제정 되었다. 그림 8은 데스크탑과 모바일에서의 구글 어스와 구글 맵스의 실행 예시 화면 이다.



그림 8. 데스크탑의 구글 어스와 모바일의 구글 맵스 실행 예시 화면

KML은 다음과 같은 기능을 제공한다. 지구상의 위치를 표시하기 위한 아이콘 및 레이블을 지정, 표시하고자 하는 지형지물에 대한 카메라 위치 설정(View 정의), 지표면이나 화면에 별도의 영상을 중첩, 스타일을 정의하여 지형지물의 형태를 지정, 하이퍼링크 및 영상을 포함한 지형지물에 대한 HTML 설명문 작성, 폴더 형식을 이용하여 지형지물의 계층구조 지원, 인터넷을 통하여 KML 파일을 동적으로 가져오거나 갱신하기, 구글어스에서 뷰의 변화에 따라 KML 데이터 가져오기, COLLADA 방식으로 텍스처 처리한 3차원 객체의 표시 등이다.

KML 파일을 생성하기 위해서는 아래와 같은 내용이 문서에 꼭 삽입되어야 한다.

```
<?xml version = "1.0" encoding = "UTF-8"?>
<kml xmlns = "http://www.opengis.net/kml/2.2">
```

KML 파일은 <kml> 엘리먼트만을 포함할 수 있으며 파일의 끝은 반드시 </kml>로 닫아야 한다. 파일에는 세 개의 엘리먼트를 가지고 있는 <PlaceMark>가 포함되어 있는데 내용은 표 2와 같다.

표 2. Children of <PlaceMark> (J.Werneck, 2008)

| | |
|---------------|--|
| <name> | Label for the placemark |
| <description> | Text (and optional images) providing additional information about the placemark. The <description> appears in the information balloon pops up when the user clicks the placemark name in the Place panel or the placemark icon in the 3D viewer of Google Earth. |
| <Point> | Contains the <coordinates> element. The <coordinates> element contains values for the longitude, latitude, and altitude of the <Placemark>. See the section in chapter 3 on "Coordinates" for more detail. |

<name>은 해당 PlaceMark의 레이블로 사용되고 <description>은 PlaceMark의 풍선팝업에 나타나는 내용, <Point>는 해당 PlaceMark의

지구상의 위치를 지정한다. (경도와 위도, 높이는 옵션이다)

KML은 XML 표준을 따르는 태그 기반의 구조이기 때문에 일정한 패턴을 가지는 구분하기 쉬운 구조로 되어 있다. 그리고 단순하면서도 가장 널리 사용되는 기능은 PlaceMark 이지만, 다른 도형 요소인 LineString, Polygon, Model 등을 포함하기는 해도 표현 할 수는 없기 때문에 직접 에디터에서 스타일을 정의하고 생성해야 한다.

제 3 장 시스템 설계 및 구현

제 1 절 GIS 3D 가시화 모델링을 통한 기초 연구

1. 기초 연구 시스템 설계

본 연구에서는 Web 3D 처리를 위해 Java ME 환경에서 M3G API를 시험 구현하였다. 3차원 공간정보 처리 및 가시화를 위해서는 성능 최적화 문제를 해결하는 방법이 필요하며, M3G는 객체 간의 참조 및 재사용이 가능하여 효율성을 높인 것이 장점이다. 기존의 M3G API를 활용한 3차원 공간 정보 처리 시스템을 위한 선행 연구는 없었기 때문에 GIS 3D 가시화 모델링을 통한 기초 연구를 진행하였다. 그림 9는 확장성을 고려한 기본적인 3차원 객체의 자료구조이다.

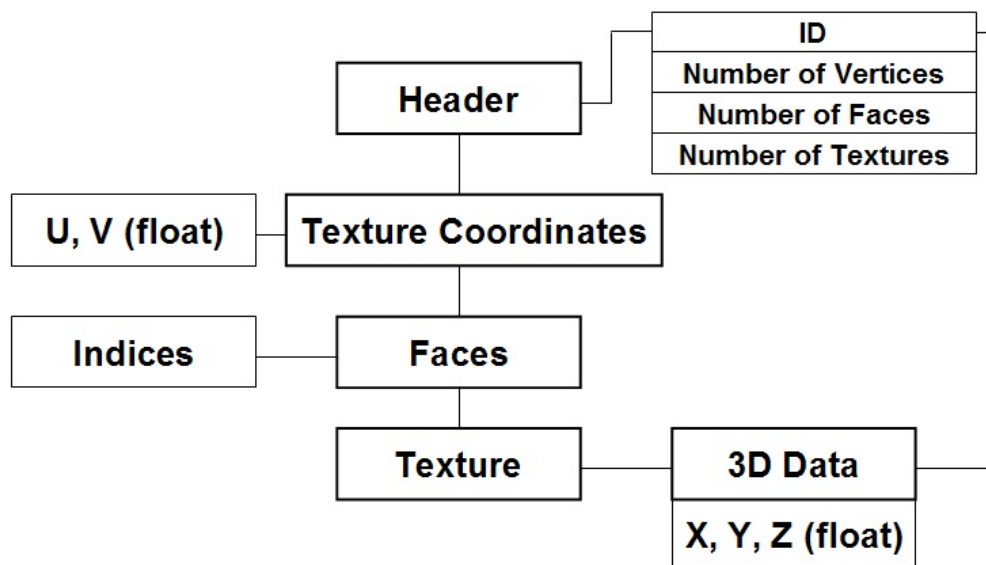


그림 9. 기본 3D 객체 자료구조

2. 기초연구 시스템 시험 구현

M3G의 가시화를 위한 개발 환경은 데스크탑에서 수행 되었고 에뮬레이터를 통한 디버깅과 구현으로 진행하였다. 전체적인 개발환경 및 운영환경은 Windows XP 기반에 JDK 1.5.0_15, NetBean IDE 6.01 mobility pack 그리고 WTK(wireless ToolKit) 2.52(CLDC1.1/MIDP2.0)이며 임의 지역의 영상 정보와 텍스처 매핑된 건물을 구현하였다. 배경과 바닥, 건물 구현 후 카메라 클래스를 이용한 양방향 이동 및 회전을 구현하였고 에뮬레이터를 통하여 구동 하였다.

이렇게 처리된 객체는 .m3g의 포맷으로 내보내기하거나 저장되어 웹이나 다른 3차원 그래픽 소프트웨어 등에서 재사용이 가능한데 이러한 .m3g 파일의 내부구조는 그림 10과 같다.

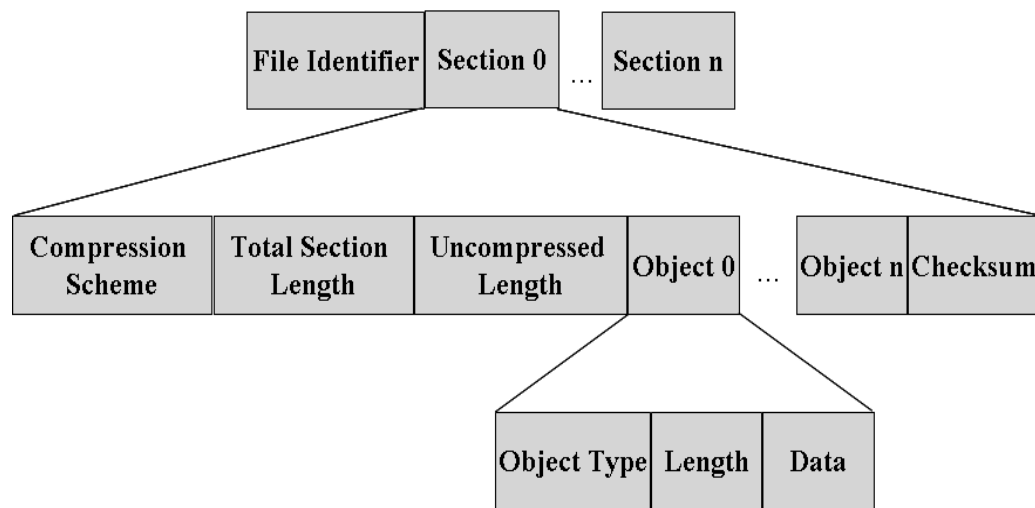


그림 10. m3g 파일 내부구조(Höfele, C., 2007)

.m3g 파일은 File Identifier와 Section0 …… Section n까지의 바이너리 타입의 구조로 되어있으며 가시화를 위해서는 Section에 포함되어 있는 오브젝트의 정보를 통해 접근하게 되고, 이 오브젝트의 요소들을 통하여 Scene Graph 형식으로 정의된다.

가시화 결과 화면은 그림 11과 같으며, 3개의 텍스처 매핑된 건물과 바닥, 배경을 구현하였으나 시점 변화에 의한 성능의 변화는 없었다.



그림 11. 모바일 에뮬레이터에서의 3D 객체 가시화 결과

GIS 3D 가시화 모델링에 있어서는 M3G의 Scene Graph 구조에 의한 모델링은 객체 기반의 3차원 그래픽 처리의 기본 개념과 부합하며 기존 구축된 3차원 모델의 참조와 재사용 문제에서도 효과적일 수 있다.

제 2 절 시스템 설계

1. 시스템 구현

본 연구에서는 1절의 기초 연구를 토대로 시스템의 시나리오를 설계하였다. 기본적으로 KML과 X3D는 XML 표준을 따르는 태그 기반의 언어이며, 그로인해 둘 사이의 상호운영이 이론적으로 가능하다. M3G의 경우엔 .m3g라는 파일의 형태로 저장이 가능하지만 M3G와 KML과의 직접적인 변환은 쉽지 않다. 그러나 M3G와 X3D는 가시화의 방법은 여러 노드들로 연결된 유기적인 형태인 Scene Graph의 구조로 이루어지는 유사한 형태이며 하나의 노드는 하나 또는 다수의 필드로 구성이 가능하고 노드는 또 다른 노드를 포함할 수 있으며, 같은 노드라도 다양한 형태의 3차원 물체를 구현할 수 있다. 즉 이런 노드들의 정보를 매칭 시키게 되면 이론상으로 가능하다. 물론 M3G와 KML의 직접적인 파일 교환이나 변환도 가능하겠지만 직접변환은 쉽지는 않기 때문에 X3D를 통한 개별적인 변환을 통해 사용자가 생성한 3차원 객체에 대한 서로 다른 환경에서의 가시화가 가능하게 하도록 한다. M3G의 경우에는 .m3g 파일의 구조가 바이너리 형태이기 때문에 처리를 위한 임의의 응용프로그램을 통해 .m3g 파일의 Section에서 Object의 가시화에 필요한 정보 값을 얻어야 한다. 그리고 얻어낸 정보 값은 매칭 되는 X3D의 관련 노드들에 대입 해준다. KML과 X3D의 경우에는 KML이 WGS84 좌표계를 사용하기 때문에 X3D에서 "GeoSpatial" 컴포넌트를 사용하여 기존의 TM 좌표계를 WGS84 좌표계로 매칭 시

켜 KML의 좌표 정보를 통해 변환하도록 한다.

표 3은 전체적인 개발 환경 및 운영환경을 나타내고 있다. 모든 환경은 Window XP 기반의 Web 3D의 표준안으로는 KML과 X3D를 이용하였고 Mobile 3D의 표준안으로는 M3G를 사용하였다. 각각의 표준안의 변환과 교환을 위해서 NetBean 6.01과 visual C++ 6.0을 사용하여 개발하였고, 결과화면은 각 API의 뷰어인 Google Earth 와 BS Contact Geo, M3G Viewer4.3 을 통해 가시화 하였다.

표 3. 개발환경 및 운영환경

| | Web 3D | | Mobile 3D |
|-------------|-------------------------------|----------------|-------------------------------|
| | KML | X3D | M3G |
| Environment | Windows XP | | |
| | OGC KML | W3C X3D | JCP JSR-184 |
| | | | JDK 1.50_15 |
| | | | WTK 2.52 (CLDC1.1/MIDP2.0) |
| | Google Earth | BS Contact Geo | HT- M3G Viewer4.3 |
| | NetBean 6.01 / Visual C++ 6.0 | | |
| | MS XML4.0 | | |

2. m3g 파일 처리

바이너리 형태의 .m3g 파일을 접근하기 위해서는 기본적으로 파일의 구조와 정보를 파악해야 한다. 그래서 본 연구에서는 m3g 파일의 주소에 접근하여 원하는 정보를 얻을 수 있는 일종의 처리기를 구현하였다. 먼저 파일의 접근은 File Identifier와 Section0 …… Section n까지의 각 Section에 포함되어 있는 오브젝트의 정보를 통해야 한다. 최초의 File Identifier는 처음 12바이트로 모든 m3g 파일에 무조건 포함된다 (AB 4A 53 52 31 38 34 BB 0D 0A 1A 0A).

Section은 하나 또는 여러 개의 오브젝트들의 데이터 저장소 역할을 한다. 그림 12는 임의의 m3g 파일의 section 0의 구조이며 첫 번째 섹션, Header Object로 File Identifier를 포함하는 섹션 데이터로부터의 정보(compression scheme, total section length, uncompressed length, checksum)를 나타낸다.

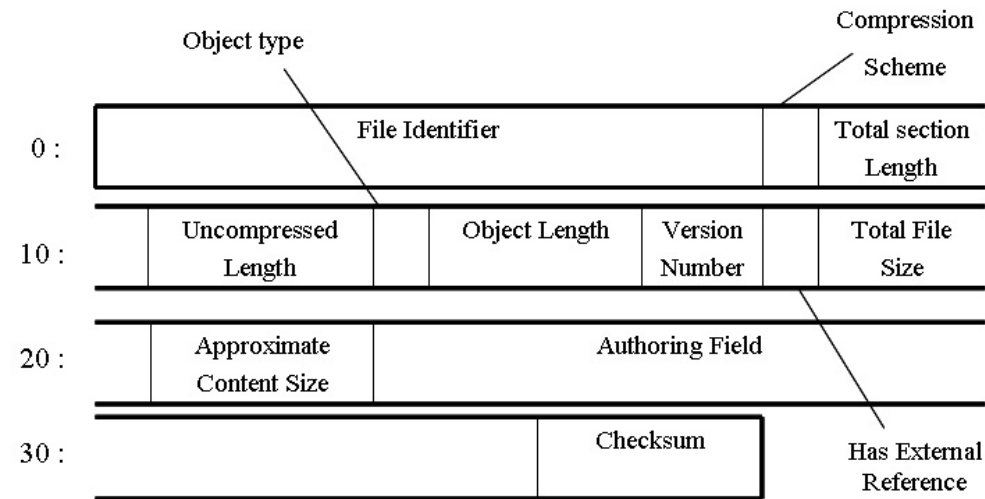


그림 12. m3g 파일 내부구조 예제 : Section 0 (Höfele, C., 2007)

Section 0의 경우에는 전체 파일에 대한 정보들이 담겨 있는데 특별히 참고할 부분은 없으나, Has External Reference(boolean)의 유무를 확인해야한다. m3g파일에 따라서 포함하고 있을 경우도 있고, 아닌 경우도 있기 때문이다. 포함하고 있을 경우에는 Section 0에 바로 다음 섹션으로 존재하며, 그렇지 않을 경우엔 Scene Objects의 정보를 가지고 있는 섹션이 다음에 존재하게 된다.

Scene Object의 정보를 가지고 있는 섹션의 Object는 ObjectType, Length, Data의 순서와 형태로 구성되어지는데, ObjectType은 0부터 255까지 각각의 노드들의 타입을 정의하고, Length는 정의된 노드들의 크기 값을 나타내며, Data는 노드들의 속성 값을 보여준다. 이렇게 정의 되어 있는 노드들에 접근하여 정보를 얻기 위해서는 정해진 구조를 활용한다. 장면이 그려지는 전체적인 구조에 대한 정보를 얻을 필요는 없기 때문에 먼저 필요한 노드들의 값을 정한다. M3G에서 직접적인 객체의 가시화는 그림 7에서 표현했던 Mesh 노드의 정보를 이용하기 때문에 Mesh의 노드들인 VertexBuffer와 IndexBuffer의 자식 노드들의 정보를 참조하는데, 먼저 position, normal, colors, texture coordinate 의 정보를 VertexArray에서 얻어오고, triangle strip을 활용하기 위해서 TriangleStripArray에서 값을 얻어온다. 정리하자면 Mesh 노드의 VertaxArray의 정보와 TriangleStripArray의 정보를 이용해 동일한 객체를 구현할 정보를 제공 받는다.

우선 파일에서 노드들이 어떤 순서로 정의 되어 있는지 모르기 때문에 노드들의 Object Type의 값을 얻어온다. VertaxArray의 값과 TriangleStripArray은 값은 20, 11 이다. 만약 일치하지 않을 경우에는 바로 다음 주소인 Length의 값을 읽어 들여 다음 노드로 이동한다. 일치할 경우엔 두 노드들에 공통적으로 적용되는 상위 노드인 Object3D의 정의 부분(user ID, animationTracks, userParameterCount)의 크기를 고려하고 실제 원하는 값의 주소로 접근하여 정보를 얻어오게 된다. 이렇게 얻어온 값은 그림 12에서 표현했듯이 처리기에서 XML 파서

인 MSXML4.0을 이용하여 기본적인 X3D 문서의 형태를 정의 해주는 전처리 작업을 해주고 정해진 형태에 대응되는 부분으로 대입하는 형태로 X3D로 변환된 문서를 생성한다.

3. 연구 처리 과정 및 결과

각 처리 결과는 선행연구에서 설정했던 그림 7의 기본 3D 객체 자료 구조를 사용하여 확장성을 고려하였고, 파일의 교환 및 변환을 위한 파서를 처리기로 설정하여 공간자료의 값을 변환하여 대응시킬 수 있는 응용프로그램을 구현하였다. 바이너리 형태의 .m3g 파일의 정보를 읽어 들여 처리하는 프로세스의 소스코드는 그림 12와 같으며, 그림 13은 "Geospatial" 컴포넌트를 적용하여 kml과의 노드를 매칭 시키는 작업을 진행 한 결과이다. 그리고 KML의 태그는 표 4와 같이 X3D와 대응시켰다. 또한 그림 14는 3차원 샘플 객체가 구현된 KMZ(Google Earth) - M3G (Mobile3D:JSR184) 파일이 변환된 결과를 보여준다.

표 4. KML-X3D 태그 매칭

| KML | X3D |
|------------|--|
| LookAt | GeoViewpoint |
| Coordinate | GeoCoordinate |
| Linearing | IndexedFaceSet(using a GeoCoordinate) |
| Extrude | Extrusion |

```

IXMLDOMProcessingInstructionPtr pX3DPI;
IXMLDOMElementPtr pX3DRoot;
IXMLDOMElementPtr pX3DScene, pX3DNIInfo, pX3DTrans, pX3DShape, pX3DIF, pX3DCoord;

CoInitialize(NULL);
pX3DDoc.CreateInstance(__uuidof(DOMDocument40));
pX3DPI = pX3DDoc->createProcessingInstruction(L"xml", L"version='1.0#' encoding='UTF-8'");
pX3DDoc->appendChild(pX3DPI);

pX3DRoot = pX3DDoc->createElement(L"X3D");
pX3DRoot->setAttribute(L"profile", (_variant_t)"Immersive");
pX3DRoot->setAttribute(L"version", "3.0");
pX3DRoot->setAttribute(L"xmlns:xsd", "http://www.w3.org/2001/XMLSchema-instance");
pX3DRoot->setAttribute(L"xsd:noNamespaceSchemaLocation", "http://www.web3d.org/specification/X3D-1.2.1.xsd");
pX3DDoc->appendChild(pX3DRoot);

pX3DScene = pX3DDoc->createElement(L"Scene");
pX3DRoot->appendChild(pX3DScene);

pX3DNIInfo = pX3DDoc->createElement(L"NavigationInfo");
pX3DNIInfo->setAttribute(L"type", (_variant_t)"EXAMINE");
pX3DScene->appendChild(pX3DNIInfo);

pX3DTrans = pX3DDoc->createElement(L"Transform");
pX3DTrans->setAttribute(L"translation", (_variant_t) m_m3gFile.tStr);
pX3DScene->appendChild(pX3DTrans);

pX3DShape = pX3DDoc->createElement(L"Shape");
pX3DShape->setAttribute(L"DEF", (_variant_t)"Cube_1");
pX3DTrans->appendChild(pX3DShape);

pX3DIF = pX3DDoc->createElement(L"IndexedFaceSet");
pX3DShape->appendChild(pX3DIF);
pX3DIF->setAttribute(L"coordIndex", (_variant_t) m_m3gFile.cStr);
pX3DIF->setAttribute(L"solid", (_variant_t)"False");
pX3DCoord = pX3DDoc->createElement(L"Coordinate");
pX3DIF->appendChild(pX3DCoord);
pX3DCoord->setAttribute(L"point", (_variant_t) m_m3gFile.pStr);

```

X3D 파일 전 처리부분

좌표값 처리부분

그림 13. m3g 정보 처리 프로세스 소스코드

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN"
"http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile="Immersive" version="3.1" xmlns:xsd="http://www.w3.org/200
<head>
<component name="Geospatial" level="1"/>
</head>
<Scene>
<GeoViewpoint
description="GeoViewpoint_0_00"
geoSystem="GDC","WGS84"
position="37.960285690635715 -122.35234754156428 312128.1944839284"
orientation="-1 0 0 1.57">
<GeoOrigin DEF="ORIGIN"
geoCoords="0.0 0.0 -6378137.0"
geoSystem="GDC","WGS84"/>
</GeoViewpoint>

```

Geospatial Component 적용

태그 매칭

그림 14. kml/kmz과 x3d 매칭 처리 화면

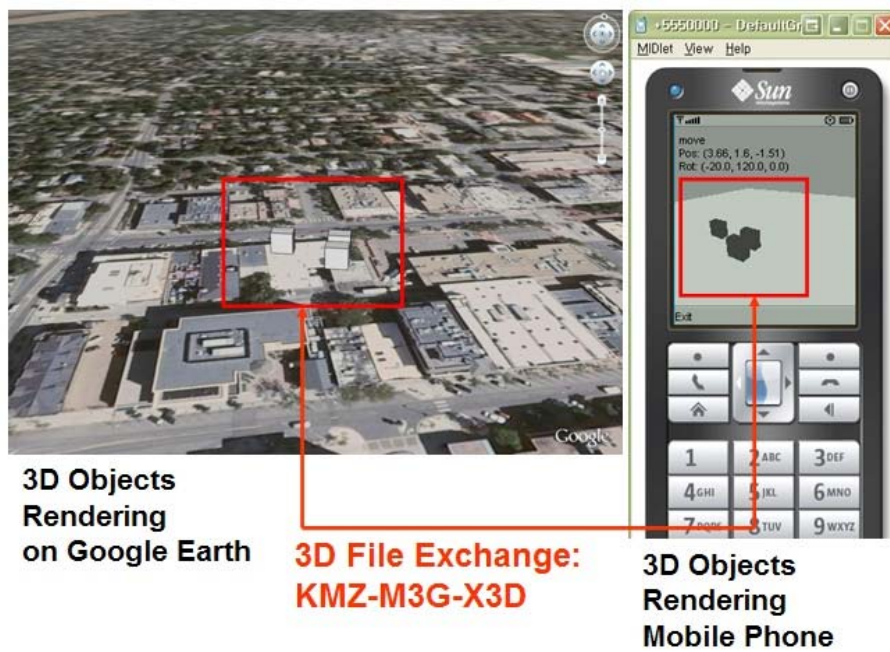


그림 15. 3D file 변환: KMZ (Google Earth)-M3G
(Mobile3D:JSR184)

제 4 장 결론 및 향후 연구

본 연구에서는 모바일과 웹에서의 3차원 공간자료 연계 활용에 관한 연구를 진행 하였다. 모바일과 웹이라는 서로 다른 환경에서 생성된 3차원 공간정보의 가시화를 위해, 각 플랫폼에서의 3차원 그래픽 표준을 이용 하였다. 서로 다른 환경에서 데이터를 공유하기 위해서 추가적으로 각각의 표준에 맞게 호환이 가능하도록 별도의 변환작업을 진행하였고, 표준들 간의 연계활용을 위해 상호호환 및 연계성에 대한 가능성을 검토하였다.

먼저 선행 연구로 GIS 3D 가시화 모델링을 통한 기초 연구를 통해 Java ME 환경에서 M3G API를 시험 구현하였다. 모바일 단말에서의 3차원 공간정보 처리를 위한 모델링에 있어서는 M3G의 Scene Graph 구조에 의한 모델링이 객체 기반의 3차원 모델의 참조와 재사용 문제에서 효과적일 수 있다는 방향으로 접근 할 수 있었다.

이런 선행 연구를 바탕으로 진행한 모바일과 웹에서의 X3D - M3G - KML 간의 3차원 공간자료의 연계 활용에 관한 연구를 활용해, 서로 다른 환경에서의 3차원 그래픽 표준안의 연계 적용은 향후 크로스 플랫폼 환경에서의 3차원 공간정보의 활용에 적용 가능하다.

현재 각 3차원 그래픽 표준은 개별적인 스펙의 업데이트가 진행되어지고 있으며, 웹과 모바일에서도 각각 통합된 표준을 위한 연구가 진행중이기 때문에 향후 구현 시 고려되어야 할 것이다.

본 연구는 웹과 모바일 상에서의 3차원 적용을 위한 표준 API를 활용하여 진행하였기 때문에 향후 공간정보 활용 측면에서의 사용자의 요구를 충족 시켜줄 수 있는 기초 연구로 의의가 있으며, 보다 완전한 연계 활용을 위해서 현재 OGC에서 표준으로 제정된 CityGML과의 연계를 통해 XML 기반의 교환 포맷을 이용하여, 보다 다양한 환경과 기존에 사용되고 있는 프로그램들에서 복잡한 3차원 공간자료들을 이용

할 수 있는 활용도 높은 모바일과 웹의 3D GIS 공간정보 연계 활용도 검토해야 할 것이다.

또한 본 연구는 유비쿼터스 환경에서의 일반인들에게도 손쉽게 3D 공간정보 콘텐츠를 웹과 모바일에서 접할 수 있는 기초 연구로서도 의의가 있다.

참 고 문 헌

- Brutzman, D. and Daly, L., 2007. Extensible3D graphics for Web Authors, pp10-18.
- Kiwon Lee and Seung-Yub Kim, 2006, Development of Mobile 3D Urban Landscape Authoring/Rendering System, Korean Jour. of Remote Sensing, 22(3).
- Josie Wernecke, 2008. The KML HANDBOOK, pp11-12.
- JSR-184 Expert Group, 2005, Mobile 3D Graphics API Technical Specification, JCP.
- Höfele, C. (Ed), 2007. *Mobile3DGraphics: Learning 3D Graphics with the Java MicroEdition*, 305p, 309p.
- Malizia, A., 2006. *Mobile3D Graphics*, Springer, 93p, 96p.
- Morales, C. and D. Nelson, 2007. *Mobile3D Game Development: From Start to Market*, CharlesRiver Media, 71p, 132p.
- Pulli, K., Aarnio.T., Roimela,K., and Vaarala, J., 2005. Designing Graphics Programming Interfaces for Mobile Devices, *IEEEComputerGraphicsandApplications*,25(8).
- Pulli, K., Aarnio.T., Roimela,K., and Vaarala, J., 2008. *Mobile3D Graphics with OpenGL ES and M3G*, Elsevier, pp.120-129, 314p.

Thomas H. Kolbe, 2007, CityGML - 3D Geospatial and Semantic Modelling of Urban Structures, OGC.

Tim O'Reilly, "What is Web 2.0 Design Pattern and Business Model for the Next Generation of Software".

Tim Wilson.(ED), 2008, OGC@KML OGC 07-147r2.

Khronos Group, <http://www.khronos.org>.

web3D consortium, <http://www.web3d.org>.

ABSTRACT

Interoperable application of 3D Geo-spatial Features on Web3D and Mobile3D

Dong, Woo-Cheol

Major in Information System Engineering

Dept. of Information System Engineering

Graduate School, Hansung University

It is more greater need in Web and Mobile for Geo-spatial information utilization. Nowadays, however, it is main tendency for Geo-spatial processing to visualize 2D feature and to deal with image contents. Mobile 3D Graphics field for visualizing Geo-spatial transfers, due to development of Mobile device and telecommunication technology, the service that these days are offered in web to demand via mobile. Mobile 3D Graphics field is most likely to advance with 3D Graphics field in GIS field from now on.

Therefore, in this study, using the standard for 3D visualization related with standards, it progresses about Geo-spatial process and interoperable application on web3D and mobile3D.

Keywords : GIS, 3D GIS, M3G, Web3D, Mobile3D