

모바일 플랫폼을 위한 장면그래프 라이브러리 개발

김준호[†], 서진석^{**}, 박창훈^{***}, 황재인^{****}, 고희동^{*****}

요 약

본 논문에서는 모바일 플랫폼을 위한 범용적인 그래픽스 미들웨어로써, 모바일 장면그래프 라이브러리를 '모바일 OpenSceneGraph (모바일 OSG)'를 제안하고 그 개발 사례를 소개한다. 제안하는 모바일 그래픽스 미들웨어는 데스크탑 환경에서 그래픽스 미들웨어로 널리 쓰이고 있는 OpenSceneGraph를 모바일 환경에 적합하도록 재설계하였으며, 이를 통해 범용성, 확장성, 모바일 플랫폼 UI 지원, 호환성 등의 특성을 확보하였다. 복잡한 가상환경의 실시간 렌더링, 터치 UI 기반의 카메라 조작, 노드 전환을 통한 애니메이션 등과 같은 응용프로그램을 통해 제안하는 모바일 장면 그래프 라이브러리의 효용성을 보이도록 한다.

Development of Scene Graph Library for Mobile Platforms

Junho Kim[†], Jinseok Seo^{**}, Changhoon Park^{***}, Jane Hwang^{****}, Heedong Ko^{*****}

ABSTRACT

In this paper, we introduce a novel scene graph library for mobile platforms, called the 'Mobile OpenSceneGraph (Mobile OSG)', as a mobile graphics middleware. Our mobile scene graph library supports several nice properties, including platform-independence, extensibility, touch-based UI supports, and compatibility, by carefully adapting the OpenSceneGraph library, one of the most widely used graphics middlewares for desktop platforms, to mobile platforms. We validate the usefulness of the proposed library for mobiles with the several experimental results including real-time rendering, camera manipulations with a touch-based UI, and animations with switching geometric nodes.

Key words: Mobile Platform(모바일 플랫폼), Scene Graph(장면그래프), Mobile Graphics Middleware (모바일 그래픽스 미들웨어), OpenSceneGraph, OpenGL ES

1. 서 론

최근 휴대폰, PDA 등을 비롯한 모바일 하드웨어와 관련된 기술의 발달로 인해, 낮은 전력소비량을 가지면서도 불과 1~2년 전의 데스크탑 컴퓨터와 유사한 계산처리 능력을 가지고 있는 고성능 모바일

기기가 널리 쓰이고 있다. 또한 대용량 데이터 전송이 가능한 3세대 이동통신망이 보편화됨에 따라 인터넷 서핑, 정보 서비스, 위치기반 서비스, 전자상거래 등과 같은 대용량 멀티미디어 데이터 기반의 모바일 콘텐츠 서비스 시장이 점차 활성화 되고 있어, 효과적인 모바일 멀티미디어 콘텐츠 개발 및 활용 기술

* 교신저자(Corresponding Author): 서진석, 주소: 부산광역시 부산진구 가야3동 동의대학교 산학협력관 502호, 전화: (051)890-2712, FAX: (051)890-2639, E-mail: jsseo@deu.ac.kr

접수일: 2009년 12월 4일, 수정일: 2010년 1월 27일
완료일: 2010년 3월 5일

[†] 정회원, 국민대학교 컴퓨터공학부 전임강사
(E-mail: kim.junho@gmail.com)

^{**} 정회원, 동의대학교 게임공학과 조교수

^{****} 호서대학교 게임공학과 교수
(E-mail: chpark@hoseo.ac.kr)

^{****} 한국과학기술연구원 선임연구원
(E-mail: hjji@imrc.kist.re.kr)

^{*****} 정회원, 한국과학기술연구원 영상미디어센터장
(E-mail: ko@imrc.kist.re.kr)

* 본 연구는 한국과학기술연구원의 기관고유과제인 "Tangible Web 기술개발"과제의 지원으로 수행되었음

에 많은 관심이 쏟아지고 있다.

근래 들어 대중화되기 시작한 스마트폰의 경우, 모바일 그래픽스 가속 칩(mobile graphics acceleration chip)이 기본으로 장착되어 있어 이미지·비디오·삼차원 형상 등과 같은 대용량 멀티미디어 데이터를 하드웨어 가속기능을 통해 효과적으로 처리할 수 있다. 모바일 플랫폼에 탑재되어 있는 그래픽스 가속칩은 데스크탑용 삼차원 그래픽스 API인 OpenGL을 임베디드 시스템 환경에 맞춰 재정의한 OpenGL ES (OpenGL for Embedded System) [1]를 지원한다. OpenGL ES는 모바일 OS와 무관하게 삼차원 가속 기능을 사용할 수 있도록 설계되어 있으며, 블록 단위 데이터 전송을 통한 렌더링 방식을 지원함으로써 제한된 하드웨어 리소스 환경을 가지는 임베디드 및 모바일 환경에서 빠르고 효과적인 렌더링 성능을 제공하도록 설계되어 있다. 하지만, OpenGL ES는 삼차원 그래픽스 응용프로그램을 위한 최하위 수준의 API이기 때문에, 이를 통해 복잡하고 다양한 멀티미디어 응용프로그램을 모바일 플랫폼용으로 개발할 경우 개발 기간과 비용이 많이 들며 응용프로그램의 유지/보수에 많은 어려움이 따르게 되는 단점을 가지고 있다.

일반적으로 복잡한 멀티미디어 응용프로그램은 장면그래프 라이브러리(scene graph library) 기반의 그래픽스 미들웨어(graphics middleware)를 활용하여 개발된다. 장면그래프 라이브러리는 이차원 이미지·삼차원 형상의 입출력, 가상카메라 처리, 객체의 다중해상 렌더링, 파티클 시스템의 지원 등과 같이 멀티미디어 응용프로그램의 구현에 있어 빈번히 사용되는 자료구조 및 알고리즘들을 체계적으로 제공한다. 이와 같은 장면그래프 기반의 미들웨어를 사용할 경우, 개발자는 하위 수준의 그래픽스 API를 직접 사용할 필요 없이 고수준의 객체 지향적 방법론을 통해 다양한 멀티미디어 응용프로그램을 손쉽게 개발할 수 있다는 장점이 있다. 데스크탑 환경의 경우, OpenSceneGraph[2], OpenSG[8], Performer, OpenInventor 등과 같은 장면그래프 라이브러리를 활용하면 가상현실, 증강현실, 컴퓨터그래픽스, 정보가시화 등 다양한 분야의 응용프로그램을 쉽고 빠르게 개발할 수 있다. 이에 반해 모바일 환경의 경우, 데스크탑 환경에 버금가는 복잡하고 다양한 응용프로그램에 대한 개발수요가 최근 꾸준히 증가하고 있으나, 범용적으

로 쓰일 수 있는 모바일 그래픽스 미들웨어의 부재로 인해 고수준의 모바일 멀티미디어 응용프로그램을 개발하기 쉽지 않은 실정이다.

본 논문에서는 모바일 플랫폼을 위한 장면그래프 기반의 범용적인 그래픽스 미들웨어인 ‘모바일 OpenSceneGraph(모바일 OSG)’를 제안하고 그 개발 사례 및 응용 사례를 소개하고자 한다. 빠르게 변하는 모바일 컴퓨팅 환경에 맞춰, 모바일 장면그래프 기반의 그래픽스 미들웨어는 다음과 같은 특성을 반영해야 한다.

- 범용성: 모바일 플랫폼 시장은 현재 심비안 OS, 아이폰 OS, 윈도우즈 모바일, 임베디드 리눅스 등 다양한 운영체제들이 그 시장을 나눠 가지고 있다. 따라서 모바일 플랫폼용 장면 그래프 라이브러리는 특정 플랫폼과 무관한 (platform-independent) 범용성을 가지고 있어 쉽게 여러 모바일 플랫폼으로 이식이 가능해야 한다.
- 확장성: 모바일 하드웨어가 급속도로 발전함에 따라, 모바일 플랫폼에 새로운 기능의 추가가 빈번히 일어나고 있다. 모바일 플랫폼용 장면 그래프 라이브러리는 새로운 기능을 유연하게 확장할 수 있는 플러그인 형식의 모듈화된 기능을 제공해야 할 것이다.
- 호환성: 기존에 개발되었던 여러 응용프로그램들을 최대한 재사용할 수 있도록 호환성을 확보할 수 있도록 한다.
- 모바일 플랫폼 UI 고려: 스마트폰의 경우, 플터치 스크린 방식을 통해 효과적인 UI 조작용을 지원하는 경우가 일반적이다. 모바일 플랫폼용 장면 그래프 라이브러리는 이와 같은 터치 방식의 새로운 UI 방식을 지원해야 할 것이다.

제안하는 모바일 OSG는 데스크탑용 공개 소프트웨어인 OpenSceneGraph(이하 OSG)를 모바일 환경에 적합하도록 재설계함으로써, 범용성, 확장성, 모바일 플랫폼 UI의 지원, 호환성을 확보하였다. 복잡한 가상환경의 실시간 렌더링, 터치 UI 기반의 카메라 조작, 노드 전환을 통한 애니메이션 등과 같은 예시 응용프로그램을 통해 본 논문에서 제안하는 모바일 장면그래프 라이브러리의 효용성을 보이도록 한다. 마지막으로, 멀티터치 및 기울기 센서 등을 지원하는 스마트폰 플랫폼인 아이폰(iPhone)에서 모바일 OSG의 응용사례를 보이도록 한다.

2. 관련 연구

2.1 장면그래프

장면그래프(scene graph)는 삼차원 공간상에 존재하는 다수의 가상물체 및 멀티미디어 객체를 효과적으로 관리하기 위해 사용되는 계층적 트리 형식의 자료구조이다[2]. 일반적으로 장면그래프 트리의 최상단은 주어진 삼차원 장면을 대표하는 루트 노드(root node)가 위치하며, 최하단에는 삼차원 장면 상에 존재하는 형상에 관한 실제 데이터를 가지고 있는 다수의 리프 노드(leaf node)들이 위치한다. 루트 노드와 리프 노드 사이에는 일반적으로 그룹노드(group node)들이 위치하는데, 이들은 삼차원 형상의 변환 및 렌더링 상태 관리를 통해 복잡한 삼차원 장면을 보다 효과적으로 렌더링할 수 있게 하는 기능을 담당한다.

이러한 장면그래프 자료구조에 삼차원 응용프로그램에서 자주 활용되는 기능을 덧붙여 복잡한 삼차원 응용프로그램을 손쉽게 개발할 수 있도록 설계된 라이브러리가 장면그래프 미들웨어(scene graph middleware)이다. 그림 1은 장면그래프 미들웨어를 통한 삼차원 응용프로그램 개발의 구성도를 보여준다. 그림 1에서 볼 수 있듯이, 장면그래프 미들웨어는 저수준 렌더링 API (low-level rendering API)를 기반을 설계되어 하드웨어 가속을 통한 실시간 렌더링이 가능하며, 삼차원 응용프로그램은 장면그래프 미들웨어 기반으로 설계되어 다중해상 렌더링, 렌더링 최적화, 파일 입출력, 파티클 렌더링, 섀도우 기능, 뷰어(viewer) 등과 같은 고급 기능들을 저수준 렌더링 API에 영향을 받지 않고 설계할 수 있게 된다.

일반적으로 많이 사용되는 장면그래프 라이브러리는 대표적으로 Performer, OpenInventor, Open SG [8], OpenSceneGraph(OSG) [2] 등이 있다. 장면그래

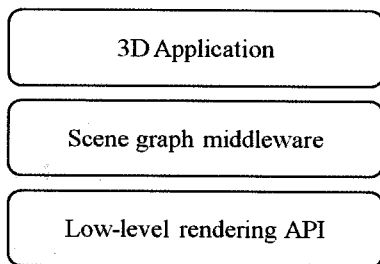


그림 1. 장면 그래프 미들웨어를 통한 삼차원 응용프로그램 개발 구성도

프 라이브러리는 게임, 가상현실, 증강현실 시스템 구현에 주로 사용되는데 이 가운데 OSG는 현재 전 세계적으로 가장 널리 쓰이고 있는 공개 소프트웨어 기반의 미들웨어이다. OSG는 산업계 표준인 OpenGL을 저수준 그래픽스 API로 쓰고 있으며 ANSI C++로 작성되어 있어 높은 이식성을 가지고 있다. 그 외에도 OpenInventor 역시 널리 쓰이는 장면 그래프 중의 하나이며, Coin3D[7]로 이전이 되어서 최근까지 지속적으로 사용되고 있다.

최근에는 증강현실 API와 장면 그래프 라이브러리를 결합하여 증강현실 미들웨어로 만드는 움직임이 가속화 되고 있다. 그 중 가장 대표적인 것이 OSGART로 OpenSceneGraph와 ARToolKit을 결합한 미들웨어이다[4]. 그리고 일반에 공개되고 있지는 않으나 Graz 대학의 StudierStube 및 StudierStube ES라는 증강현실 미들웨어에서는 장면그래프로 Coin3D를 변형한 형태를 사용하고 있다[5]. 국내에서는 KIST 영상미디어연구센터에서 OSG를 이용한 혼합현실 미들웨어인 VARU 프레임워크를 개발하였고[6], 광주과학기술원에서는 CAMAR 2.0이라는 유비쿼터스 가상환경에서의 모바일 증강현실이 나아가야 할 방향을 제시하고 있다.

이와 같이 장면 그래프를 이용한 데스크탑 기반의 증강현실 미들웨어의 개발이 세계 각국에서 이루어지고 있는 실정이지만, 모바일용 증강현실 미들웨어는 현재까지 거의 없는 실정이다. 이는 근본적으로 범용적인 장면그래프가 모바일용으로 아직까지 공개되고 있지 않기 때문이다. Graz 대학의 StudierStube ES가 현재까지 발표된 유일한 모바일용의 증강현실 미들웨어로 알려져 있지만, StudierStube ES는 Coin3D의 일부 기능만을 사용하고 있기 때문에 제한된 미들웨어로서의 역할을 하고 있으며, 일반에 공개된 형태로 배포되고 있지 않아 널리 쓰이지 않고 있다. 데스크탑 환경에서는 OpenSceneGraph와 ARToolKit의 범용성 덕분에 OSGART가 널리 사용되고 있으나, 모바일용 OpenSceneGraph가 개발되지 않아서 모바일 OSGART의 공개가 늦어지고 있는 상황이다. 본 논문에서는 제안하는 '모바일 OSG'는 데스크탑 환경에서 널리 쓰이고 있는 OpenSceneGraph의 범용성을 그대로 가지고 왔기 때문에, 증강현실 기능의 확장을 통한 장면그래프 기반 증강현실 미들웨어의 개발이 추후 손쉽게 이루어 질 수 있다.

2.2 모바일 그래픽스

OpenGL ES는 모바일폰, PDA, 휴대용 게임기 등과 같은 제한된 리소스 환경을 가지는 임베디드 기기에서 원활히 동작할 수 있도록 OpenGL을 재설계한 개방형 포맷의 저수준 삼차원 그래픽스 API이다. 최근 삼차원 가속 기능을 가지는 모바일 기기는 모두 OpenGL ES를 지원하고 있으며, OpenGL ES의 API 프로파일(API profiles)은 현재 비영리성 컨소시엄인 Khronos 그룹의 주도하에 설계되고 있다[1].

OpenGL ES 함수는 OpenGL 함수에 기반을 두어 설계하되, 임베디드 환경에 적합하지 않은 OpenGL 기능들은 모두 삭제하고, 각각의 함수들을 저전력 모바일 3D 가속칩에 알맞게 재설계하였다. 뿐만 아니라, 플랫폼 OS에 따라 WGL/AGL/GLX 등과 같이 서로 다른 플랫폼 인터페이스 레이어(platform interface layer)를 사용해야 했던 OpenGL과는 달리, EGL이라 불리는 공용 모바일 플랫폼 인터페이스 레이어(common mobile interface layer)의 표준을 됴으로써, 다양한 모바일 하드웨어 상에서 보다 쉽게 OpenGL ES 기반의 삼차원 응용프로그램을 구현할 수 있도록 설계되어 있다. 현재 OpenGL ES는 2.0 버전까지 발표되었으나, 현존하는 대부분의 모바일 기기들은 1.1 버전을 지원하고 있다. 본 논문에서는 최근 스마트폰들이 널리 지원하는 OpenGL ES 1.1을 고려대상으로 하여 모바일용 장면그래프를 개발하였다.

2.3 OpenSceneGraph(OSG)

OSG는 비주얼 시뮬레이션, 게임, 가상현실, 과학적 가시화/모델링 등 여러 분야에서 널리 활용되고 있는 오픈소스 그래픽스 미들웨어이다[2]. OSG는 모든 소스 코드가 ISO 표준 C++와 OpenGL로만 구성되어 있어 이식성이 매우 뛰어나다는 것이 가장 큰 장점 중에 하나이며, 실제로 Microsoft Windows, Apple Mac OS X, GNU/Linux, IRIX, Solaris, HP-UX, AIX, FreeBSD 등의 운영체제에서 구동될 수 있다.

그림 2는 OSG의 구조를 보여주고 있다. OSG의 핵심인 Core 부분은 장면그래프와 이를 구성하는 노드가 구현되어 있으며 렌더링 기능과 더불어 벡터/행렬/형상/렌더링 상태 등에 관한 것들도 포함한다. osgUtil은 장면그래프를 조작하는데 쓰이는 클래스와 Delauny 삼각화, 텍스처 좌표 형상 등 삼차원 형상에 관련된 기능을 지원하는 클래스로 구성되어 있

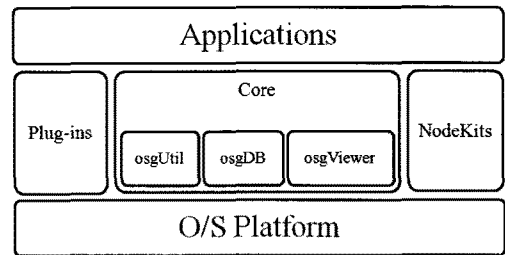


그림 2. OpenSceneGraph(OSG)의 구조도

다. osgDB는 삼차원 물체들에 대한 데이터베이스를 만들고 이를 렌더링에 사용할 수 있는 클래스와 함수들로 구성되어 있다. osgViewer는 다양한 운영체제의 윈도우 시스템에서 OSG를 이용한 렌더링 화면을 표시하기 위해 제공되는 범용적인 뷰어 클래스이다.

핵심적인 기능을 제공하는 Core 이외에도 OSG에 확장성을 부여하는 NodeKits와 Plug-ins 등을 통해 다방면의 응용프로그램의 작성이 가능하며, 더불어 다양한 포맷으로 이루어진 리소스의 사용이 가능하다. NodeKits는 장면 그래프의 노드를 확장하여 범프 매핑, 카툰셰이딩, 파티클 렌더링, 지형 렌더링과 같은 특수 효과를 지원하고, Plug-ins는 다양한 포맷의 2D이미지와 3D 모델의 입출력을 지원한다.

3. 모바일 장면그래프 라이브러리

3.1 개요

본 연구에서는 전 세계적으로 가장 널리 쓰이는 데스크탑용 장면그래프 미들웨어인 OSG를 모바일 플랫폼에 맞게 재설계함으로써, 범용성, 확장성, 호환성, 모바일 플랫폼 UI 지원 등의 이점을 가지는 모바일 장면그래프 미들웨어를 개발하였다.

- 범용성: 장면 그래프 라이브러리 중 삼차원 렌더링 부분은 OpenGL ES를 통해 구현함으로써 모바일 플랫폼에 무관한 범용성을 확보하였다.
- 확장성: 플러그인 방식의 모듈화된 기능을 제공함으로써, 새로운 기능을 쉽게 확장할 수 있으며, 필요한 기능만 컴파일(혹은 런타임) 단계에서 편리하게 선택할 수 있어 보다 가벼운 모바일용 응용프로그램 개발이 가능하다.
- 호환성: 데스크탑용 OSG를 활용하여 개발된 기존의 응용프로그램을 별다른 수정 없이 모바일 용

용프로그램으로 재사용할 수 있도록 소스코드 단계에서의 호환성(source-code compatibility)을 최대한 확보할 수 있도록 하였다.

- 모바일 플랫폼 UI 고려: 삼차원 물체에 대한 회전, 확대/축소, 이동 등과 같은 조작을 플터치스크린 방식의 UI를 통해 지원할 수 있도록 설계하였다.

모바일 장면그래프 미들웨어 설계에 있어 가장 핵심적인 사항은 저수준 렌더링 API가 OpenGL ES로 변경되어야 한다는 점이다. 그림 1에서 나타난 바와 같이, 가장 밑단의 저수준 렌더링 API가 수정되면 미들웨어 혹은 응용프로그램에 그 영향이 미치게 된다. 특히, OpenGL ES는 데스크탑용 저수준 렌더링 API에 비하여 제한된 기능만을 지원하므로 이에 기반한 모바일 장면그래프 미들웨어 역시 그 기능이 제한될 수밖에 없다. 자연스럽게 가장 윗단에 위치한 삼차원 응용프로그램이 그 영향을 받게 되는데, 이 때 기존에 개발된 응용프로그램의 소스코드를 최대한 수정하지 않더라도 모바일 환경에 효과적으로 재사용할 수 있도록 하는 것이 매우 중요한 문제로 대두된다. 또한, 대부분의 기존 응용프로그램들은 OSG에서 제공하는 범용적 뷰어인 osgViewer를 기반으로 설계되어 있다. osgViewer의 핵심 기능 중 하나는 플랫폼 인터페이스 레이어를 여러 개의 메소드를 통해 플랫폼에 무관하게 추상화해 둔 것인데, 모바일 환경으로 확장할 경우 EGL 기반의 모바일 플랫폼 인터페이스 레이어를 통해 모바일 하드웨어에 무관하게 동작하도록 osgViewer를 재설계하는 것이 또 하나의 문제로 대두된다.

3.2 모바일 환경을 위한 설계

모바일 OSG 설계에 있어 고려해야 할 문제는 크게 다음의 두 가지로 요약된다.

- OpenGL ES를 사용함으로써 생기는 렌더링 방식의 변화
- 모바일 하드웨어를 위한 osgViewer의 재구성

3.2.1 렌더링 방식의 변화

삼차원 응용프로그램의 렌더링 방식은 화면의 갱신을 위해 필요한 형상 데이터를 어디에서 읽어 오느냐에 따라, 즉시모드(immediate mode)와 유지모드(retained mode)로 구분된다. 즉시모드는 메인 메모

리로부터 형상 데이터를 추출(fetch)해서 그래픽스 하드웨어를 통해 렌더링 하는 방법이다. 유지모드는 형상 데이터를 메인 메모리로부터 그래픽스 하드웨어의 비디오메모리로 한번 전송해 놓은 뒤, 화면의 갱신이 필요할 때마다 비디오 메모리에 이미 전송되어 있는 데이터를 이용하여 렌더링 하는 방법이다.

OpenGL ES에서는 좁은 로컬 버스 대역폭을 가지는 모바일 기기의 특성을 고려하여 즉시모드는 지원하지 않고 정점배열(vertex array) 혹은 정점버퍼객체(vertex buffer object) 기반의 유지모드만을 지원한다. 이와 같은 유지모드를 통한 렌더링 방법은 물체의 형상데이터를 블록(block) 단위로 전송해야 하기 때문에 형상데이터의 자료구조는 반드시 배열(array)이 되어야 한다.

OSG는 기본적으로 즉시모드 기반의 렌더링을 사용하고 있으나, OpenGL ES를 저수준의 렌더링 API로 사용해야 하는 모바일 OSG의 경우, 모든 렌더링 코드가 배열 자료구조를 이용한 유지모드로 재설계될 필요가 있다. 데스크탑용 OSG의 렌더링 코드를 모바일 OSG에서는 그림 3(a)와 같이 유지모드 기반으로 변형하였다. 그림 3(a)은 데스크탑용 OSG의 함수 중 즉시모드를 통한 렌더링 코드의 예를 보여주고 있으며, 그림 3(b)는 (a)에 대응하는 렌더링 코드를 OpenGL ES에서 지원하는 유지모드를 이용도록 변형한 결과이다.

3.2.2 모바일 환경을 위한 뷰어 설계

삼차원 응용프로그램은 단일 시점(single view)이나 다중 시점(multiple views)을 통해 주어진 장면을 관찰하는 경우가 일반적이다. 이를 위해, OSG는 하위 플랫폼과 무관하게 단일 시점 및 다중 시점을 지원하는 응용프로그램을 쉽게 구현할 수 있도록 osgViewer 클래스를 제공하고 있다. osgViewer 클래스는 삼차원 장면에 있는 형상 데이터에 접근하기 위해 사용되는 장면그래프 객체와 삼차원 장면을 관찰할 때 사용되는 카메라 객체를 멤버변수로 가지고 있으며, 저수준 렌더링 API의 호출 결과가 화면에 반영되도록 WGL/AGL/GLX 등과 같은 플랫폼 인터페이스 레이어를 추상화시켜 놓은 메소드를 가지고 있다. 그러나 현재 OSG의 osgViewer는 윈도우즈 OS, 맥 OS, 유닉스 계열 등의 데스크탑 환경을 지원하며, 모바일 플랫폼에 대한 지원은 없는 상태이다.

<pre>glBegin(draw_mode); ... glTexCoord2f(...); glNormalf(...); glVertex3f(...); glTexCoord2f(...); glNormalf(...); glVertex3f(...); ... glEnd();</pre>	<pre>glEnableClientState(GL_VERTEX_ARRAY); glEnableClientState(GL_TEXTURE_COORD_ARRAY); glEnableClientState(GL_NORMAL_ARRAY); ... glVertexPointer(3, GL_FLOAT, 0, v.ptr()); glTexCoordPointer(2, GL_FLOAT, 0, v.ptr()); glNormalPointer(3, GL_FLOAT, 0, v.ptr()); ... glDisableClientState(GL_VERTEX_ARRAY); glEnableClientState(GL_TEXTURE_COORD_ARRAY); glEnableClientState(GL_NORMAL_ARRAY);</pre>
(a)	(b)

그림 3. 렌더링 코드의 재설계의 예. (a) 즉시모드를 통한 렌더링 코드, (b) 유지모드를 통한 렌더링 코드

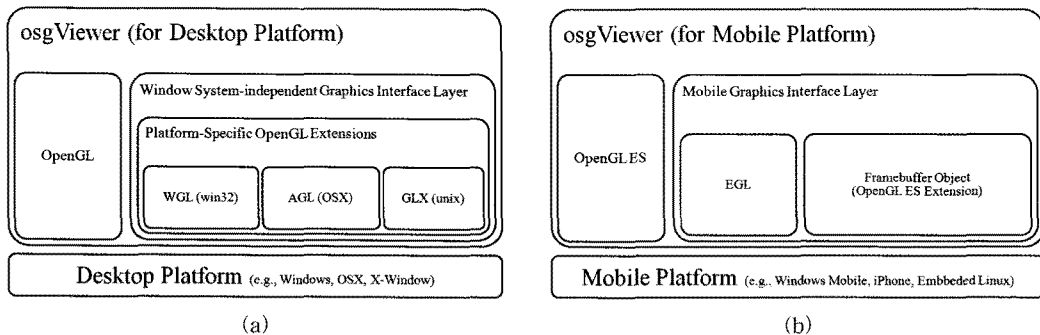


그림 4. osgViewer의 구성도. (a) WGL/AGL/GLX 등의 플랫폼 인터페이스 레이어를 기반으로 한 데스크탑 기반의 osgViewer 구성도, (b) EGL 및 프레임버퍼 객체를 기반으로 한 모바일 플랫폼을 위한 osgViewer 구성도

본 논문에서는 모바일 장면그래프 미들웨어를 위해, 하단의 모바일 OS와 무관하게 동작할 수 있도록 osgViewer를 확장하였다. 이를 위해, 모바일용 윈도우 인터페이스 레이어에 대응하는 EGL을 사용하는 방법과 프레임버퍼 객체를 사용하는 방법을 제안하고자 한다.

EGL은 데스크탑 플랫폼에서 윈도우 인터페이스 레이어로서 서로 같은 기능을 하고 있지만 OS별로 서로 다른 네이티브 API 프로파일로 작성되어 쓰이고 있는 WGL/AGL/GLX 함수들을 모바일 플랫폼 환경에 맞게 API 프로파일을 표준화한 것이다. 따라서 각 EGL 함수는 WGL(혹은 AGL/GLX) 함수와 각각 대응된다[1]. EGL을 기반으로 osgViewer를 모바일 플랫폼으로 확장할 경우, WGL 함수를 추상화시킨

osgViewer의 메소드를 찾고, 이에 대응하는 EGL 함수로 대체함으로써 손쉽게 모바일 환경을 위한 osgViewer를 재설계할 수 있다.

그림 5(a)은 osgViewer와 관련된 클래스의 메소드인 makeCurrentImplementation()에 대해 EGL 함수를 이용하여 모바일 플랫폼을 위한 윈도우 인터페이스 레이어를 작성한 예이다.

EGL은 컨텍스트(context) 기반의 WGL/AGL/GLX와 같은 플랫폼 인터페이스 레이어를 모바일 환경으로 변형한 것이어서, 이들이 가지는 단점을 그대로 내포하고 있다. 다중 시점이 필요한 응용프로그램은 각 시점을 렌더링하기 위해 만들어진 윈도우가 여러 개 있게 된다. 각 윈도우는 하나의 컨텍스트를 가지게 되는데, 한 응용프로그램에서는 임의의 순간 단 하나의

<pre> bool GraphicsWindowWM:: makeCurrentImplementation() { ... eglMakeCurrent(_display, _surface, _surface, _ctx); ... } </pre>	<pre> bool GraphicsWindowiPhone:: makeCurrentImplementation() { ... glBindFramebufferOES(GL_FRAMEBUFFER_OES, viewFramebuffer); ... } </pre>
(a)	(b)

그림 5. 모바일 플랫폼을 위한 osgViewer 재설계의 예. (a) EGL에 기반한 방법, (b) 프레임버퍼 객체에 기반한 방법

컨텍스트만 활성화 될 수 있다. 따라서 다중 시점을 가지는 응용프로그램은 화면이 갱신될 때마다, 각 윈도우가 이전 컨텍스트를 비활성화 시키고 자신의 컨텍스트를 활성화시키는 컨텍스트 교체(context switch) 과정이 지속적으로 일어난다. 이러한 컨텍스트 교체 과정은 시간 소모가 큰 과정에 속해, 특히 제한된 모바일 환경에서 컨텍스트 교체 기법을 통한 다중 시점 뷰어를 구현하는 것은 좋은 선택이 아니다.

프레임버퍼 객체는 이 같은 컨텍스트 기반 윈도우 인터페이스 레이어의 단점을 극복하기 위해 제안된 확장기능(extension)으로써, OpenGL 1.1 및 OpenGL ES 1.1에서 이를 지원하고 있다[3]. 프레임버퍼 객체는 마치 프레임버퍼처럼 동작하는 비디오 메모리 버퍼로써, 하나의 프레임버퍼 객체에는 컬러버퍼(color buffer), 깊이버퍼(depth buffer), 스텐실버퍼(stencil buffer) 등 다수의 버퍼를 종류에 상관없이 유기적으로 관리할 수 있다. 특히, 프레임버퍼 객체가 여러 컬러버퍼를 관리하더라도 컨텍스트 교체가 일어나지 않기 때문에, 다중 시점 응용프로그램의 경우 EGL에 기반한 방법보다 빠른 렌더링 속도를 보장할 수 있다. 그림 5(b)은 osgViewer와 관련된 클래스의 메소드인 makeCurrentImplementation ()을 프레임버퍼 객체 기반으로 구현한 예이다.

3.2.3 기능의 제거

모바일 OSG는 임베디드 시스템에 최적화된 저수준 그래픽스 API인 OpenGL ES를 기반으로 하고 있기 때문에, OpenGL ES로 인해 모바일 OSG 설계 시

다음과 같은 추가적인 제약 사항이 고려되었다.

- 기본 데이터형의 제약: 정수형 자료구조 중 short, byte 등의 자료형만을 지원하여 타 정수형 자료구조는 모두 short 또는 byte로 변경하였다. 실수형 자료구조는 float 형만을 지원하기 때문에 double 형을 사용하는 함수들은 모두 float 형을 사용하도록 변형하였다.
- 프리미티브로 삼각형만을 지원: OpenGL ES에서는 삼각형을 제외한 기타 프리미티브는 모두 삭제되었기 때문에, 기타 프리미티브는 삼각화하여 삼각형들의 집합으로 렌더링하였다.
- 이차원 텍스처만을 지원: 일차원 및 삼차원 텍스처는 모바일 하드웨어에서 지원하지 않기 때문에 삭제하였다.
- 고급 래스터라이저 기술 삭제: 스티플링 등을 비롯한 OpenGL 기반의 고급 래스터라이저 기술은 지원하지 않기 때문에 모바일 OSG에서는 해당 기능을 삭제하였다.

3.2.4 터치 방식의 UI 지원

최근 스마트폰은 터치 방식의 UI를 지원함으로써 편리하고 직관적인 조작을 지원해 주고 있다. 본 논문에서는 osgViewer에 터치방식의 UI를 지원하는 기능을 새로이 정의함으로써 터치를 통한 모델의 회전, 확대/축소, 이동 등이 가능하도록 하였다. 그림 6은 osgViewer와 관련된 Viewer 클래스 상에서 이와 같은 기능을 터치 기반의 UI 입력으로부터 가능하도록 하는 코드 부분을 간략화시켜 보여주고 있다.

<pre>bool GraphicsWindowWM:: makeCurrentImplementation() { ... eglMakeCurrent(_display, _surface, _surface, _ctx); ... }</pre>	<pre>bool GraphicsWindowiPhone:: makeCurrentImplementation() { ... glBindFramebufferOES(GL_FRAMEBUFFER_OES, viewFramebuffer); ... }</pre>
(a)	(b)

그림 6. 모바일 플랫폼을 위한 osgViewer의 재설계의 예. (a) EGL에 기반한 방법, (b) 프레임버퍼 객체에 기반한 방법

5. 구현 및 실험

본 논문에서 제안하는 모바일 OSG는 OpenGL ES 및 표준 C++를 지원하는 모든 모바일 플랫폼에 적용이 가능하다. 본 장에서는 대표적인 스마트폰인 아이폰을 대상으로 데스크탑용 OSG를 기반으로 한 기존의 응용프로그램이 소스코드의 수정 없이 구동되는지 확인함으로써, 제안하는 모바일 OSG의 효용성을 보여주하고자 한다. 뿐만 아니라, 아이폰이 제공하는 터치기반의 UI와 기울기 센서 등을 활용한 응용프로그램의 예를 보임으로써 본 논문이 제안하는 모바일 OSG의 응용 사례를 소개하고자 한다.

5.1 기본적인 기능 테스트

그림 7은 모바일 OSG의 osgViewer에 대해 간단

히 몇 줄의 코드를 추가하여 모바일 OSG의 기본적인 기능이 제대로 수행되는지를 검증한 예이다. 그림 7(a)는 형상노드(geometry node)를 활용하여 유지모드 기반의 프리미티브(primitive) 처리가 모바일 OSG에서 지원되는지 검증한 예이다. 그림 7(b)는 모바일 OSG에서 텍스처 및 조명 처리가 제대로 지원되는지 검증한 예제이며, 그림 7(c)는 스위치 노드(switch node)를 활용해 애니메이션 기능이 제대로 처리되는지 검증한 예제이다.

5.2 복잡한 예제를 통한 테스트

그림 8은 모바일 OSG를 이용하여 여러 개의 공들이 서로 충돌하고 반응하는 물리 시뮬레이션의 결과를 가시화한 것이다. 본 예제에서는 아이폰의 기울기 센서를 이용하여 각각의 공에 중력을 가하고 그

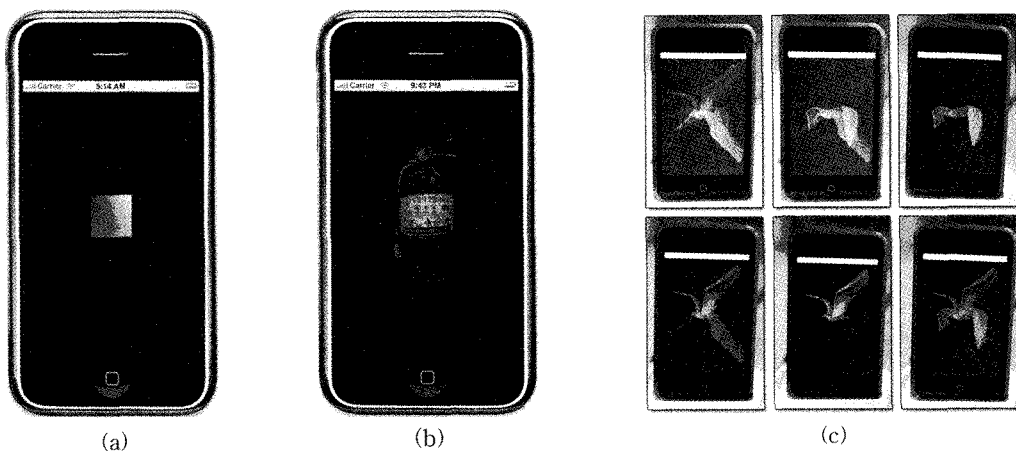


그림 7. 아이폰 플랫폼에서 동작하는 모바일 OSG 기반의 삼차원 응용프로그램들

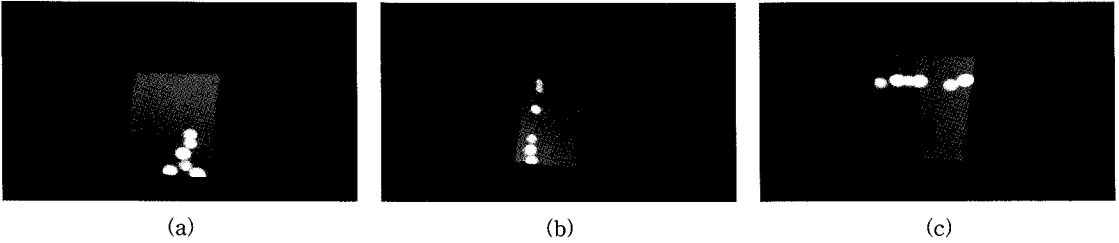


그림 8. 모바일 OSG를 이용한 공 충돌 및 반응 시뮬레이션

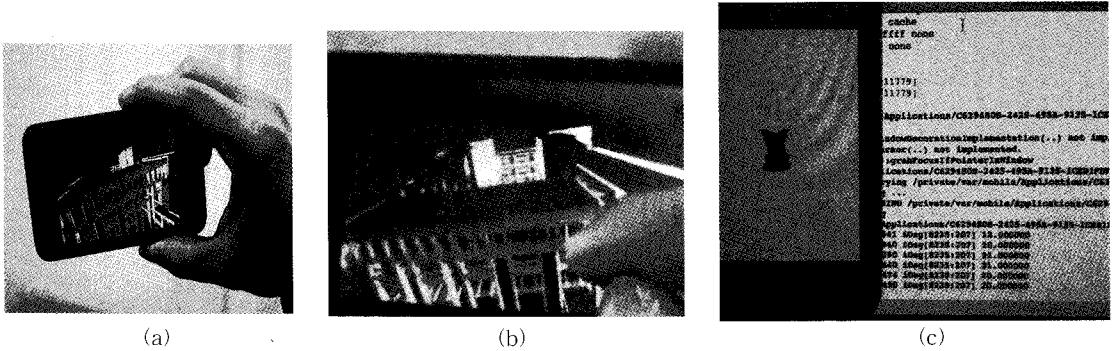


그림 9. 모바일 OSG를 이용한 터치 기반의 삼차원 모델 조작

결과 공들끼리 서로 충돌반응이 일어나도록 하였다. 그림 8에서 볼 수 있듯이 기존의 데스크탑 환경에서 지원하지 않는 기울기 센서 등의 기능이 모바일 OSG에서는 손쉽게 추가될 수 있어 직관적이며 새로운 스타일의 모바일 응용프로그램의 개발이 간편하다는 장점이 있다.

그림 9는 터치 UI를 지원하는 아이폰 상에서 모바일 OSG를 통해 대용량의 삼차원 모델이 조작되는지를 검증한 예제이다. 그림 9(a)는 대용량 삼차원 모델이 모바일 OSG를 통해 로딩되어 실시간 렌더링됨을 보이고 있으며, 그림 9(b)는 터치 UI를 통해 로딩된 대용량 모델의 애니메이션 및 회전, 확대/축소, 이동 등의 인터랙션이 용이하게 이루어지는지 보여주고 있다. 그림 9(c)는 iPhone 3GS에서 OSG를 사용했을 시 렌더링 속도를 측정한 결과인데 10만 폴리곤의 모델을 로딩하여 보았을 때 20 fps 정도의 속도가 나오는 것을 확인할 수가 있었다. PC의 경우(Intel Core2 Quad Q6600, NVidia GeForce 9600 GT 사용)인 60 fps 의 속도 보다는 느렸지만 육안으로 iPhone에서 10만 폴리곤 렌더링 시에 느려짐은 그리 확인할 수가 없었다. 이는 본 논문에서 제시한 iPhone OSG의 안정성을 보여주는 결과라고 할 수 있다.

6. 결 론

본 논문에서는 데스크탑 환경에서 널리 쓰이고 있는 그래픽스 미들웨어인 OSG를 모바일 플랫폼을 지원하도록 재설계한 모바일 OSG의 개발 사례를 소개하였다. 본 논문에서 제안한 모바일 OSG는 다양한 모바일 플랫폼을 위한 범용성, 기울기 센서 등과 같은 모바일 기기가 지원하는 기능을 활용할 수 있는 확장성, 터치기반의 UI 지원 등의 특성을 가지고 있다. 다양한 예제를 통해 기존의 데스크탑 응용프로그램이 별다른 소스 코드의 수정 없이 모바일 상에서 구동됨을 보임으로써 모바일 OSG의 효용성을 보였다. 향후 모바일 OSG에 증강현실 기능을 추가하여, 모바일 기기의 카메라 기능을 손쉽게 활용할 수 있는 모바일 증강현실 미들웨어를 개발하고자 한다.

참 고 문 헌

- [1] K. Pulli, T. Aarnio, V. Miettinen, K. Roimela, and J. Vaarala, *Mobile 3D Graphics with OpenGL ES and M3G*, Morgan Kaufmann Pub., 2007.

- [2] P. Martz, *OpenSceneGraph Quick Start Guide*, 2007.
- [3] S. Green, *The OpenGL Framebuffer Object Extension*, GDC 2005, 2005.
- [4] Looser, J., Grasset, R., Seichter, H., and Billinghurst, M. (2006) "OSGART-A Pragmatic Approach to MR," 5th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 06): Industrial Workshop, 22-25 Oct 2006.
- [5] Schmalstieg Dieter and Wagner Daniel, "Experiences with Handheld Augmented Reality," The Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2007), pp. 3-15
- [6] Choonsung Shin, Wonwoo Lee, Youngjung Suh, Hyoseok Yoon, Youngho Lee, and Woontack Woo, "CAMAR 2.0: Future Direction of Context-Aware Mobile Augmented Reality," International Symposium on Ubiquitous VR 2009, pp. 21-24, 2009
- [7] Sylvia Irawati, Sangchul Ahn, Jinwook Kim, and Heedong Ko, "VARU Framework: Enabling Rapid Prototyping of VR, AR and Ubiquitous Applications," *IEEE Virtual Reality*, 2008
- [8] Coin3D, <http://www.coin3d.org/>
- [9] OpenSG, <http://www.opensg.org/>



김 준 호

1998년 포스텍 컴퓨터공학과 공학사
 2000년 포스텍 컴퓨터공학과 공학석사
 2005년 포스텍 컴퓨터공학과 공학박사
 2005년 3월~2005년 10월 포스텍

박사후 연구원

2005년 11월~2008년 2월 뉴욕주립대(스토니브룩 소재) 포스트닥 연구원

2008년 3월~2009년 2월 동의대학교 게임공학과 전임강사
 2009년 3월~현재 국민대학교 컴퓨터공학부 전임강사
 관심분야: 컴퓨터 그래픽스, 디지털 영상처리, 인터랙티브 게임



서 진 석

1998년 건국대학교 전자계산학과 공학사
 2000년 포스텍 컴퓨터공학과 공학석사
 2005년 포스텍 컴퓨터공학과 공학박사
 2005년~현재 동의대학교 게임공학과 조교수

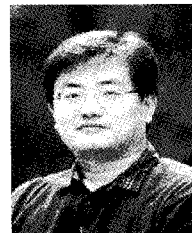
관심분야: 컴퓨터 게임, 저작도구, 가상/증강 현실



박 창 훈

1995년 2월 단국대학교 전자계산학과(이학사)
 1997년 2월 단국대학교 전자계산학과(이학석사)
 2003년 2월 고려대학교 컴퓨터학과(이학박사)
 2006년 3월~현재 호서대학교 게임공학과 교수

관심분야: 게임, 가상현실



고 희 동

1982년 State U. of New York (SUNY) at Albany 응용수학 학사
 1989년 University of Illinois at Urbana-Champaign (UIUC) 전산과학 박사

1990년~현재 한국과학기술연구원 영상미디어센터장
 관심분야: 가상현실, 증강현실, QoLT, 인공지능



황 재 인

1998년 포스텍 전자계산학과 학사
 2000년 포스텍 컴퓨터공학과 석사
 2007년 포스텍 컴퓨터공학과 박사
 2008년~현재 한국과학기술연구원 선임연구원

관심분야: 모바일 증강현실, 모바일 인터랙션, 가상현실