

석사학위논문

HTML5 WebGL 기반 3D 지형공간
정보처리 시스템 설계 및 구현

2013년

한성대학교 대학원

정보시스템공학과

정보시스템공학전공

김 광 섭

석사학위논문
지도교수 이기원

HTML5 WebGL 기반 3D 지형공간 정보처리 시스템 설계 및 구현

Desing and Implementation of 3D Geo-spatial information
Processing System based on HTML5

2013년 06월 일

한성대학교 대학원

정보시스템공학과

정보시스템공학전공

김 광 섭

석사학위논문
지도교수 이기원

HTML5 WebGL 기반 3D 지형공간 정보처리 시스템 설계 및 구현

Desing and Implementation of 3D Geo-spatial information
Processing System based on HTML5

위 논문을 공학 석사학위 논문으로 제출함

2013년 06월 일

한성대학교 대학원

정보시스템공학과

정보시스템공학전공

김 광 섭

김광섭의 공학 석사학위논문으로 인준함

2013년 06월 일

심사위원장 _____ 김 남 윤 인

심 사 위 원 _____ 이 기 원 인

심 사 위 원 _____ 이 관 우 인

국 문 초 록

HTML5 WebGL 기반 3D 지형 공간 정보처리 시스템 설계 및 구현

한성대학교 대학원
정보시스템공학과
정보시스템공학전공
김 광 섭

정보 통신 기술 분야의 스마트폰이 대중화가 되면서 생활환경이 빠르게 변화 되고 있으며, 다양한 분야에서는 이를 이용한 여러 가지 연구 및 서비스가 진행되고 있다. 최근 모바일 분야의 기기는 스마트폰 외에도 태블릿 PC, 스마트 TV, 스마트 자동차 등 다양한 기기들이 출시되면서 활용성이 더욱 더 커지고 있다. 이러한 스마트 기기들은 개발자가 직접 어플리케이션을 개발하여 배포 할 수 있어, 기존 고정된 서비스를 제공한 피쳐 폰에 비해 다양한 서비스를 사용자에게 제공 할 수 있는 장점이 있다. 공간 정보 분야에서도 스마트폰을 이용한 지도검색 서비스, 위치기반 서비스(LBS: Location based service) 등 다양한 앱이 제공되고 있다. 하지만 이러한 서비스에서 사용되고 있는 공간정보 영상은 단순한 시각화에만 사용되며, 스마트폰에서 직접 처리하거나, 분석이 되는 연구 및 서비스는 미흡한 실정이며, 앱뿐만 아니라 웹에서도 다양한 공간정보 서비스가 제공이 되고 있지만 상황은 동일한 실정이다.

이러한 점을 고려하여 이 연구에서는 스마트폰을 통해 다양한 공간 정보의 지형공간 데이터를 3차원 시각화와 동시에 정보처리 할 수 있는 웹 시스템을 설계 및 구현하였다. 기존의 웹 시스템은 데스크톱 웹 브라우저만을 고려하였지만 최근 데스크 탑 뿐만 아니라 모바일에서 같은 콘텐츠를 보여주기 위한 기술인 HTML5(Hypertext Markup Language5)를 이 연구에 적용하여 구현

함으로써 다양한 플랫폼에 구동이 가능하도록 하였다. HTML5는 현재 월드 와이드 웹 컨소시엄(W3C: World Wide Web Consortium)과 WHATWG(Web Hypertext Application Technology Working Group)에서 공동으로 작업하고 있는 차세대 웹 기술이다. HTML5는 기존 HTML에서 플러그인(Plug-in)을 사용한 서비스를 별도의 플러그인을 사용하지 않고 서비스를 하기 위한 목적이 강한 국제 표준이다. 그렇기 때문에 웹 언어만을 이용한 다양한 라이브러리가 포함되어 있으며, 이 연구에서는 HTML5 기술 중 지형 공간 3D 시각화를 위해 WebGL(Web Graphics Library)을 적용하였다. WebGL은 OpenGL ES 2.0(OpenGL for Embedded Systems 2.0) 기반으로 되어 있으며, 기존 기술인 웹 브라우저에서 플러그인을 설치해야 하는 어도비(Adobe)의 플래시(Flash)나 마이크로소프트(Microsoft)의 실버라이트(Silverlight)와는 다르게 플러그인 없이 동적으로 3D 기술을 적용 할 수 있어 모바일에도 적용이 가능하다. 웹 시스템에 사용된 지형공간 데이터는 우리나라의 행정중심복합도시인 세종시 지역의 광학(Optical) 영상 데이터, 합성개구레이더(SAR: Synthetic Aperture Radar) 영상 데이터, 환경 센서 데이터, 수치지도(Digital Map) 데이터이다. 광학 데이터는 우리나라 위성인 KOMPSAT-2 영상과 아랍에미리트의 DUBAISAT-1 영상, SAR 데이터는 일본의 ALOS PALSAR 영상, 환경 데이터는 세종시 인근 고정형 환경 센서 측정소 8군데 데이터를 이용하여 그리드 데이터를 생성하여 사용하였다. 정보 처리에 사용된 기술로는 서버와 통신을 하기 위한 JQuery, WebGL을 이용한 블렌딩 기법, 공간정보 분야에서 많이 사용되고 있는 융합 알고리즘 중 계산법이 간단한 IHS(Intensity-Hue-Saturation) 융합 알고리즘을 적용하여 실시간 융합 처리를 적용하였다.

이 연구에서는 데스크톱뿐만 아니라 스마트폰에서의 3D 공간정보를 처리할 수 있는 웹 시스템을 설계 및 구현하여 앞으로 웹 브라우저에서의 지형 정보 처리에 대한 기초 연구로 기여 될 것으로 예상된다.

【주요어】 HTML5, WebGL, 모바일 웹 앱, 3D 지형공간 정보처리, 오픈 소스

목 차

제 1 장 서 론	1
제 1 절 연구의 배경	1
제 2 절 연구의 목적 및 방법	4
제 2 장 지형공간 모바일 앱 및 정보처리	6
제 1 절 모바일 앱	6
1. 모바일 앱 종류	7
2. 지형공간 모바일 앱	9
3. 모바일 웹 앱	10
제 2 절 HTML5 WebGL	12
1. HTML5 Javascript API, CSS3	13
2. WebGL	17
제 3 절 3D 지형공간 정보처리	23
제 3 장 3D 지형공간 정보처리 시스템 설계 및 구현	25
제 1 절 지형공간 데이터	25
제 2 절 3D 지형공간 전처리	29
제 3 절 3D 지형공간 정보처리 적용기술	32
1. jQuery Ajax	33
2. WebGL CubicVR.js	35
3. 블렌딩	37
4. IHS 융합	39
제 4 절 3D 지형공간 정보처리 시스템 설계	40
1. 오픈 소스 적용 시스템 환경	41

2. 시스템 구성도 및 흐름도	42
제 4 장 3D 지형공간 정보처리 시스템 구현 결과	45
제 1 절 3D 다중 시각화 구현 결과	45
1. 광학 영상 3D 다중 시각화	48
2. 광학 영상과 다양한 지형 정보 3D 다중 시각화	50
제 2 절 광학 영상과 SAR 영상 융합 구현 결과	54
제 5 장 결 론	57
【참고문헌】	59
ABSTRACT	64

【 표 목 차 】

[표 2-1] 모바일 네이티브 앱과 모바일 웹 앱 비교	10
[표 2-2] W3C 웹 애플리케이션 관련 표준 (전종홍과 이승윤, 2012)	14
[표 2-3] 데스크톱 웹 브라우저 별 HTML5 지원 현황	15
[표 2-4] 모바일 웹 브라우저 별 HTML5 지원 현황	16
[표 2-5] WebGL 프레임워크, 유틸리티, 디버그 헬퍼	21
[표 3-1] 에어 코리아 대기 환경 기준	31
[표 3-2] CubicVR.js API 문서	36
[표 3-3] 블렌딩 기법 별 WebGL 함수 및 식	39
[표 3-4] 3D 지형공간 정보처리 웹 시스템 환경	42
[표 4-1] 장치별 테스트 웹 브라우저 버전	47

【 그림 목 차 】

〈그림 2-1〉 모바일 앱과 모바일 웹 트래픽 비교	7
〈그림 2-2〉 지형공간 모바일 앱 국내·외 서비스 사례	8
〈그림 2-3〉 WebGL 그래픽 파이프라인	20
〈그림 2-4〉 HTML5 WebGL의 브라우저에서의 3개의 주요 구성요소 역할	20
〈그림 2-5〉 다중 유형 공간정보 융합 개념	24
〈그림 3-1〉 연구에 사용된 광학 및 SAR 원본 데이터	26
〈그림 3-2〉 여덟 개의 대기 환경 측정소 및 연구 지역	28
〈그림 3-3〉 이 연구에 사용된 세종시 지역 최신 수치지도	28
〈그림 3-4〉 측정소 정보 및 IDW 그리드 데이터	30
〈그림 3-5〉 3D 실시간 정보처리 시스템에 사용된 광학 데이터 및 SAR 데이터	31
〈그림 3-6〉 블렌딩을 이용한 시간에 따른 3D 시각화에 사용된 광학 데이터	32
〈그림 3-7〉 Google Trends 자바스크립트 라이브러리 사용률	34
〈그림 3-8〉 3D 지형공간 정보처리 웹 시스템 구성도 및 흐름도	45
〈그림 4-1〉 사용자 인터페이스 구성	48
〈그림 4-2〉 3D 지형공간 정보처리 시스템 메뉴 구성	48
〈그림 4-3〉 데스크톱, 스마트폰, 태블릿 PC에서의 광학 영상 3D 다중 시각화	50
〈그림 4-4〉 데스크톱 및 스마트폰에서의 광학 영상(KOMPSAT-2) 및 환경 그리드 데이터 3D 다중 시각화	52
〈그림 4-5〉 데스크톱 및 스마트폰에서의 광학 영상(KOMPSAT-2) 및 SAR 영상(ALOS PALSAR) 데이터 3D 다중 시각화	53
〈그림 4-6〉 블렌딩 기법에 따른 적용 결과	54
〈그림 4-7〉 데스크톱 및 스마트폰의 IHS 융합 메뉴 구성	56
〈그림 4-8〉 IHS 융합 과정	57

제 1 장 서 론

제 1 절 연구의 배경

정보 통신 기술(ICT: Information and Communication Technology)의 모바일기기들이 스마트폰 등장과 함께 현대 사회에 없어서는 안 되는 기술로 보편화 되면서 사회적으로 다양한 변화가 생겨나고 있다. 과거 대부분의 모바일기기로 출시되었고 아직까지 소규모로 출시되고 있는 피쳐 폰은 현재 대부분 사람들이 사용하고 있는 스마트폰보다 낮은 연산 능력을 가지고 있어 활용 가치가 저조하였다. 하지만 모바일 산업의 스마트폰과 태블릿 PC 등 다양한 스마트기기들이 출시되면서 이를 이용하여 다양하고 편리한 생활이 가능해지고 있다. 한국인터넷진흥원의 2012년도 상반기 스마트폰 이용실태조사에 따르면 스마트폰 이용자의 85.5%가 ‘스마트폰 이용 후 생활 전반이 편리해졌다’고 응답하였다. 이러한 스마트기기에는 가속도(Acceleration), 중력(Gravity), 카메라(Camera), 자이로스코프(Gyroscope), 위성항법장치(GPS: Global Positioning System) 등 센서들이 탑재 되어 있으며, 향후 인간이 가지고 있는 5가지 감각인 오감에 대한 정보도 적용하여 사용 할 수 있을 것으로 예상하고 있다 (KT 종합기술원, 2010; 권진혁, 2012). 또한, 기존의 폐쇄적 이었던 모바일 앱 시장이 스마트폰의 보급화로 인해 개방형으로 바뀌면서 많은 변화가 일어나고 있다. 개방형 모바일 시장은 범용 운영체제 기반의 모바일 플랫폼인 안드로이드와 iOS 등이 등장하면서 변화를 가지고 왔으며, 대표적인 사례로 구글 플레이와 애플의 앱 스토어가 있다. 이러한 모바일 시장은 양면 시장 형태를 나타내고 있으며 각각의 마켓플레이스는 기존 핵심역량을 활용할 뿐만 아니라, 영역 확장을 통해 개발자 및 이용자를 동 시장으로 유인함으로써 어플리케이션 마켓 활성화에 주력하고 있다 (권지인, 2009).

현재 스마트기기에 다양한 운영체제를 탑재한 장치들이 나오고 있으며, 대표적인 사례로 구글의 안드로이드, 애플의 iOS, 마이크로소프트사의 윈도우 모바일, 리서치 인 모션(RIM: Research In Motion Limited)의 블랙베리 등

이 있다. 이러한 모바일 기기에 서비스를 모두 하기 위해서는 개발 언어가 다르기 때문에 같은 서비스를 하더라도 운영체제별 개발을 별도로 해야 한다. 또한, 운영체제가 같더라도 장치의 해상도가 다를 경우 구동이 안 될 가능성이 있다. 그렇기 때문에 스마트폰이 보급화 되면서 웹 시장 또한 변화가 되고 있다. 스마트폰에는 데스크톱과 동일하게 웹 브라우저가 탑재되어 있지만 현재 사용되고 있는 웹 기술로는 동시에 서비스하기 어려운 점이 있다. 웹 기술의 HTML(Hypertext Markup Language)은 W3C에서 1995년부터 국제 표준으로 관리를 하고 있으며, 초기에는 정보를 단순하게 보여주는 수준의 서비스를 제공하였다. 하지만 현재는 서로 상호작용 하는 웹 앱 형태로 발전하고 있으며, 최근 W3C에서 초안을 발표하고 현재 표준화가 진행되고 있는 HTML5는 아직 국제 표준화가 진행 중이지만 이미 산업계에서는 이를 이용하여 다양한 서비스를 진행하고 있다. 웹 언어로 개발하여 서비스를 할 경우 플랫폼과 독립적으로 구동되기 때문에 한 번의 개발로 다양한 장치에 적용이 가능하며, 데스크톱 뿐 아니라 해상도가 다양한 스마트폰의 웹 브라우저에서 자동으로 동일한 콘텐츠를 알맞게 적용하여 보여 줄 수 있다. 그렇기 때문에 최근 산업계에서는 모바일 앱 서비스뿐만 아니라 HTML5를 이용한 모바일 웹 앱에 대한 서비스도 동시에 증가 하고 있다.

이러한 발전에 따른 모바일 앱 서비스들은 다양한 분야에서 영향을 받고 있으며, 공간정보 분야에서도 많은 영향을 받고 있다. 공간정보 서비스는 공간에 관한 정보를 생산·관리·유통하거나 다른 산업과 융·복합하여 시스템을 구축하고 제공 하는 서비스를 의미하며, 위치 기반 서비스(LBS: Location Based Services), 증강현실(AR: Augmented Reality), RFID(Radio-Frequency Identification), Geo-Web 등이 있다 (손맥 등, 2010). 이 중에 위치 기반 서비스는 스마트폰에 탑재된 GPS를 이용하여 현재 자신의 위치를 근거로 제공하는 서비스로 네비게이션, 위치추적, 운전편의정보, 친구 찾기, 안전 서비스 등이 있다. 가트너(Gartner)는 2012년 가장 주목받을 스마트폰 앱 중의 하나로 위치 기반 서비스를 선정하였다 (전황수, 2011). 최근에는 모바일 앱 뿐만 아니라 모바일 웹에서도 공간 정보 서비스를 하기 위한 연구가 많이 진행되고 있으며, HTML5를 이용하고 있다 (Boulos, 2010; Sauerwein, 2010;

Johansson, 2010; Butchar, 2010; Corcoran, 2011). 또한, 데스크톱과 스마트기기에서 온라인 지도 서비스 제공이 보편화 되는 추세이다. 특히 위성 영상과 같은 지형공간 데이터들은 다양한 용도로 사용되고 있으며, 활용하는 서비스 중 대표적인 국외 지도 서비스로는 구글 맵, 야후 맵, 마이크로소프트사의 Bing 맵 등이 있으며 국내 지도 서비스로는 네이버 지도, 다음 지도 등이 있다. 하지만 이러한 서비스들은 위성 영상을 단순 배경 영상으로만 사용 하고 있으며, 정보 처리에 대한 기능은 가지고 있지 않다. 처리를 목적으로 사용되고 있는 소프트웨어는 상업적 소프트웨어가 대부분이며, 클라이언트 기반으로 보급되고 있다. 지형공간 상업적 소프트웨어는 개인적으로 구입을 하거나 단체로 구입 하기는 소프트웨어 가격이 비싸 쉽게 환경을 갖추기 어려운 실정이다.

또한 공간정보를 모바일에서 3D 시각화하기 위한 연구는 과거에서부터 지속적으로 연구가 되고 있다. 3D 공간 정보 기술은 초기에 3D 지형 가시화부터 시작하여 2000년대에 컴퓨터 그래픽 기술의 발전으로 인해 능동적 3D 가상세계 구현 까지 가능하게 되었다 (박세환, 2011). 모바일에서 3차원으로 시각화 및 처리하는 연구들로 Kim (2006)은 PDA(Personal Digital Assistant) 단말에서 OpenGL ES 1.0(OpenGL for Embedded Systems)를 이용하여 3D 공간정보 처리를 위한 시스템을 개발한 바 있으며, Lee (2008)은 3D 공간 객체 렌더링을 위하여 M3G API(Mobile 3D Graphic Application Programming Interface)를 적용 한 바 있다. 선행된 연구들과 같이 모바일에서 3D 기술을 구현하기 위해서는 OpenGL ES를 사용하거나 M3G API를 사용했다. 현재 스마트폰에서도 3D 기술이 접목된 앱을 구현하기 위해서는 OpenGL ES 2.0 기술을 사용하고 있다. 스마트폰에서의 OpenGL ES 2.0을 이용하여 3D 시각화가 가능하지만 앞서 언급한 것과 같이 운영체제별 개발을 별도 해야 한다는 단점이 있다. 하지만 최근에 OpenGL ES 2.0 기반인 HTML5의 WebGL이 발표되면서 웹 브라우저 상에서도 3D 구현이 가능하게 되어 다양한 스마트기기에 3D 기술을 적용 할 수 있게 되었다.

제 2 절 연구의 목적 및 방법

최근 스마트폰 뿐 아니라 다양한 해상도를 가진 스마트기기들이 출시가 되고 있다. 이러한 스마트기기들은 빠른 속도로 높은 사양이 탑재되어 출시되고 있으며, 최근 데스크톱과 비교해도 대등 할 정도의 사양이 출시되고 있다. 하지만 아직까지 공간 정보 분야의 스마트기기의 앱 또는 웹들은 대부분 위치 기반 서비스이거나, 지도 찾기 서비스며, 사용되고 있는 위성 영상 지도들은 단순한 배경 영상으로 사용 되고 있다. 위치 기반 서비스나 지도 찾기 등 일반 사용자를 위한 서비스도 중요하지만, 다양한 위성 영상을 직접 처리 하거나 분석하는 전문 서비스에 대한 고려도 필요하다. 하지만 아직까지 스마트폰에서 위성 영상을 가지고 직접 처리하는 어플리케이션에 대한 연구는 미흡한 실정이다. 현재 데스크톱에서 지형 정보를 처리하기 위한 상용 소프트웨어 및 공개 소프트웨어들이 많이 출시되어 사용 되고 있다. 하지만 이러한 소프트웨어는 플랫폼에 종속되어 설치를 하여 사용해야 하며, 사용법 또한 복잡하다. 위성 영상 처리가 아닌 이미지 처리 기능을 제공하고 있는 스마트폰 앱은 현재 서비스가 되고 있는 단계이다. 실제 이미지를 기본 처리 또는 고급 처리까지 모바일에서 직접 할 수 있도록 제공하고 있다. 위성 영상 이미지는 일반 이미지 처리보다 단계가 복잡하며, 데이터 크기도 상대적으로 크기 때문에 고려해야 될 사항이 많다. 이러한 사항을 고려하여서 이 연구에서는 웹 브라우저가 탑재되어 있는 데스크톱뿐만 아니라 스마트기기에서도 지형 영상을 처리 가능한 웹 앱을 설계 및 구현하였다. 웹 브라우저에서 다양한 공간 정보 영상을 3D 시각화와 동시에 실시간으로 영상을 처리 할 수 있도록 구현하였으며, 웹 브라우저에서 구동되므로 별도의 설치 필요 없이 사용 할 수 있고 플랫폼에도 독립적이다. 이번 연구의 목적으로는 스마트기기에서도 위성 영상을 단순 배경 영상으로 사용 하는 것이 아니라 전문가를 위한 처리를 할 수 있는 가능성을 보여주며, 설계 및 구현 시 오픈 소스를 이용하여 향후 서비스로 발전이 될 경우 비용 절감 및 다양한 장점을 이용하는 것이다. 이 연구는 향후 웹상에서 영상 정보 처리 기술을 사용 할 수 있는 어플리케이션에 대한 기초 연구가 될 것으로 예상된다.

이 연구의 연구 방법은 다음과 같다. 현재 서비스되고 있는 모바일 앱의 형태는 현재 네이티브 앱, 모바일 웹 앱, 하이브리드 앱이 있다. 이러한 세 종류의 앱 형태의 기본적인 사항을 조사하였고 이 연구에서 구현된 모바일 웹 앱에 대해 심층적으로 조사하였다. 또한 현재 지형공간 관련된 모바일 앱에 대해 조사를 하여 현 실태에 대해 분석을 하였다. 기술적으로는 HTML5를 사용하여 다양한 기기에서 구동이 가능하도록 구현하였다. HTML5는 기존 HTML과 다르게 다양한 Javascript API가 개발 되어 있어 별도의 플러그인 없이 리치 웹 어플리케이션 구현이 가능하다. 그러므로 플러그인 설치가 불가능한 스마트기기에서 사용하기 적합하다. 하지만 아직 HTML5의 표준안이 제정 된 것이 아니므로 이에 대한 분석이 필요하다. 이 연구에서는 HTML5와 이에 속해있는 Javascript API에 대한 사양과 장치 해상도에 맞게 사용자 인터페이스(UI: User Interface)를 적용 할 수 있도록 발표된 종속형 스타일 시트3(CSS3: Cascading Style Sheets3)에 대해서도 분석하였다. HTML5의 Javascript API 중에서 지형공간을 3D 시각화하기 위해 WebGL을 사용하였다. WebGL은 기존 임베디드(Embedded) 장치에 그래픽 가속을 위한 라이브러리인 OpenGL ES 2.0 기반으로 되어 있는 그래픽 라이브러리를 Javascript로 포팅(Porting)한 것으로 웹 브라우저에서 인터랙티브한 3D 그래픽 구현뿐 아니라 블렌딩(Blending) 기법을 사용하여 다중 시각화가 가능하다. 이 연구에서는 WebGL에 대한 사양을 분석하고 이를 편리하게 사용하기 위해 개발 되고 있는 프레임워크들과 연구에서 사용된 CubicVR.js 프레임워크에 대해서 소개하였다. 3D 지형공간 정보처리 시스템에 사용되는 데이터를 소개하고 이를 시스템에 적용하기 위한 전처리 과정을 수행하였으며, 지형 정보를 실시간으로 처리 하는 방법들과 알고리즘 중에서 IHS 융합 방법을 이번 시스템에 적용하였다. 스마트기기 웹 브라우저에서 지형 정보를 처리 하는 연구 사례가 없는 관계로 최대한 간단한 알고리즘을 적용하여 가능성을 테스트하고자 하였다. 이 연구에서는 오픈 소스를 적용하여 기능별 모듈화를 하여 향후에도 적용 할 수 있도록 설계하였고, 다양한 장치의 적용 가능성을 보기 위하여 현재 HTML5의 WebGL을 지원하고 있는 장치들의 웹 브라우저에서 모두 구동하였다.

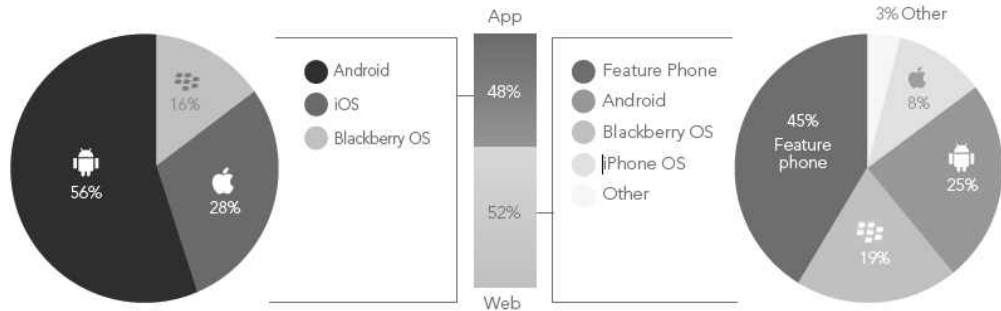
제 2 장 지형공간 모바일 앱 및 정보처리

제 1 절 모바일 앱

과거 피쳐 폰 소프트웨어의 가장 중요한 부분은 통신사업자들이 정한 플랫폼이었고 제조사들은 이 플랫폼을 모뎀 칩셋 개발자로부터 받은 RTOS(Real Time Operating System)에 최적화만 하면 되었지만, 스마트폰은 다양한 앱 구동이 가능한 GPOS(General Purpose Operating System)를 기반으로 구동되기 때문에 소프트웨어의 역할이 피쳐폰보다 훨씬 중요해졌다(신동형과 유미연, 2012). 이러한 변화로 인해 앱 시장도 통신 사업자 중심의 폐쇄형에서 개방형으로 바뀌었다. 개방형 앱 시장의 시작인 대표적 예로는 애플의 앱스토어와 구글의 플레이가 있다. 이러한 마켓에서는 현재 다양한 소프트웨어들이 출시되고 있다. 그리고 대부분의 사람들은 스마트폰을 이용하게 된 계기로는 ‘다양한 모바일 앱을 사용하고 싶어서’ 라고 대답하였고(한국인터넷진흥원, 2012), 사용자들의 마켓에 대한 사용률은 계속적으로 증가하고 있다. 이러한 정보 통신 분야에 대한 변화는 공간 정보 분야에서도 상당한 영향을 주고 있다. 공간정보 분야의 데스크톱의 서비스들은 넓은 범위의 공간 정보를 검색하였지만, 스마트폰이 보급화 되면서 실시간 정보와 자신의 위치에 대한 소규모의 정보를 원하는 경우가 많아 졌다. 이러한 서비스가 가능한 이유는 스마트폰의 가속도, 중력, 카메라, 자이로스코프, GPS 등 다양한 센서들이 탑재되어 있어 이를 이용하여 자신의 위치에 대한 소규모 정보 검색이 가능해 졌다. 일반적인 모바일 지도 앱은 현재 자신의 위치를 GPS 센서를 통해 보여주고 근처의 정보들을 지도상에서 사용자가 알기 쉽게 보여준다.

스마트폰이 보급되면서 모바일 앱뿐만 아니라 모바일 웹에 대한 발전도 급속도로 빨라지고 있다. 데스크톱 웹 브라우저에서 보이는 콘텐츠를 데스크톱과 비교했을 때 해상도가 낮은 스마트폰 웹 브라우저에서도 동일하게 보이도록 서비스가 변화 되고 있다. 또한 단순한 홈페이지가 아닌 웹 앱으로 발전하면서 스마트폰의 모바일 앱과 웹의 사용량은 거의 비슷해 지고 있는 현상이

다. <그림 2-1>은 모바일 앱과 모바일 웹의 비교를 꾸준히 발표 하는 기관인 Jumptap에서 통계한 자료인 2011년 12월 모바일 앱과 웹의 트래픽 비중으로 거의 동일 한 것을 보여주고 있다.



<그림 2-1> 모바일 앱과 모바일 웹 트래픽 비교 (Jumptap, 2011).

이 연구에서는 3D 지형공간 정보처리 시스템을 설계 및 구현하기 전 모바일 앱의 지형공간 관련된 앱에 대한 조사를 수행한 후 현재 상황에 대해 조사하고 모바일 앱의 세 가지 종류인 네이티브 앱, 웹 앱, 하이브리드 앱에 대해 간단히 분석하였으며, 이 연구에서 구현된 시스템인 모바일 웹 앱에 대한 기술분석을 하였다.

1. 모바일 앱 종류

모바일 앱의 종류는 네이티브 앱과 웹 앱으로 나뉜다. 현재 마켓에서 서비스 되고 있는 대부분의 모바일 앱은 네이티브 앱으로 서비스가 되고 있다. 이러한 네이티브 앱들은 각 플랫폼에 맞게 개발이 별도로 진행되고 있으며, 대표적인 스마트폰 운영체제 개발 언어로 애플의 iOS기반의 Object-C, 구글의 안드로이드의 Java, 마이크로소프트사의 윈도우 모바일의 C#/C++이 있다. 모바일 네이티브 앱은 속도가 빠르고 장치에 탑재되어 있는 센서들을 사용하기에 유용하지만, 모든 스마트폰에 서비스를 하기 위해서는 오랜 시간의 테스트 및 개발이 필요하다.

모바일 웹 앱은 네트워크 통신이 가능한 장치인 동시에 웹 브라우저가 탑재된 모바일 단말기에서 서비스가 가능하다. 이는 플랫폼에 독립적이며, 웹 브라우저의 영향을 많이 받는다. 모바일 웹 앱의 서비스 대부분은 온라인 환경에서 서버와 통신하는 형태로 이루어져 있다. 개발 환경은 일반적인 웹 개발 환경이며, HTML, CSS, Javascript를 사용한다. 하지만 기존의 모바일 웹 앱은 네이티브 앱보다는 스마트폰의 다양한 센서들을 사용이 용이하지 않다. 하지만 최근 HTML5가 발표되면서 모바일 웹 앱에서도 스마트폰의 다양한 센서들을 사용 할 수 있도록 API를 제공하고 있어 다양한 센서를 웹 언어만으로 사용 할 수 있다. 모바일 웹 앱은 2가지로 나눌 수 있다. 일반적으로 웹 브라우저에서 실행되는 앱을 웹 앱이라고 부르고 있으며, 또 하나는 하이브리드 앱으로 네이티브 앱과 모바일 웹 앱이 공존하는 형태이다. 하이브리드 앱은 네이티브 앱과 동일하게 설치형 앱으로 마켓에 등록되어 앱을 다운 받아 사용하게 된다. 하지만 기존 개발 방식인 Java, Object-C를 사용하는 것이 아닌 웹 언어를 사용하여 개발 한 후 하이브리드 앱 플랫폼에 빌드를 하기 때문에 다양한 운영체제를 탑재한 스마트폰에 서비스를 할 수 있을 뿐만 아니라 하이브리드 앱 플랫폼의 API를 이용하면 다양한 장치 센서들의 사용도 용이하다는 장점이 있다. 그렇기 때문에 최근에는 하이브리드 앱에 대한 서비스를 선호 하고 있는 실정이다. 국내에서 현재 서비스 되고 있는 대표적인 하이브리드 앱으로는 푸딩얼굴인식, 푸딩카메라, 아임IN 등이 있으며, 공간정보 분야에서의 하이브리드 앱 관련된 연구로는 김광섭과 이기원(2001(a))은 위성 영상정보의 공간 메타데이터 검색 서비스를 설계 및 구현하였으며, 김광섭과 이기원(2011(b))에서도 GPS를 이용하여 대기환경정보와 방사능정보 위치기반 서비스 하이브리드 앱을 설계 및 구현한 사례가 있다. 또한 Kim (2011)에서도 오픈 소스인 OpenLayers의 Open Street Map에 실제 위성영상과 수치지도 데이터 시각화를 하이브리드 앱으로 구현된 적이 있다. [표 2-1]은 세부적으로 모바일 네이티브 앱과 모바일 웹 앱과 하이브리드 앱에 대해 특징을 비교한 표이다.

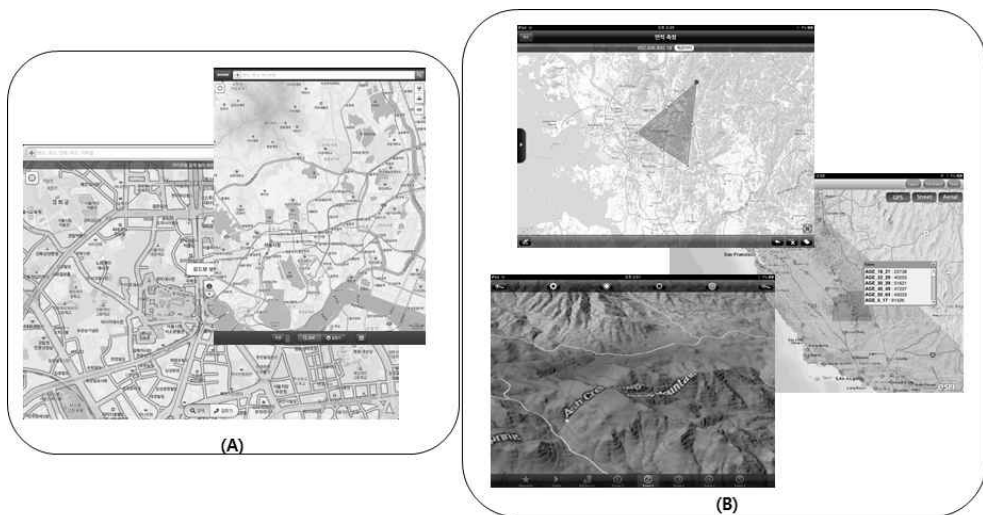
[표 2-1] 모바일 네이티브 앱과 모바일 웹 앱 비교

구 분	모바일 네이티브 앱	모바일 웹 앱
호환성	· 탑재된 운영체제마다 상이	· 웹 브라우저마다 상이
장 점	· 속도가 빠름 · 장치 센서기능 용이	· 유지보수가 용이 · 개발 시간 단축
단 점	· 유지보수 절차가 복잡 · 플랫폼 API 별도 학습	· 상대적으로 속도가 느림 · 장치 센서기능 활용 어려움
개 발 수 단	· 플랫폼별도 개발언어 - iOS: Object-C - Android, RIM : Java - Window Mobile : C#	· HTML5, CSS3, Javascript · 프레임워크 - jQuery, jQTouch, SenshTouch · 하이브리드 앱 프레임워크 - Phonegap, Titanuim, Appspresso, Rhomobile
등 록	· 전용 마켓에 등록	· 별도 등록 필요 없음 · 하이브리드 앱의 경우 전용 마켓 등록
테스트	· 적용되는 단말기별 테스트	· 브라우저별 엄격한 테스트

2. 지형공간 모바일 앱

스마트폰 앱은 다양한 분야에서 서비스 되고 있으며 공간정보 분야에서도 스마트폰을 이용하여 다양한 앱들이 출시되어 서비스 되고 있다. 지형공간정보 활용에 대한 기초적인 스마트폰 기능으로는 지도 조작, 위치 조회, 공간 검색 등이 있다. 지도 조작 기능의 세부 기능으로 지도를 불러오거나 축소 및 확대 그리고 이동하는 것이며, 위치 조회 세부 기능으로 스마트폰의 GPS를 이용하여 자신의 위치를 표시하는 기능이며, 공간 검색 세부 기능으로 기존 단순 지도 서비스가 아닌 관심지역(POI: Point of Interest)정보를 적용하여 건물, 시설물, 주소 등을 지도상에서 검색하는 기능이다. 이 외에도 위치기록 조회, 시간 및 거리 계산 등 다양한 기능들이 포함된다. 현재 국내·외에서는 기본적으로 이러한 기능들이 포함된 지형공간 모바일 앱이 다양하게 출시되고 있으며 많이 사용되고 있다. 국내에서 대표적으로 서비스 되고 있는 모바일 앱으로 포털 사이트들의 지도 서비스이다. 국내 포털 사이트들의 지형공간 모바일 앱은 기본적으로 지도 조작, 위치 조회 등이 포함되어 있으며 추가적

으로 로드 뷰, 항공 뷰 등 다양한 기능들이 추가되어 있다. 이러한 지형공간 모바일 앱을 통해 목적지 주변 관광, 교통, 숙박, 음식점 등을 쉽게 확인 할 수 있다는 장점이 있다. 또한 국토해양부의 주요사업으로 개발된 스마트 NSDI 앱, 개인이 개발한 한반도 위성사진 앱은 천리안 위성사진을 통해 실시간으로 한반도의 날씨를 보여주고 있다. 국외에서는 다양한 기능이 포함된 모바일 앱을 제공 하고 있다. 애플의 앱스토어에는 ScenicMapGC와 MyTopoMaps와 같이 지형도(Topographic map)를 실세계와 비슷하게 3D로 보여주는 앱과 수치지도를 직접 앱에서 보여주는 앱도 서비스가 되고 있다. <그림 2-2>는 국내·외에서 서비스가 되고 있는 지형공간 모바일 앱을 보여준다.



<그림 2-2> 지형공간 모바일 앱 국내·외 서비스 사례: (A) 국내 위치 기반 서비스 모바일 앱, (B) 국외 면적 계산, 지형도 3차원 지도 서비스.

3. 모바일 웹 앱

기존의 웹 서비스들은 정적인 HTML 문서로 구성되어 하이퍼링크를 통해 단순히 자신이 읽을 문서 페이지로 이동하는 정도의 상호작용만 있는 한계점이 있었다. 하지만 웹에 대한 꾸준한 발전으로 인해 단순한 웹 집합체가 아닌

웹 앱을 제공하는 하나의 완전한 플랫폼으로의 발전이 되었으며(채원석 등, 2012), 스마트폰이 보급화 되면서 데스크톱 웹 앱뿐만 아니라 모바일 웹 앱에 대한 관심도 증가되었다. 웹 앱은 인터넷이나 인트라넷을 통해 웹 브라우저에서 이용할 수 있는 응용 소프트웨어이며 웹 브라우저에서 구동이 되므로 인터넷만 연결 되어 있으면 누구든지 사용 할 수 있다. 웹 앱 서비스로는 메일, 온라인 전자상거래, 게시판, 블로그 등이 있으며 최근에 HTML5가 발표되면서 웹 앱 서비스는 발전되어 게임과 같은 복잡한 응용 프로그램들도 나오고 있다. 웹 앱의 구조는 크게 서버와 클라이언트로 나뉜다. 웹 브라우저가 클라이언트 역할을 하며, 서버로 페이지 요청을 하게 되면 내부의 프로그래밍 로직을 처리 하는 웹 어플리케이션 서버(WAS: Web Application Server)를 통해 수행되고 그 결과를 다시 클라이언트로 보내주게 된다. 서버에서 처리되는 로직의 경우 상관이 없지만 모바일에서는 데스크톱보다 해상도가 낮으므로 클라이언트 부분에서 고려해야 할 부분이 많이 있다. 이를 해결 할 수 있는 기술적인 방법으로 위에서 언급했듯이 HTML5를 사용한 라이브러리가 출시되고 있다. 그렇기 때문에 최근에는 모바일 웹 앱은 네이티브 앱과 같은 사용자 인터페이스를 구현 할 수 있다 (Power, 2011). IT 시장조사기관인 Strategy Analytics에서는 모바일의 HTML5 지원 단말기 전망이 지속적으로 높아 질 것이라고 예측하고 있으며 2016년까지 85%정도의 단말기가 HTML5의 모든 기술을 지원 할 것으로 예측하고 있다.

공간정보 분야에서의 HTML5를 사용한 지형공간 모바일 웹 앱은 국외에서 다양한 연구 및 서비스가 진행되고 있다. Boulos는 모바일 웹 브라우저 상에서 HTML5의 Canvas를 이용하여 지도에 다양한 데이터를 매핑(Mapping)하는 연구를 수행 한적 있으며, Butchar (2010)은 LBS(Location Based Services)를 하이브리드 앱으로 구현하는 방법에 대해 소개 한적 있다. 또한 Johansson (2010)에서도 HTML5의 SVG, Video, Caching을 이용하여, 프로토타입 지도 웹 앱을 구현 한 적이 있다. 국내에서도 지도 서비스 및 다양한 공간정보 서비스들이 웹상에서 제공되고 있지만 HTML5를 사용하지 않고 기존의 HTML을 사용하여 어도비의 플래시나 마이크로소프트의 실버라이트 등을 기반으로 구동이 되므로 모바일 웹 브라우저에서 사용하기엔 어려운 점이

있다.

제 2 절 HTML5 WebGL

HTML은 소프트웨어 및 하드웨어와 관계없이 웹에 접근 할 수 있는 호환성이 중요하다고 보고, 웹에서 통용될 수 있는 표준안을 제안하고 테스트와 피드백을 반영한 최종 권고안을 마련하는 일을 진행해왔다 (이은민, 2011). 1990년대에 HTML2.0 부터 시작하여 HTML4.01 등 HTML관련 표준안들을 제안하며 웹 환경 확산에 크게 기여를 하였으며, 2000년 이후 웹 환경은 참여, 공유, 개방으로 바뀌게 되고 스마트폰 보급을 통해 모바일 웹 등으로 빠르게 변화하였다. 하지만 W3C에서는 기존 HTML에 대한 개선보다는 XHTML(eXtensible HyperText Language), 클라이언트와 서버 간 데이터 주고받는 AJAX(Asynchronous JavaScript and XML) 등에 대한 논의가 많이 일어났다.

이에 반해 웹 브라우저 개발 업체들은 W3C와는 별도로 하이퍼텍스트 어플리케이션 워킹그룹(WHATWG: Web Hypertext Application Technology Working Group)을 만들고 “Web Application 1.0”이라 불리는 표준을 제안하였다. 이 제안이 개발자들에게 관심을 받자 2007년부터 W3C와 공동 작업을 시작하면서 HTML5라 명칭을 바꿔 국제 표준 작업으로 진행을 하고 있다. HTML5는 2014년 말에 최종 권고안이 공개가 될 예정이며, 현재 초안이 발표 된 상태이다. HTML5는 W3C 회원 외에도 기업, 개발자, 개인 등 누구나 이에 대한 의견을 제기 할 수 있어 사업계에서는 이를 사용하여 다양한 서비스를 출시하고 있다. HTML5는 기존의 HTML 및 XHTML과 호환이 가능하지만 처리 방식 및 생략 마크업 같은 HTML의 난해한 SGML(Standard Generalized Markup Language) 문법과는 호환되지 않는다 (W3C, 2012). HTML5는 기존의 HTML과 XML 문법으로 작성되었으며, 웹 앱을 개발하기 위한 다양한 마크업(Markup) 및 Javascript API가 제안되고 있으며, 다양한 해상도에 적용 할 수 있도록 CSS3가 발표되었다.

1. HTML5 Javascript API, CSS3

HTML5는 다양한 마크업 및 Javascript API가 제공되고 있어 웹 언어만으로 고급스러운 웹 앱을 설계 및 구현 할 수 있다. 이러한 사항은 기존에 액티브X(ActiveX), 플래시 같은 플러그인을 설치하여 사용을 하였던 기능들을 플러그인 없이 사용자에게 서비스 할 수 있도록 도와주고 있으며 모바일에서도 동시에 서비스가 가능해졌다. 핵심기술의 종류로는 멀티미디어 부분의 2D/3D 그래픽 관련한 캔버스(Canvas) 및 WebGL과 비디오(Video) 및 오디오(Audio)가 있으며, 시멘틱(Semantics) 요소인 엘리먼트(Elements), 폼(Forms), 사용자 인터페이스가 있으며, 장치 접근 기술로 웹 브라우저에서 위치 정보를 받기 위한 위치정보(GeoLocation) 지원, 마지막으로 퍼포먼스(Performance) 및 통합(Integration) 기술로 웹 브라우저가 실시간으로 서버와 데이터를 주고받을 수 있는 웹 소켓(Web Socket), 기존 캐시에 대한 제한적인 사용을 개선한 스토리지 기술인 세션 스토리지(Session Storage)와 로컬 스토리지(Local Storage), 멀티스레드(Multi Thread) 웹 앱을 구현 할 수 있는 웹 워커(Web Worker), 다중 파일을 서버로 업로드 할 수 있는 파일 API 등이 있다. 현재 웹 애플리케이션을 위한 W3C의 표준들은 그래픽, 멀티미디어, 장치 적응, 양식, 사용자상호작용, 데이터 스토리지, 개인정보 관리, 센서 및 하드웨어 통합, 네트워크, 통신 및 발견, 패키징, 성능 및 최적 화 등의 12개 범주로 구분 할 수 있으며 [표 2-2]와 같이 34종 이상의 표준들이 10개 이상의 워킹 그룹들을 통해 개발되고 있다 (전종홍과 이승윤, 2012). 이러한 기술들은 데스크톱과 모바일에 탑재된 웹 브라우저 마다 지원이 상이하며, [표 2-3]와 [표 2-4]은 각 각 현재 출시된 데스크톱 웹 브라우저와 모바일 웹 브라우저의 지원율 현황 표이며 이 테스트 결과는 HTML5 지원율을 기기 및 웹브라우저 별로 테스트 할 수 있는 사이트인 The HTML5 Test(<http://html5test.com>)의 결과를 정리한 것이다..

[표 2-2] W3C 웹 애플리케이션 관련 표준 (전종홍과 이승윤, 2012)

기능 분류	관련표준문서 (개발 중 또는 완료)	문서 상태	개발 WG	비고
그래 픽스	SVG HTML Canvas 2D Context CSS Backgrounds and Borders CSS 2D Transforms Module Level 3 CSS Animations Module Level 3 Media Queries WOFF File Format 1.0	IS WD CR WD WD CR LC	SVG WG JYML WG CSS WG CSS WG CSS WG CSS WG Web Fonts WG	TTA 단 체 표 준
멀티 미디어	HTML5 HTML Media Capture HTML Canvas 2D Context	LC WD WD	HTML WG Device APIs & Policy WG HTML WG	
폼	HTML5	LC	HTML WG	
사용자 상호작용	Touch Events Specification (not published yet)	N/ A	Web Events WG	
데이터 스토리지	Web Storage File API Contact API	WD WD WD	Web Application WG Web Application WG Device APIs & Policy WG	
센서 및 HW 처리	Geolocation API The System Information API The Media Capture API	CR WD WD	Geolocation WG Device APIs & Policy WG Device APIs & Policy WG	
네트 워크	XMLHttpRequest XMLHttpRequest Level 2 Cross-Origin Resource Sharing Server-Sent Events The WebSocket API HTML5 onLine DOM state Network Property in The System Information API	CR WD WD WD WD WD WD	Web Application WG Web Application WG Web Application WG Web Application WG Web Application WG HTML WG Device APIs & Policy WG	TTA 단 체 표 준
커뮤니 케이션	The Messaging API HTML5 Web Messaging	WD WD	Device APIs & Policy WG Web Application WG	
패키징	HTML5 Application Cache Widgets Packaging & Configuration Digital Signatures for Widgets Wdiget Access Request Policy	WD LC CR CR	HTML WG Web Application WG Web Application WG Web Application WG	
성능 및 최적화	Navigation Timing Web Workers Mobile Web Application Best Practices	LC WD IS	Web Performance WG Web Application WG MWBP WG	KS/K CLS 표준

[표 2-3] 데스크톱 웹 브라우저 별 HTML5 지원 현황

기 능 \ 브라우저 종류	Internet Explorer 10	Chrome 22	Opera 12	Safari 6.0	Firefox 16
Parsing rules	●	●	●	●	●
Canvas	●	●	●	●	●
Video	▲	▲	▲	▲	▲
Audio	▲	▲	▲	▲	▲
Elements	▲	●	●	●	●
Forms	▲	▲	●	▲	▲
User Interaction	●	●	●	●	●
GeoLocation	●	●	●	●	●
WebGL	▲	●	●	▲	●
WebSocket	●	●	X	●	●
Files	●	●	●	●	●
Storage	▲	●	▲	▲	▲
Web Workers	●	●	●	●	●
● : 지원 ▲ : 일부 지원 X : 미지원					

[표 2-4] 모바일 웹 브라우저 별 HTML5 지원 현황

기 능 \ 브라우저 종류	Android 4.1.1	iOS 6.0	Windows Phone 8	Firefox Mobile 16	Opera Mobile 12.10
Parsing rules	●	●	●	●	●
Canvas	●	●	●	●	●
Video	▲	▲	▲	▲	▲
Audio	▲	▲	▲	▲	●
Elements	●	▲	X	●	●
Forms	●	▲	▲	▲	●
User Interaction	▲	▲	▲	▲	▲
GeoLocation	●	●	●	●	●
WebGL	●	▲	▲	●	●
WebSocket	●	●	●	●	●
Files	●	●	●	●	●
Storage	●	▲	▲	▲	▲
Web Workers	●	●	●	●	●
● : 지원 ▲ : 일부 지원 X : 미지원					

표에서 보이는 것과 같이 데스크톱과 모바일 웹 브라우저의 HTML5의 지원율을 거의 동일하다.

공간 정보 분야에서도 위의 API와 같은 기술들을 사용하여 다양한 서비스를 제공 할 수 있으며, 특히 이 연구에서 사용된 WebGL은 3D 그래픽을 지원하기 때문에 지형공간을 3D로 시각화 할 수 있다. 현재 대부분의 브라우저에서 일부 지원하거나, 전부 지원을 하고 있지만 일부 지원하고 있는 브라우저에서는 아직까지 3D 콘텐츠를 지원하고 있지 않아 현실적으로 사용하기엔 불가능하다. 하지만 웹 브라우저 업체가 직접 HTML5 표준 제정 벤더로 활동하고 있으므로, 향후 모든 브라우저에 WebGL이 지원 될 것으로 예상된다.

최근 데스크톱 외에도 스마트폰, 태블릿 PC, 스마트 TV 등 다양한 해상도를 가진 장치들이 출시되고 있으며, 앞으로도 기존 전자제품에 안드로이드와 같은 운영체제를 탑재한 다양한 스마트 장치들이 출시 될 예정이다. 이러한 다양한 해상도에 웹 앱을 적용하기 위해 HTML5와 함께 CSS3가 발표 되었다. 기존의 CSS2와 큰 차이점으로는 모듈 기반으로 개발 되고 있다는 점이다. CSS3에는 다양한 기능들이 새로 추가되거나 기존의 기술이 개선되어 발표된 표준이다. 그 중에서 미디어 쿼리(Media Query)를 이용하여 다양한 해상도에 맞게 웹 앱을 설계 할 수 있다. 미디어 쿼리는 반응형 웹디자인 이라고 불리고 있으며, W3C의 권장 안 상태에 도달했으며 가장 일반적인 미디어 쿼리 형식은 아래와 같다.

```
@media all and ( media_feature )
```

```
[and ( media_feature ) ] { styles }
```

자료: <http://www.w3.org/TR/css3-mediaqueries/>

CSS3의 미디어 쿼리는 HTML5에서 새롭게 생긴 기술은 아니다. 하지만 기존의 CSS의 미디어 쿼리와는 다르게 미디어의 특징을 잘 표현 할 수 있도록 확장하고 세분화된 스펙을 공개하였다. CSS2의 미디어 타입(Media Type)은 단말기에 따라 각각 다른 스타일 시트를 적용하는 기술 이였지만, 이것만을 가지고 해당 기기의 특성을 제대로 파악 할 수 없으므로 자주 사용되지 않았다. 이를 개선하여 더욱 구체적인 조건에서 필요한 스타일을 적용 할 수 있도록 확장된 표준이다. 이 연구에서 설계 및 구현된 HTML5의 WebGL을 이용하여 3D 지형공간 시스템은 다양한 장치에 적용하기 위해 CSS3의 미디어 쿼리를 적용하였다.

2. WebGL

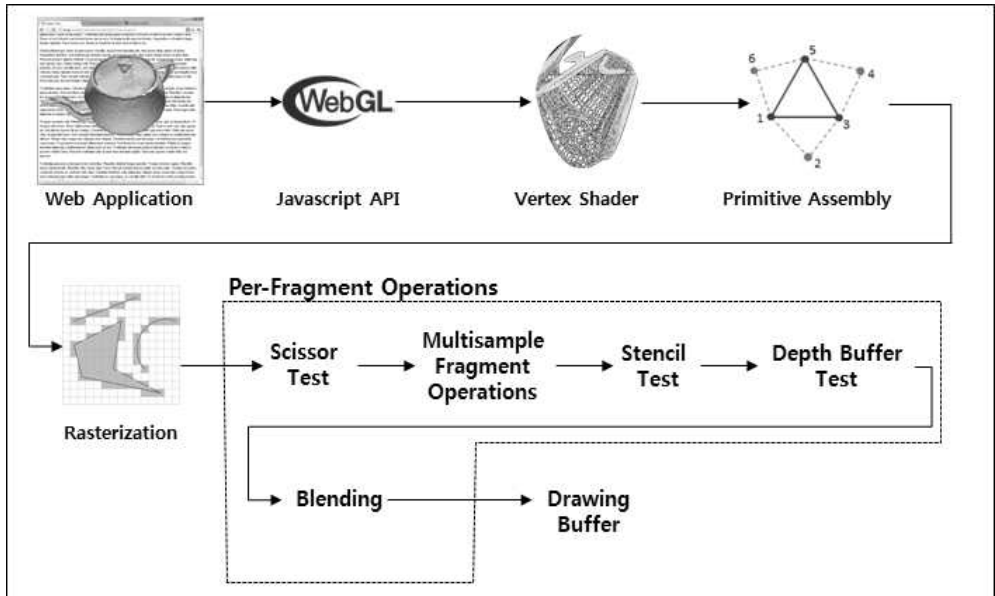
3D 렌더링을 하기 위한 기술로 다양한 기술이 현재 개발되어 사용되고 있다. 데스크톱에서 3D 렌더링을 하기 위한 기술로 다양한 운영체제를 지원하

고 있는 OpenGL과 윈도우 플랫폼에 맞춰져 있는 Direct3D가 현재 많이 사용되고 있다. 모바일 임베디드 3D 렌더링을 위해 개발된 OpenGL ES 2.0은 OpenGL 기반으로 되어 있어 많은 개발자가 이를 이용하여 모바일 상에서 3D 렌더링 관련 연구 및 서비스가 진행되고 있다. 또한 XML 기반 기술로 3D 그래픽을 묘사하기 위한 VRML(Virtual Reality Markup Language)과 X3D(Extensible 3D)가 웹 브라우저에서 사용되고 있다. 하지만 이러한 XML 기반 기술들은 별도로 그래픽 하드웨어 가속을 사용 하는 것이 아니므로 앞에서 언급한 OpenGL이나 Direct3D와는 많은 차이가 있다. 그렇기 때문에 웹 브라우저에서 하드웨어 가속을 이용한 3D 렌더링을 하기 위해선 별도의 플러그인을 설치하여 사용하는 경우가 대부분이다. 어도비(Adobe)의 플래시(Flash)나 마이크로소프트(Microsoft)의 실버라이트(Silverlight)가 대표적인 예이다. 이것들을 사용하면 웹 브라우저에서도 3D 렌더링을 자유롭게 구현하여 서비스가 웹 브라우저 상에서 플러그인을 설치를 해야 되며, 국내에서 많이 사용되고 있는 웹 브라우저인 익스플로러(Explorer)에서는 플러그인 설치가 용이하지만, 이 외에 다른 브라우저에서는 이러한 플러그인 설치를 지향하고 있다.

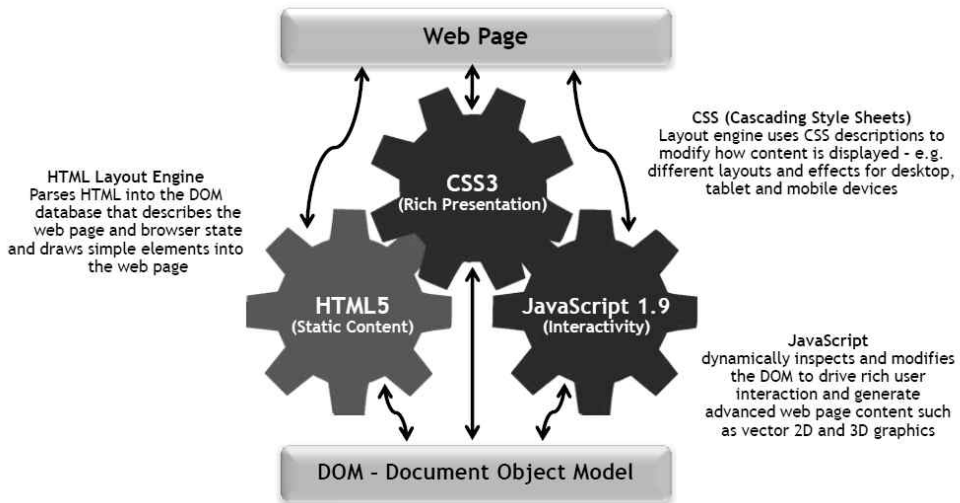
이를 대처하기 위한 기술로 HTML5의 웹 앱을 제작하기 위한 기술 중 WebGL은 웹 브라우저 상에서 인터랙티브한 3D 그래픽을 사용 할 수 있다. WebGL은 비영리 단체인 크로노스 그룹(KHRONOS GROUP)에 의해 관리 되고 있으며, 2011년 3월 1.0 버전을 발표 하였다. OpenGL ES 2.0 기반으로 되어 있어 버텍스 셰이더(Vertex Shader)와 픽셀 셰이더(Pixel Shader)를 통한 GPU(Graphics Processing Unit)가속이 가능하다. WebGL의 그래픽 파이프라인(Pipeline)은 <그림 2-3>에 나타내고 있으며, WebGL이 웹 브라우저의 3개의 메인 컴포넌트에서의 역할을 나타내는 그림은 <그림 2-4>와 같다. WebGL을 장점으로서는 공개된 표준이므로 누구든지 제한 없이 사용 할 수 있다는 점과 그래픽 하드웨어 가속을 통해 3D 렌더링 하므로 속도가 빠르다는 점이다. 기존에 사용하였던 플래시나 실버라이트는 플러그인을 설치해야 하므로 윈도우를 제외한 다른 운영체제를 가진 웹 브라우저나 모바일 웹 브라우저에서는 구동이 불가능하였다. 하지만 HTML5의 WebGL은 웹 브라우저가

지원한다면 플러그인 없이 3D 렌더링이 가능하며, 많은 개발자가 사용을 하고 있는 임베디드 기기의 3D 렌더링 API인 OpenGL ES 2.0 기반으로 되어 있어 이를 사용한 개발자나 OpenGL을 사용한 개발자들은 쉽게 개발 할 수 있다는 장점이 있다. WebGL이 OpenGL보다 편리한 점으로 OpenGL은 메모리를 수동으로 할당하고 해제해야 하는 작업을 하는 작업이 필요하다. 하지만 WebGL은 JavaScript 변수의 유효 범위에 대한 규칙을 따르기 때문에 자동으로 할당하고 해제하므로 코드를 줄이고 단순화하여 이해하기 쉽게 개발을 할 수 있다 (Cantor, 2012).

위에서 언급한 바와 같이 WebGL을 이용하여 3D 렌더링을 하기 위해선 버텍스 셰이더와 픽셀 셰이더 프로그래밍을 작업을 해야 한다. 이러한 작업을 하기 위해선 별도의 프로그래밍 실력이 있어야 하지만, WebGL에서는 선행 작업을 간편하게 해주는 프레임워크들이 많이 공개 되어 있다. 이러한 프레임워크들은 총 30여개가 공개되어 사용되고 있으며, 유틸리티 및 디버그 헬퍼 등이 사용되고 있다. 각 각의 프레임워크들은 다양한 특징을 가지고 있으며 대부분이 3D 게임 제작을 위한 관련 함수들이 편리 하게 되어 있다. [표 2-5]는 WebGL Public Wiki에 등록되어 있는 프레임워크 및 유틸리티 내용을 정리한 것이다.



〈그림 2-3〉 WebGL 그래픽 파이프라인 (Andreas, 2012).



〈그림 2-4〉 HTML5 WebGL의 브라우저에서의 3개의 주요 구성요소 역할 (Trevett, 2011).

[표 2-5] WebGL 프레임워크, 유틸리티, 디버그 헬퍼
(WebGL Public Wiki, 2012)

Name		Features
Frame works	C3DL	Cameras, lights, swappable shader effects, collada model loading etc support.
	Cesium	The best possible performance, precision, visual quality, ease of use, platform support.
	CopperLicht	3D files format support.
	CubicVR.js	XML and BadgerFish-JSON format support.
	GammaJS	Creating simple 2.5D Platform games powered.
	GLGE	Keyframe animation and collada model support.
	GlowScript	Create 3D WebGL animations.
	GTW	The actual UI look'n feel is abstracted from the application using an 'external' JSON formatted 'scene file' and GTW classes can be easily extended to create custom UI elements.
	Inka3D	Export plugin for Autodesk Maya.
	J3D	C#/Unity3d scene exporter.
	Jax	Designed from the very beginning around bringing big ideas to market fast.
	JebGL	Java applet that emulates a WebGL canvas.
	KickJS	Game oriented abstraction of WebGL and makes game programming in a browser easy.
	KriWeb	Flexible playground for experimenting with WebGL.
	Lightgl.js	Easier to quickly prototype WebGL applications
	O3D	Collada model convert, format loading and Scene graph API support.
	Oak3D	Math library, Model library, skeletal animation library and etc support.

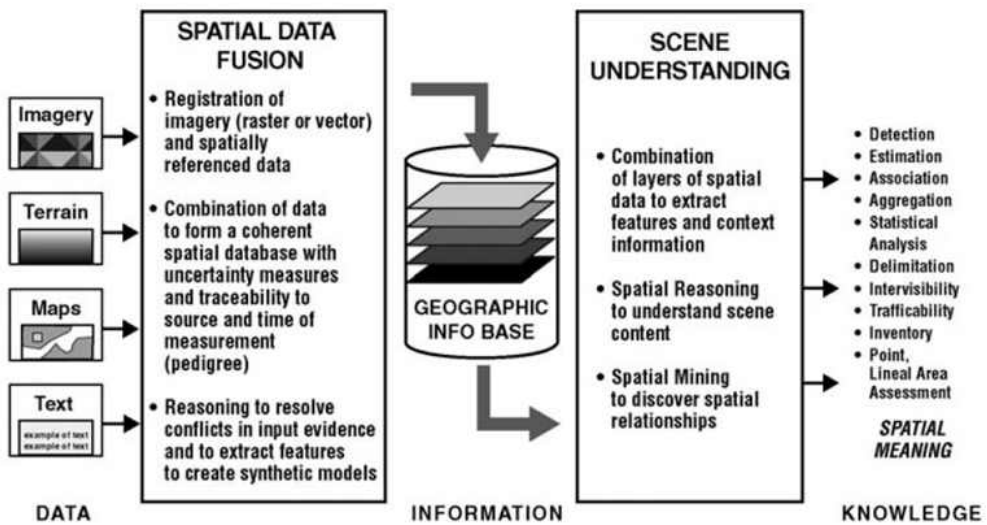
	OpenWeb Globe SDK	Image data, height data, POI, vertex data, 3D object support.
	OSG.JS	The OSGJS api is similar to the OpenSceneGraph api.
	Parallax	Google Web Toolkit 3D library.
	PhiloGL	Data Visualization, Creative Coding and Game Development.
	SceneJS	Scene graph framework
	SpiderGL	Graphic API base on JSON.
	StormEngineC	Provide an easy way to load objects in OBJ format
	TDL	Used to create many of the samples at webglsamples.googlecode.com
	Three.js	Canvas, SVG, WebGL render support.
	WebGL Google Web Toolkit bindings	WGT, gwt-g3d, GwtGL under development.
	Wt WGLWidget	WebGL API to the server-side C++ programmer.
	X3DOM	integrating and manipulating scenes as HTML5-DOM element.
Utilities & Debug Heplers	GLSL Sandbox	Live-editor for fragment shaders
	WebGL playground	An online tool for creating, testing, and sharing WebGL projects.
	WebGLU	Javascript utility library for working with WebGL.
	WebGL Trace	WebGL context wrapper that prints a trace of an application's WebGL calls to the JavaScript console.
	WebGL DebugUtils	WebGLDebugUtils include functions for wrapping every WebGL call with your own callback.
	WebGLUtils	Three functions. (1) properly setup WebGL. (2) handle render loops in the most cross browser and user friendly way.(3) get the current time in a cross browser and 'correct' way.
	gluUnProject	The interface is very similar to the C variant of the function with the same name.

제 3 절 3D 지형공간 정보처리

정보산업 분야에서의 정보처리란 기존의 정보를 가공하여 새로운 정보를 만들어 내는 것이며, 지형공간 정보처리도 마찬가지로 다양한 지형데이터를 가지고 처리하여 새로운 정보를 생성하는 것을 말한다. 과거에서부터 지형공간 정보처리 하는 다양한 알고리즘이 연구되어 왔고, 현재도 새로운 알고리즘이 발표되기도 하며, 이를 적용한 다양한 소프트웨어가 사용 되고 있다. 특히, 2D 지형공간 정보처리 대한 데스크톱 소프트웨어로는 상업적 서비스인 Esri의 ArcGIS, Intergraph의 ERDAS, Exels의 Envi 등이 널리 사용 되고 있으며, 이 외에도 GRASS, Sextante, Orfeo Toolbox, LasTools 등과 같은 오픈소스 소프트웨어들이 많이 존재하고 있다. 스마트폰에서도 2D 지형공간 정보처리에 대한 연구로 강상구와 이기원 (2011)은 오픈 소스를 이용하여 위성영상정보를 이용하여 코너 포인트 객체 추출 모바일 시스템으로 설계 및 구축한 적이 있다. Esri에서는 다양한 공간정보를 처리하기 위한 Javascript API 공개하였고 최근 HTML5 기술까지 적용하여 2D 지형공간 정보처리 웹 앱 제작에 도움을 주고 있다. 하지만 아직까지 웹 브라우저 상의 3D 지형공간 정보처리에 대한 연구 및 서비스는 현재 초기 단계인 3D 시각화 단계 밖에 되고 있지 않다. 전 세계적으로 사용되고 있는 맵 서비스인 구글 맵은 지형공간 데이터의 지형 높이를 보여주는 수준이며, 최근 WebGL을 적용한 노키아 맵은 실질적인 객체 데이터를 시각화 하는 수준이며, 사용되고 있는 영상들은 배경 영상으로 사용하고 있는 정도이다.

정보처리 기술은 영상처리, 의학, 공간정보 등 다양한 분야에서 사용되고 있다. 이 연구에서는 지형공간 기술 중 융합 기술을 적용 하였으며, 융합 관련된 기술도 다양한 알고리즘이 연구되었다. 융합에 사용되고 있는 데이터 유형은 다양하며 <그림 2-5>에서 나타난 것과 같은 과정을 통해 융합되어 활용되고 있다. 융합 기술로 많이 사용되고 있는 알고리즘은 IHS 융합, Brovey 융합, PCA(Principal Component Analysis), MWD(Multi-resolution Wavelet Decomposition), FF(Filter Fusion), HPF(High Pass Filter), LMM(Local Mean Matching), LMVM(Local Mean and Variance

Matching)이 있으며 이러한 알고리즘은 데이터의 그 특성에 맞게 사용해야 한다. 대부분의 영상 융합은 저해상도 칼라 영상을 고해상도 흑백 영상을 이용하여 고해상도 칼라 영상을 생성하는데 사용되고 그 외에도 지표 물들을 분석하여 새로운 주성분을 만들거나 히스토그램의 대조를 향상시키기 위한 목적으로도 사용되고 있다.



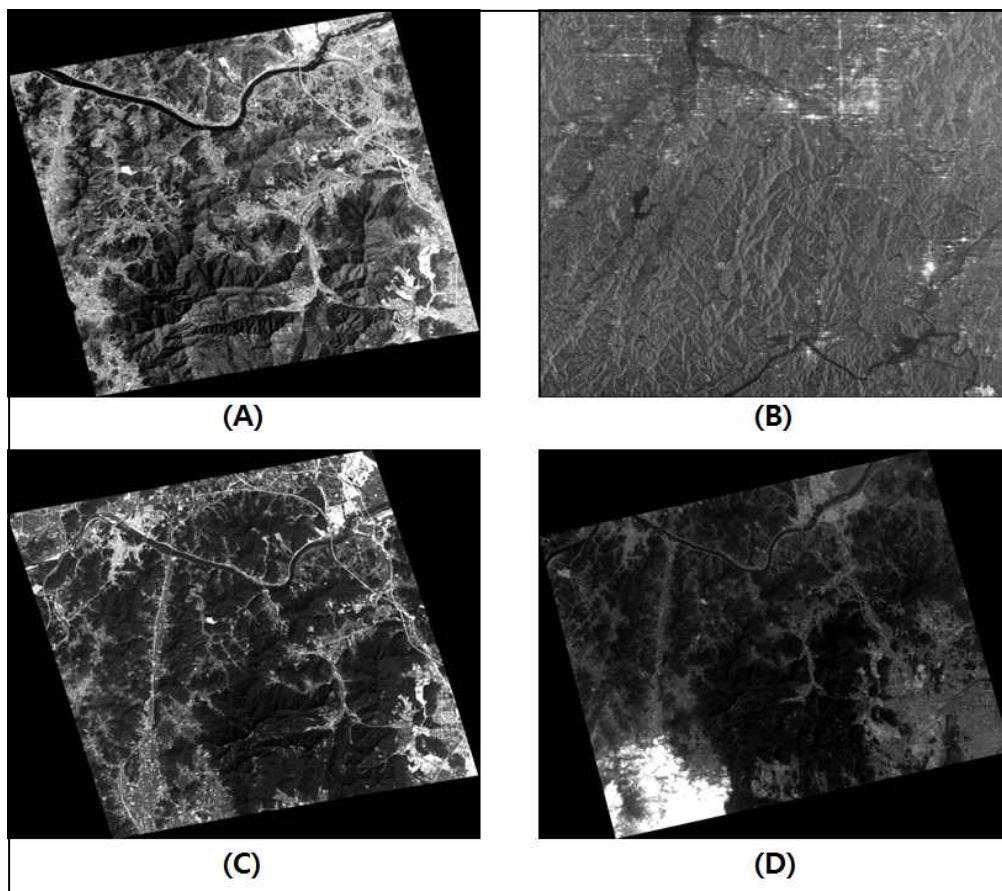
〈그림 2-5〉 다중 유형 공간정보 융합 개념 (Mitchell, 2007).

제 3 장 3D 지형공간 정보처리 시스템 설계 및 구현

제 1 절 지형공간 데이터

공간정보 분야에서는 다양한 지형공간 데이터를 사용하고 있다. 그 중에서 위성영상과 수치지도가 가장 많이 사용되고 있다. 위성영상 데이터는 공간정보 분야에서 지도로 사용하거나 이를 가지고 분석하여 새로운 자료를 만드는 데 가장 많이 사용되고 있으며, 전 세계적으로 고해상도 위성을 만들기 위해 계속적으로 연구가 진행되고 있으며, 수치지도는 기존 종이로 된 도로, 철도, 건물, 하천 등 다양한 인공지물과 자연지형을 디지털로 표현한 지도이며 국내에서는 2004년도 이후 2.0 버전을 사용하고 있다. 이 연구에서는 3D 지형공간 정보처리 시스템을 구현하기 위해 다양한 지형공간 데이터가 사용되었다. 연구에 사용된 데이터로는 위성영상 데이터의 광학 데이터, SAR 데이터, 대기 환경 데이터와 수치지도이다. 광학 데이터는 국가우주개발계획 하에 한국항공우주연구원이 주관하여 우리나라 기술로 만들어진 다목적실용위성 2호(KOMPSAT-2) 영상과 아랍에미리트 정부출현 연구기관인 EIAST(Emirates Institution for Advanced Science and Technology)와 공동개발 사업으로 시작하여 우리나라 기술로 만들어진 두바이샷 1호(DubaiSat-1)를 사용하였다. 광학 데이터는 사람 눈으로 판별이 가능하므로 가장 널리 사용되고 있는 위성데이터이다. 또 다른 위성 데이터인 SAR 데이터는 일본의 ALOS PALSAR를 사용하였다. SAR 데이터는 환경모니터링, 지구자원지도 제작 및 군 관련 시스템에 많이 사용되고 있다. 특히, 레이더 신호로 촬영을 하기 때문에 밤에 촬영하거나 날씨에 영향을 받는 광학 영상의 단점을 많이 보완 할 수 있다. 지형공간 정보처리를 할 경우 광학 영상과 SAR 영상을 융합하여 처리 하는 경우가 많으며, 다양한 융합 처리 방법에 대해서는 앞에 언급한 바 있다. 이 연구 시스템에서는 성질이 다른 두 개의 데이터를 이용하여 웹 브라우저 상에서 3D로 다중 시각화 동시에 실시간 정보처리를 하기 위해 사용 되었다.

〈그림 3-1〉은 이번 연구에 사용되는 KOMPSAT-2 영상, DubaiSat-1 영상, ALOS PALSAR 영상 원본이다.



〈그림 3-1〉 연구에 사용된 광학 및 SAR 원본 데이터: (A) KOMPSAT-2, (B) ALOS PALSAR, (C) 2009년 10월 DubaiSat-1, (D) 2012년 04월 DubaiSat-1.

광학 영상과 SAR영상 외에 이 연구에서 사용된 지형공간 데이터로는 대기 환경 데이터와 수치지도 데이터이다. 사람들은 환경 제해로 인해 대기 환경 수치에 대한 관심이 커지고 있다. 이러한 사항은 국가에서도 인지하여 대기 환경 수치를 실시간으로 제공하려는 노력이 증가하고 있다. 이에 한국 환경 공단(KECO: Korea Environment Corporation)에서는 점차적으로 전국적으

로 측정소를 증설하면서 실시간으로 대기환경정보를 제공하기 위하여 전국 실시간 대기 오염도 공개 홈페이지(<http://www.airkorea.or.kr>)에서 수집된 환경 정보를 실시간으로 제공하기 시작했다. 방사능 정보 또한 한국원자력안전기술원(KINS: Korea Institute of Nuclear Safety)에서 실시간으로 제공해주고 있다. 대기 환경 측정소는 2012년 2월 기준으로 전국 97개 시,군의 304개의 도시대기, 도로변대기, 국가배경, 교외대기 측정망에서 대기환경기준물질 (Atmosphere Environmental Standard) 다섯 개 항목인 아황산가스(SO₂), 일산화탄소(CO), 이산화질소(NO₂), 미세먼지(PM-10), 오존(O₃)을 측정하고 있다. 이 연구에서는 측정소의 환경 데이터들을 이용하여 그리드 데이터를 생성하였다. 측정소가 가장 많이 분포 되어 있는 지역은 서울특별시와 부산광역시였으며 연구 지역인 세종시 인근 지역 측정소를 확인한 결과 총 여덟 개의 측정소가 검색이 되었고, 이 연구에서는 여덟 개의 측정소의 위치를 기반으로 하여 설계하였다. <그림 3-2>는 구글 어스를 이용하여 측정소의 위치와 연구 지역을 나타낸 것이다.

마지막으로 이 연구에서 사용된 수치지도는 축적 1:250,000의 총 여섯 장을 사용 하였다. 수치지도는 광학 영상과 SAR 영상에 대해 위치를 맞추는 작업인 지상 제어점(GCP: Ground Control point) 작업과 3D 시각화를 위한 DEM(Digital Elevation Models) 생성을 위해 사용되었다. 지상 제어점 일치 작업은 2개 이상의 위성 데이터를 사용 할 경우 꼭 필요한 작업으로 수집한 영상 데이터들은 위성의 위치와 자세의 변화, 위성의 속도 변화, 주사범위와 주사경의 회전속도 변화, 지구자전, 지도투영법 등의 원인으로 왜곡 된 부분을 보정 해주는 작업이다. 수치지도는 국토지리원의 국가공간정보유통시스템(<http://www.nsic.go.kr/ndsi>)에서 구매 할 수 있으며, <그림 3-3>은 이 연구에 사용된 수치지도를 국가공간정보유통시스템을 통해 구매한 수치지도를 나타낸 것이다.

제 2 절 3D 지형공간 전처리

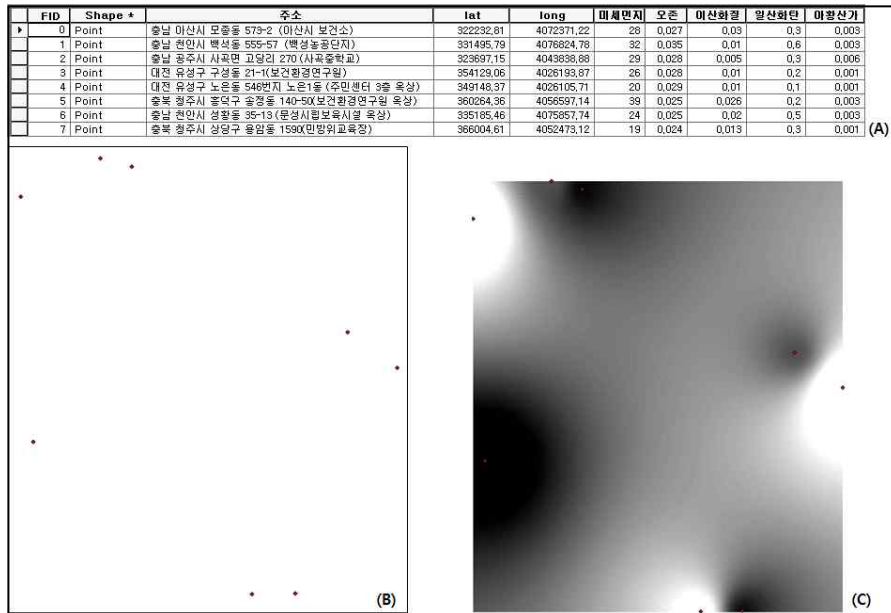
3D 지형공간 정보처리 시스템에 지형공간 데이터가 사용되기 위해서는 전처리 과정이 필요하다. 시스템을 구현하기 위해 수집된 자료들은 단일로 사용할 경우 바로 사용이 가능한 데이터들이다. 하지만 2개 이상의 데이터를 동시에 사용할 경우 전처리 과정이 필요하다. 첫 번째로 광학 영상 데이터와 SAR 영상 데이터는 서로의 위치가 일치 하지 않기 때문에 지상 제어점 일치 작업이 필요하며, 두 번째로는 측정소 기준으로 획득한 데이터를 이용하여 그리드 데이터 생성해야 한다. 마지막으로 3D로 보여주기 위해 DEM을 미리 생성해야 한다.

첫 번째 과정인 광학 영상과 SAR 영상의 지상 제어점 일치 작업은 상업 소프트웨어를 사용하여 수치지도를 기준으로 보정 작업을 시행하였다. 지상 제어점 일치하는 두개 이상의 위성영상을 동시에 사용할 경우 꼭 필요한 작업이다. 위성 영상이 찍힌 위성의 종류, 시간, 조건 등이 매번 바뀌므로 같은 장소를 찍더라도 매번 다른 왜곡이 생기게 된다.

두 번째 과정은 환경 센서 측정소 여덟 개를 기준으로 그리드 데이터를 생성 하는 과정이다. 환경 센서 측정소 위치는 한국 환경 공단의 에어코리아 사이트에서 주소 형식으로 받을 수 있으며, 주소 검색을 통해 좌표를 얻어 올 수 있다. 좌표를 가지고 포인트 데이터를 Shape 파일로 생성한다. 생성 할 때 실제 좌표 여덟 개를 포인트 데이터로 입력하고 측정이 되는 환경 데이터인 미세먼지, 오존, 이산화질소, 일산화탄소, 아황산가스의 데이터도 모두 속성 테이블에 입력한다. 그리드 데이터를 만들기 위해서는 여덟 개의 지점을 이용하여 보간법(Interpolation)을 수행해야 한다. 보간법이란 기존에 알고 있는 특정 지점이나 지역의 속성 값을 이용하여 알려지지 않은 지점이나 지역의 속성 값을 생성 하는 것이다. 이 연구에서는 다양한 보간법 중 IDW(Inverse Distance Weighted) 기법을 사용하였다. IDW는 거리의 역에 대한 가중치를 계산하는 것으로 포인트 자료가 밀집 되어 있을 때 가장 적합한 알고리즘이다(Childs, 2004). 이 연구 지역에 대해서는 포인트 자료가 밀집되어 있지 않아 정확한 그리드 데이터를 생성 할 수 없지만 현재 가장 많이 사용되는 보

간법이며, 다른 알고리즘을 적용하여도 차이가 없으므로 이 연구에서 사용되었다. 그리드 데이터를 생성 할 때 환경 측정 기준을 따라 좋음, 일반, 나쁨 기준으로 [표 3-1]과 같이 색깔을 정하여서 생성하였다. <그림 3-4>은 환경 그리드 데이터를 생성하는 과정 및 결과이다. 이 연구에서는 실제 환경 데이터를 사용하지 않았다.

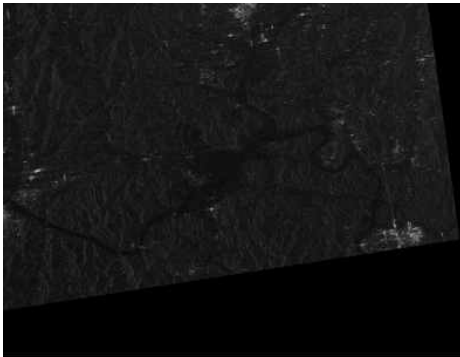
이렇게 처리된 광학 영상, SAR 영상, 환경 그리드 데이터, DEM 데이터모두가 겹치는 지역에 대하여 자르는 작업을 수행하였다. <그림 3-5>는 이번 연구에서 사용된 KOMPSAT-2, ALOS PALSAR, 임의 환경 그리드 데이터와 3D로 보여 지기 위한 DEM 데이터이며, <그림 3-6>는 블렌딩 기법을 이용하여 3D 시각화 및 지역 변화를 보기 위해 같은 지역의 다른 시간 때의 Dubaisat-1의 2009년 촬영된 데이터와 2012년 촬영된 데이터 그리고 Dubaisat-1 3D 시각화를 위한 DEM 데이터이다.



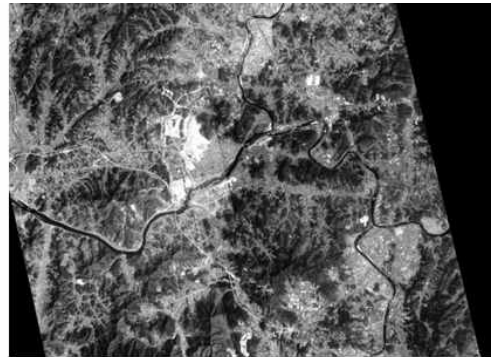
<그림 3-4> 측정소 정보 및 IDW 그리드 데이터: (A) 8개의 측정소 Shape 파일 속성 정보, (B) 8개의 측정소 실제 좌표를 이용한 포인트, (C) IDW 기법을 이용한 그리드 데이터 생성.

[표 3-1] 에어 코리아 대기 환경 기준

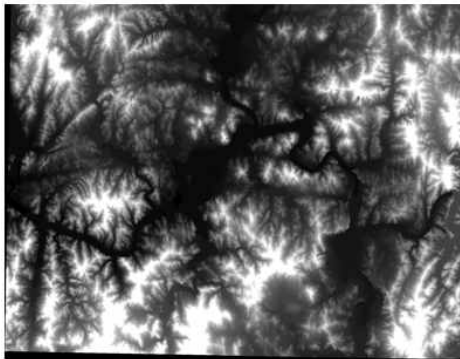
구분(단위)	좋음	일반	나쁨
미세먼지($\mu g/m^2$)	0~31	31~80	81~120
오존(ppm)	0~0.04	0.041~0.08	0.081~0.12
이산화질소(ppm)	0~0.030	0.031~0.060	0.061~0.15
일산화탄소(ppm)	0~2	2.01~9	9.01~12
아황산가스(ppm)	0~0.02	0.021~0.05	0.051~0.1



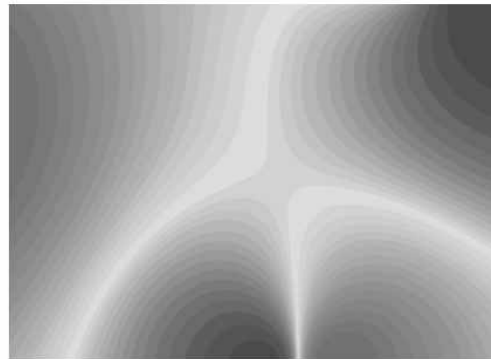
(A)



(B)

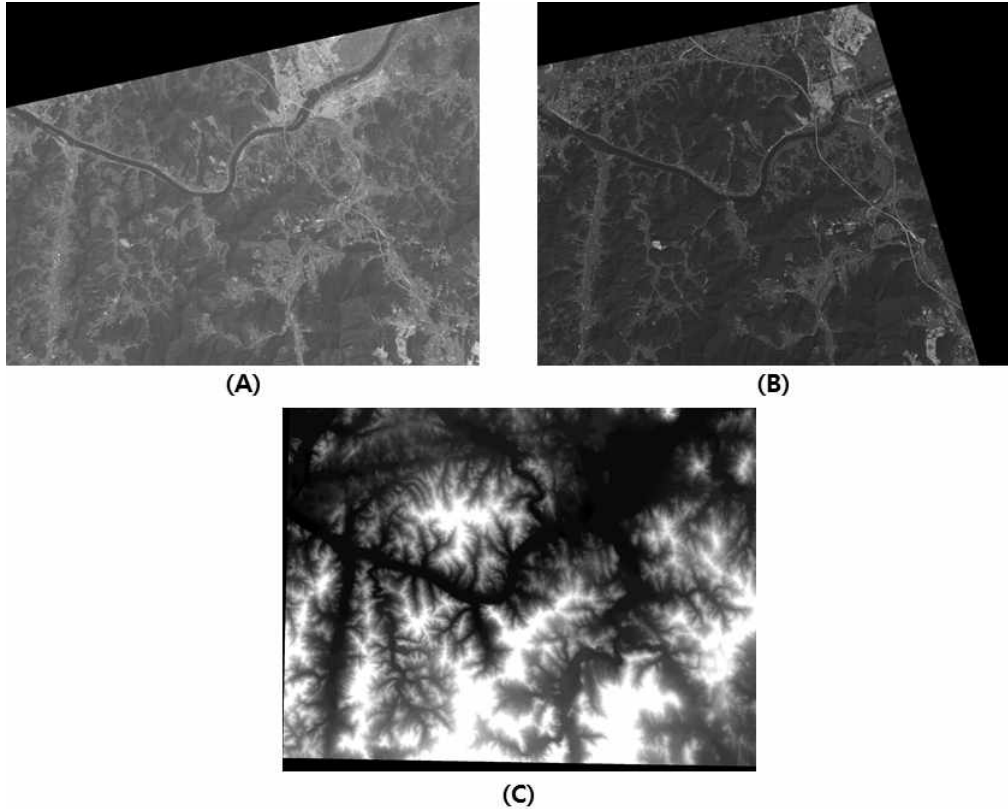


(C)



(D)

〈그림 3-5〉 3D 실시간 정보처리 시스템에 사용된 광학 데이터 및 SAR 데이터: (A) ALOS PALSAR 데이터, (B) KOMPSAT-2 데이터, (C) DEM 데이터, (D) 임의 환경 그리드 데이터.



〈그림 3-6〉 블렌딩을 이용한 시간에 따른 3D 시각화에 사용된 광학 데이터:
 (A) Dubaisat-1 2009년 10월 08일, (B) Dubaisat-1 2012년 04월 17일, (C)
 DEM 데이터.

제 3 절 3D 지형공간 정보처리 적용기술

이 연구에서 구현된 3D 지형공간 정보처리 시스템은 다양한 기술이 적용되어 있다. 시스템은 웹 브라우저 상에서 구동이 되는 웹 앱 형태를 가지고 있으므로 서버와 클라이언트로 나뉘지며 둘 사이에 통신을 하기 위해서는 Ajax 통신 기술이 필요하다. 별도의 라이브러리 없이 자바스크립트를 통해서 서버와 통신을 할 수 있지만 Ajax 통신을 편리하게 해주는 다양한 라이브러리가 존재하고 있다. 이 연구에서는 다양한 라이브러리 중에서 jQuery를 통해 서버와 통신을 하고 그 외에 사용자 인터페이스 구성하는데 사용되었다.

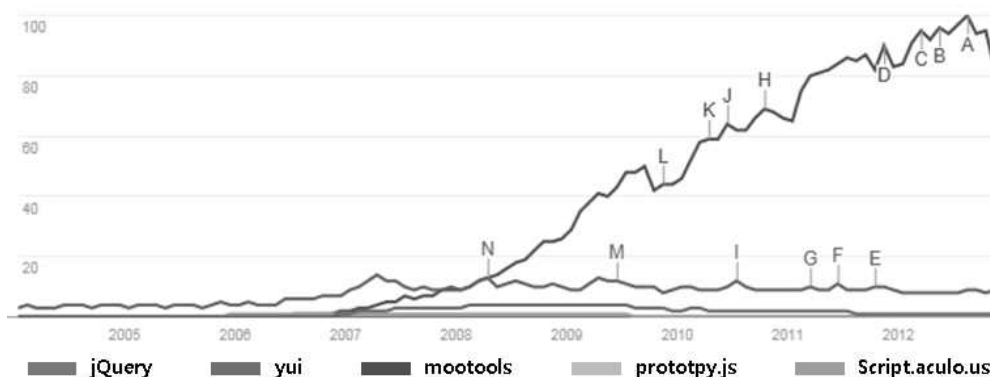
클라이언트에서는 웹 브라우저에서 3D 지형공간을 시각화 하는 기술을 적용하였다. 웹 브라우저에서 3D 지형을 시각화 할 수 있는 플래시 및 실버라이트 등 다양한 기술이 있지만 이 연구에서는 별도의 플러그인 필요 없는 HTML5의 WebGL을 사용하였다. WebGL 중에서도 3D 지형공간을 쉽게 시각화하기 위해 CubicVR.js 프레임워크를 이용하여 구축하였다. 여러 이미지를 투명 효과를 적용하여 다중 시각화가 가능한 블렌딩 효과를 적용하였다. WebGL을 이용하면 블렌딩 효과를 적용 할 수 있으며, 3D 시각화 동시에 블렌딩 효과를 적용함으로써 같은 지역의 시간 차이가 있는 영상에 대해 지역 변화를 볼 수 있다. 마지막으로 서버에서는 실시간 지형공간 정보처리의 다양한 기술 중 융합 기술인 IHS 기법을 이용하여 광학 데이터와 SAR 데이터를 융합 기술을 적용 하였다. IHS 기법은 융합 기술 중 계산식이 간단한 알고리즘이며, 현재까지도 많이 사용되고 있는 기법이다. 광학 데이터와 SAR 데이터를 융합함으로써 새로운 지형공간 데이터가 생성이 되고 생성된 데이터는 광학 데이터와 SAR 데이터가 각각 시각화 되었을 때 보여주는 자료와 또 다른 데이터를 생성 하므로 새로운 시각에서 볼 수 있다.

1. jQuery Ajax

일반적으로 웹 앱을 구축 할 때 사용되는 언어는 HTML, Javascript, CSS 이다. 다른 언어와 다르게 웹 언어는 기본적으로 제공되는 템플릿이 없으며, 개발자가 직접 모든 작업을 해야 한다. 하지만 최근 모바일 웹 앱에 대한 서비스가 증가 하면서 기존의 웹 방식이 아닌 특유의 모바일 사용자 인터페이스를 구성 할 수 있는 라이브러리가 출시되고 있다. 대표적인 라이브러리로 jQuery, jQTouch, Sencha Touch 등이 있으며 이러한 라이브러리는 복잡한 작업을 편리하게 해준다.

특히 웹 앱에서 서버와 통신을 하기 위한 수단인 Ajax 통신 기술은 중요한 기술이다. 다양한 자바스크립트 라이브러리에서는 Ajax 통신을 간편하게 사용할 수 있도록 지원 해주고 있으며 대표적으로 jQuery, script.aculo.us, prototype.js, yui, mootools 가 있다. 이 연구에서는 jQuery를 사용하였으며,

〈그림 3-7〉는 구글 경향 분석 앱에서 대표적인 라이브러리 사용량을 통계 낸 자료로 jQuery가 세계적으로 많이 사용되고 있는 자바스크립트 라이브러리라는 것을 볼 수 있다.



〈그림 3-7〉 Google Trends 자바스크립트 라이브러리 사용률(2005~2012).

jQuery는 클라이언트 스크립트 언어를 단순화 할 수 있도록 설계되었고 DOM(Document Object Model) 엘리먼트(Element) 선택, DOM 트래버설 (Tracersal) 및 수정, CSS 조작, 특수효과 및 애니메이션, Ajax 등 다양한 기능을 가지고 있다. 최근에는 모바일에 최적화된 jQuery Mobile이 등장하면서 모바일 웹 앱 구축 시 더욱 더 편리 해졌다. 하지만 이 연구에서는 jQuery Mobile은 사용하지 않았다. jQuery Mobile은 모바일 기기에 특성화되었기 때문에 이 연구에서 목표로 하고 있는 데스크톱을 포함한 모든 플랫폼에서 구동을 하기 위한 시스템으로는 적합하지 않기 때문에 jQuery Mobile은 사용하지 않았다.

Ajax는 웹 서버와 비동기적으로 XML 데이터를 교환하고 조작하기 위한 기술이다. Ajax를 사용하게 되면 페이지 이동 없이 화면 전환이 가능하며, 서버 처리를 기다리지 않고 비동기 요청이 가능하며, 또한 별도의 플러그인 설치를 할 필요가 없다. 자바스크립트 라이브러리를 사용하지 않고 Ajax 통신을 하게 되면 브라우저 별 사용 방법이 다르기 때문에 고려할 사항들이 많다. 하지만, jQuery 같은 라이브러리를 사용하게 되면 이를 내부에서 처리하기

때문에 별도의 작업 없이도 다양한 브라우저에서 통신이 가능하다. Ajax 서버 통신 방식은 다음과 같다. 사용자가 이벤트를 발생하게 되면 XMLHttpRequest 객체를 통해서 웹 서버에 요청을 전달하게 된다. 웹 서버에서는 요청을 알맞게 처리 후 결과를 XML이나 단순 텍스트를 생성해서 다시 XMLHttpRequest로 전송을 하게 되고 클라이언트에서는 응답 데이터와 DOM을 이용해 사용자 화면에 반영하게 된다.

2. WebGL CubicVR.js

웹 브라우저에서 WebGL을 사용하기 위해서는 기본적으로 버텍스 셰이더와 픽셀 셰이더 언어인 GLSL(OpenGL Shading Language)를 이용하여 프로그래밍을 해야 한다. 이러한 프로그래밍을 하기 위해서는 기본적으로 OpenGL ES 2.0에 대한 기본 지식이 있어야하지만 이를 편리하게 해주는 다양한 프레임워크들이 출시되었고 위의 표에서 언급한 바 있다. 이 연구에서는 WebGL 프레임워크 중 CubicVR.js를 사용하였으며 CubicVR.js는 CubicVR 3D engine에서 개발되었고 현재 관리되고 있는 프레임워크이다. CubicVR 3D engine은 임베디드 기계에 그래픽을 적용하여 게임을 만드는 회사이며 아이폰 게임 앱인 PocketHoops 이라는 농구 게임으로 시작된 프로젝트이다. 기존에는 C++로 되어 있는 OpenGL ES을 이용하여 아이폰 뿐만 아니라 PSP(Play Station Portable)에 3D 그래픽 게임을 구현하였다. 이러한 게임 앱을 제공하다가 CubicVR 3D engine은 개발자들이 쉽게 3D 그래픽을 사용하여 앱을 만들 수 있도록 CubicVR 이라는 API를 제공하였고, WebGL이 발표된 이후 2010년 5월에 기존 CubicVR을 WebGL 버전인 CubicVR.js으로 포팅(Porting) 하여 제공하기 시작하였다. 이 연구에서 CubicVR.js를 사용하게 된 이유는 다음과 같다. 구현된 시스템은 스마트 기기에도 구동이 빠르게 작동 되어야 한다. 기본적으로 다른 프레임워크보다 자바스크립트는 파일 용량이 작을수록 속도가 빠르다. CubicVR.js는 기본적으로 자바스크립트 파일이 가볍게 구성되어 있으며 이 연구 시스템에 필요한 모든 기능 들이 포함 되어 있다. 또한 위에서 언급한 WebGL 프레임워크들 중 가장 이해하기 쉽고 간

편하게 문서가 정리 되어 있으며, 사용하기 편리하게 되어 있다. [표 3-2]는 정리 되어 있는 API 중 이 연구의 시스템에 사용된 중요 API 들을 정리 한 내용이다.

[표 3-2] CubicVR.js API 문서(CubicVR 3D engine, 2012)

API Name	Object Constructor	contents
Core classes and objects		
GLCore	init	Used to initialize CubicVR to a new Canvas3D/WebGL context.
MainLoop	MainLoop	Provides a simple way to wrap your main drawing loop.
MouseView Controller	MouseView Controller	Preates a simple orbital view controller for a canvas and a Camera.
Utility and Math libraries		
vec2	–	Object provides tools for working with 2D Vectors.
vec3	–	Object provides tools for working with 3D Vectors or Points.
mat4	–	Object provides tools for working with 4x4 Matrices.
util	–	Object provides extra utilities for common tasks.
Mesh and Material classes		
Material	opacity	Opacity, default: 1.0.
	name	Name of this material, used for retrieving materials by name.
	textures	Texture configuration object.
Texture	Texture	Provides a container and loader for images.

UVMapper	projectionMode	U,V Projection Mode, see enums.
	projectionAxis	U,V Projection Axis, see enums.
	scale	U,V projection scale.
	rotation	U,V projection rotation.
Scene-related classes		
Camera	fov	Field of View, specified in degrees.
	nearclip	Near clipping plane, pixels closer than this will be discarded.
	farclip	Far clipping plane, pixels further than this will be discarded.
Landscape	size	The total length of the axis with the largest number of divisions.
	division_x	Total divisions on the X-axis.
	divisions_z	Total divisions on the Z-axis.
	Mat	Material to apply.
Light	type	Light type, see setType() for options.
	position	Light position in the world.
	distance	Light direction (vector) for directional lights.
SceneObject	mesh	A [[Mesh]].
	position	Position in [[Scene]] or relative to parent.
	name	A [[Motion]] for this SceneObject.

3. 블렌딩

블렌딩(Blending)은 컴퓨터 그래픽스 기술에서 나온 기법이며, 여러 개의 이미지에 투명 효과를 적용 하는 기술이다. WebGL에서도 블렌딩 효과를 주기 위한 API가 적용 되어 있으며, OpenGL ES 와 비슷한 방법으로 사용 할

수 있도록 되어 있다. 기본적인 블렌딩 함수식은 다음과 같다.

$$ColorOutput = S \times sW + D \times dW$$

S: source color, D: destination color,

sW: source scaling factor, dW: destination scaling factor

자료 : D. Cantor and Brandon J., WebGL Beginners Guide

블렌딩 효과 중 알파 블렌딩(Alpha blending)은 렌더링 과정에서 유리와 같은 매질을 처리하기 위하여 투명도를 지정하는 것이다. 일반적으로 알려진 알파 블렌딩 방법은 이미 그려진 배경화면의 투명도 값에 대하여 처리하는 알파 플랜(Alpha Plan) 방식과 새로 그릴 텍스처가 입혀진 폴리곤을 이용하여 투명도를 처리하는 알파 텍스처(Alpha Texture) 방식이다.

WebGL에서는 가산형 블렌딩(Additive blending), 감산형 블렌딩(Subtractive blending), 곱셈형 블렌딩(Multiplicative blending), 보간형 블렌딩(Interpolative blending)의 알파 블렌딩 기법을 제공 해주고 있다. 이 외에도 다양한 블렌딩 기법이 사용되고 있지만 앞의 네 가지 기법이 대표적으로 많이 사용되는 블렌딩 기법이다. 이러한 블렌딩 기법은 말 그대로 가산형 블렌딩 기법은 소스(Source) 값과 데스티네이션(Destination) 값을 더한 값이며, 감산형은 감한 값, 곱셈형은 곱한 값을 가지고 블렌딩 하는 기법이다. 보간형 블렌딩 기법은 가산형 블렌딩 기법과 감산형 블렌딩 기법이 동시에 적용된 기법으로 현재 블렌딩 기법 중 가장 많이 사용되고 있는 기법 중 하나 이다. [표 3-3]은 네 가지 블렌딩 기법의 식과 WebGL에서 기법 사용 시 적용되는 함수를 정리한 표이다.

시스템에 사용된 WebGL 프레임워크 CubicVR.js에는 기본적으로 보간형 블렌딩 기법이 적용 되어 있다. 하지만 각 각의 블렌딩 효과에 따라 특징이 있으므로 사용자가 직접 블렌딩 기법을 변경하여 적용 하게 되면 다양한 변화를 확인 할 수 있다. 이 연구에서는 CubicVR.js에서 이미 정해진 보간형 블렌딩 기법 외에도 다른 블렌딩 기법을 직접 사용자가 선택 할 수 있도록 사용자 인터페이스를 구성하였다.

[표 3-3] 블렌딩 기법 별 WebGL 함수 및 식

블렌딩 기 법	WebGL 함수	함수 식
가산형 블렌딩	gl.blendFunc(gl.ONE, gl.ONE)	Color output = $S * 1 + D * 1$
감산형 블렌딩	gl.blendFunc(gl.FUNC_SUBTRACT) gl.blendFunc(gl.ONE, gl.ONE)	Color output = $S * 1 - D * 1$
곱셈형 블렌딩	gl.blendFunc(gl.DST_COLOR, gl.ZERO)	Color output = $S * D + D * (0)$
보간형 블렌딩	gl.blendFunc(gl.SRC_ALPHA, gl.ONE_MINUS_SRC_ALPHA)	Color output = $S * S.a + D * (1 - S.a)$

4. IHS 융합

공간 정보 분야의 이미지 융합은 중요한 기술이다. 분석에 사용되는 영상들은 단일 영상을 사용하기 보단 다양한 영상을 융합하여 사용하게 되면 새로운 정보가 보여 지게 된다. 그렇기 때문에 지형공간을 가지고 이미지 융합을 사용하여 정보처리를 하는 어플리케이션이나 알고리즘 분석하는 연구 사례도 많이 진행되었고 현재까지도 진행 되고 있다 (Bretschneider, 2000, Canga, 2002, Stathaki, 2008; Klonus, 2009).

이 연구에서는 이러한 다양한 알고리즘 중에서 간단한 계산식으로 융합이 가능한 IHS 융합을 적용하였다. IHS 융합은 기존 컴퓨터에서 보여 지고 있는 색채로 많이 사용되고 있는 빨강(Red), 초록(Green), 파랑(Blue)의 조합인 RGB 색채와는 다르게 강도(Intensity), 색도(Hue), 채도(Saturation)의 색체계인 IHS 색채를 이용하는 알고리즘이다. IHS 색채의 명암영상에는 지형공간의 공간해상도 정보가 포함되어 있고, 색조영상과 채도 영상에는 분광해상도 정보가 포함되어 있다. 이러한 정보를 이용하여 기존의 RGB 색채로 되어 있는 영상을 IHS로 변환 후 강도 부분을 단일 영상으로 치환한다. 그런 후 다시 RGB 색채로 역 변환을 하게 되면 IHS 융합이 완료가 된다. IHS 융합은 세 개의 밴드 영상에만 적용 할 수 있다는 단점이 있지만, 대부분의 광학 영상은 RGB 색채로 되어 있으며, 만약 더 많은 밴드 영상에 적용을 할 경우 Brovey 영상 융합을 사용 하면 되지만 이 알고리즘의 경우 영상의 히스토그램에서

아랫부분과 윗부분의 대조를 향상시키기 위한 목적으로 주로 사용되고 있다. 이 연구에서 사용되는 영상은 밴드는 RGB 색채를 가지므로 IHS 융합을 적용시키기에 적합하다. 특히, 스마트 기기에서의 지형정보 정보처리에 대한 연구는 아직 초기 단계이므로 이 연구에서는 계산식이 간단한 알고리즘을 채택하여 가능성을 검토하고자 하였다. 아래의 식은 IHS 융합에 대한 RGB 색채를 IHS 색채로 변환하는 식과 IHS 색채를 RGB로 역 변환하는 식을 나타낸 것이다.

$0 \leq R \leq 1$	RGB to IHS
$0 \leq G \leq 1$	
$0 \leq B \leq 1$	
	$I = R + G + B$
	$H = \begin{cases} (G - B)/(I - 3B) & \text{if } B = \min(R, G, B) \\ (B - R)/(I - 3R) + 1 & \text{if } R = \min(R, G, B) \\ (R - G)/(I - 3G) + 2 & \text{if } G = \min(R, G, B) \end{cases}$
	$S = \begin{cases} (I - 3B)/I & \text{if } 0 \leq H \leq 1 \\ (I - 3R)/I & \text{if } 1 \leq H \leq 2 \\ (I - 3G)/I & \text{if } 2 \leq H \leq 3 \end{cases}$
$0 \leq I \leq 3$	IHS to RGB
$0 \leq H \leq 3$	
$0 \leq S \leq 1$	
	$R = \begin{cases} I(1 + 2S - 3SH)/3 & \text{if } 0 \leq H \leq 1 \\ I(1 - S)/3 & \text{if } 1 \leq H \leq 2 \\ I(1 - S + 3S(H - 2))/3 & \text{if } 2 \leq H \leq 3 \end{cases}$
	$G = \begin{cases} I(1 - S + 3SH)/3 & \text{if } 0 \leq H \leq 1 \\ I(1 + 2S - 3S(H - 1))/3 & \text{if } 1 \leq H \leq 2 \\ I(1 - S)/3 & \text{if } 2 \leq H \leq 3 \end{cases}$
	$B = \begin{cases} I(1 - S)/3 & \text{if } 0 \leq H \leq 1 \\ I(1 - S + 3S(H - 1))/3 & \text{if } 1 \leq H \leq 2 \\ I(1 + 2S - 3S(H - 2))/3 & \text{if } 2 \leq H \leq 3 \end{cases}$
	자료 : Open Geospatial Fundamentals

제 4 절 3D 지형공간 정보처리 시스템 설계

이 연구에서 설계 및 구축한 3D 지형공간 정보처리 시스템은 플랫폼에 독립적으로 구동 되도록 구현되었다. 특히 최근 데스크톱뿐만 아니라 다양한 스

마트기기들이 출시되고 보급화 되면서 이를 이용하여 업무에 활용하기 위한 다양한 연구 및 서비스가 출시되고 있으며, 특히 HTML5의 발표로 인해 이를 이용한 스마트 기기에서의 웹 앱에 대한 활용도가 증가되고 있다. 이 연구의 시스템은 장치의 운영체제와는 독립적으로 장치에 설치된 웹 브라우저가 HTML5를 어느 정도 지원하는지에 따라서 구동 여부를 판별 할 수 있다. iOS 기반 스마트폰을 제외하고는 기존 탑재된 웹 브라우저 외에도 다양한 웹 브라우저를 설치하여 사용 가능하므로 구축된 시스템은 대부분의 플랫폼에서 사실상 구동이 가능하다. 기존의 iOS는 애플에서 직접 개발한 사파리(Safari) 웹 브라우저만 사용 할 수 있도록 하였고 최근에 구글 크롬(Chrome)까지 허용하였지만 그 외의 다른 웹 브라우저 활용에 대한 계획은 아직까지 없다.

또한 HTML5가 발표됨과 동시에 발표된 CSS3의 미디어 쿼리를 적용함으로써 다양한 장치 해상도에 맞게 사용하기 편리 하도록 구성되어 있다. 미디어 쿼리를 통해 사용자가 웹 앱을 구동하였을 때 해상도 및 브라우저 종류를 파악하게 되고 이에 맞춰 사용자 인터페이스가 보기 쉽게 자동으로 보여 진다. 지형공간의 전처리 과정을 제외한 모든 과정은 사용자가 사용하기 쉽도록 옵션을 최소화 하였으며 모든 과정을 자동화 하였다. 웹 시스템 환경의 서버 사이드 및 클라이언트 사이드에 사용된 소프트웨어 및 라이브러리 등은 모두 오픈 소스를 사용하여 모듈화 하였다. 오픈 소스를 사용하여 모듈화 하였으므로 향후 다른 시스템에도 쉽게 사용 할 수 있다.

1. 오픈 소스 적용 시스템 환경

이 연구에서 구축된 3D 지형공간 시스템은 오픈 소스로 구성되어 있다. 웹 기반 시스템이므로 서버 사이드와 클라이언트 사이드로 나뉘져 있다. 서버 사이트에 사용된 웹 서버는 오픈 라이선스인 리눅스 페도라 운영체제의 아파치 웹 서버를 사용하였고 지형공간 데이터를 다루기 위한 GDAL 라이브러리를 사용하였다. GDAL라이브러리는 X/MIT 라이선스이므로 사용제한 없이 사용할 수 있으며, 커멘드 라인을 이용하여 다양한 지형공간 데이터를 변환하거나 프로세싱을 할 수 있는 라이브러리이다. 이 연구에서 GDAL 라이브러리를 사

용한 이유는 대부분 위성 데이터는 GeoTiff로 되어 있으며, 이 파일은 웹 브라우저 상에서 보여주는 것이 어렵다. 그렇기 때문에 GDAL 라이브러리를 사용하여 GeoTiff 파일을 이미지 파일로 변환하여 웹 브라우저 상에서 보이도록 변환하는데 사용하였다.

또한 실시간 정보처리를 위해 Python 스크립트를 이용하였다. Python 스크립트는 플랫폼에 독립적이며 인터프리터식(Interpreter), 객체지향적(Object-Oriented), 동적 타이핑(Dynamically typed) 대화형 언어이다. Python은 기존 다른 언어 뿐 만 아니라, 웹 언어와도 호환성이 강하므로 이 연구에 실시간 처리로 적합하다. 또한 Python에서 제공하고 있는 GDAL 라이브러리를 플러그인하여 GeoTiff 파일을 읽고 쓸 수 있으므로 지형공간 데이터를 다룰 수 있다. 클라이언트 사이드에서는 기본적으로 웹 언어 구성인 HTML5, Javascript, CSS3를 사용하였으며 서버와 통신을 하기 위해 jQuery를 사용하였다. 또한 3D 시각화를 위해 WebGL의 CubicVR.js를 사용하였다. [표 3-4]는 이 연구에서 사용된 오픈 소스 리스트를 나타낸 표이다.

[표 3-4] 3D 지형공간 정보처리 웹 시스템 환경

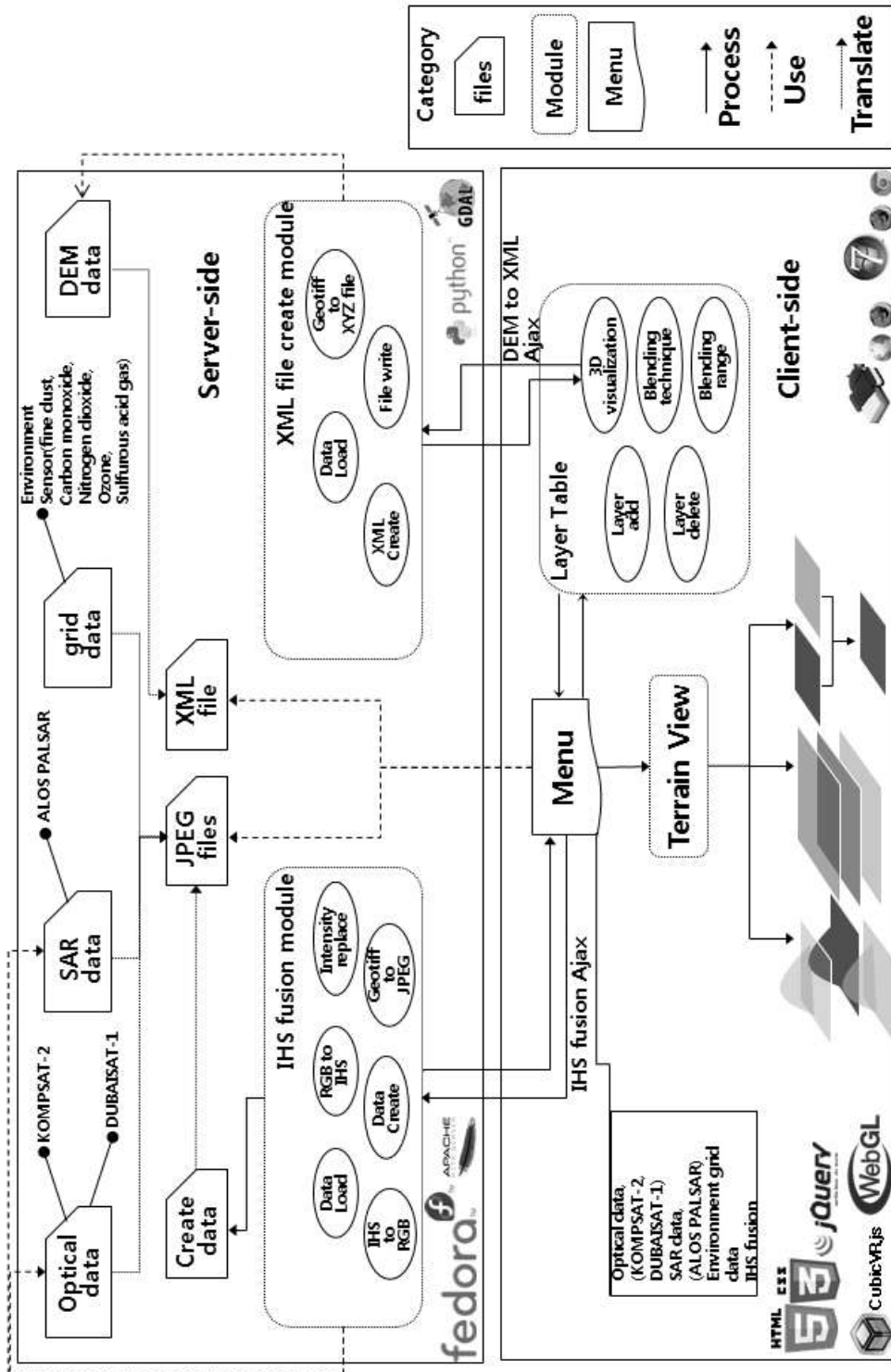
	Name	Version	Note
Server-side (Web Server)	Linux Fedora	16	Operating System
	Apache	2.22.22	Web Server
	GDAL	1.7.1	Geotiff Library
	Python	2.7	Script Language
Client-side	HTML	5	Web Language
	Javascript	1.8.5	Script Language
	CSS	3	Style Sheet
	jQuery	1.8.0	javascript Library
	CubicVR.js	-	WebGL Framework

2. 시스템 구성도 및 흐름도

위에서 언급한 것과 같이 이 연구의 시스템은 서버 사이드와 클라이언트

사이드로 구분된다. 서버에서는 실제 지형공간 데이터를 저장해 놓고 있으며 이를 Python 스크립트와 GDAL 라이브러리를 통해 실시간 처리가 가능하도록 구성하였으며 클라이언트에서는 HTML5, Javascript, CSS3를 이용하여 웹 앱의 사용자 인터페이스를 구성하였으며, jQuery를 통해 서버와 Ajax 통신을 하여 지형공간 데이터를 WebGL 프레임워크인 CubicVR.js를 통해 3D 시각화였다.

웹 시스템의 세부적인 처리 흐름은 다음과 같다. 클라이언트에서 사용자가 영상을 선택하게 되면 서버에서 실제 지형공간 데이터를 가지고 GDAL 라이브러리를 통해 미리 만들어진 이미지 파일을 사용하여 WebGL Landscape 객체를 생성하게 된다. 이렇게 생성된 Landscape 객체는 클라이언트 상에서 레이어 테이블에 저장되고 사용자가 영상을 선택할 때마다 레이어 테이블에 지형공간 레이어는 추가된다. 사용자는 레이어를 통하여 다중 시각화 및 단일 시각화를 할 수 있게 된다. 또한 테이블에서 블렌딩 범위와 기법 및 3D 시각화 선택을 할 수 있다. 블렌딩은 클라이언트에서 직접 CubicVR.js를 이용하여 효과가 적용되며, 투명 범위인 0부터 불투명 범위인 1까지의 범위 내에서 적용 되도록 사용자 인터페이스를 구성하였다. 또한 블렌딩 기법을 옵션 선택을 통해 위에서 언급한 기법 네 가지를 선택 할 수 있도록 하였다. 3D 시각화의 경우 서버에서 jQuery Ajax 통신을 통해 실제 DEM 데이터에서 Python 스크립트를 통해 GeoTiff파일을 XYZ 텍스트 파일로 생성이 되고 이를 속성 값 X, Y, Z를 가진 XML 파일 생성하게 된다. XML 파일이 생성되면 속성 값 Z 값을 가지고 높이 값이 적용된다. 마지막으로 광학 데이터와 SAR 데이터의 IHS 영상 융합 처리 과정은 다음과 같다. 사용자가 IHS 융합을 선택한 후 광학 영상과 SAR 영상을 선택이 완료되면 서버와 jQuery Ajax 통신을 시도하게 되고 Python 스크립트를 통해 실제 광학 및 SAR의 GeoTiff 파일을 가지고 실시간으로 처리가 IHS 융합 프로세싱이 서버에서 구동된다. 구동이 완료 된 후 생성된 GeoTiff 파일을 웹 브라우저에서 보여주기 위해 GDAL 라이브러리를 통해 이미지 파일로 변환이 완료 되면 다른 지형공간과 마찬가지로 시각화를 하게 된다. <그림 3-8>은 웹 시스템의 구성도 및 흐름을 나타낸 것이다.



〈그림 3-8〉 3D 지형공간 정보처리 웹 시스템 구성도 및 흐름도.

제 4 장 3D 지형공간 정보처리 시스템 구현 결과

제 1 절 3D 다중 시각화 구현 결과

이 연구에서 구현된 3D 지형공간 정보처리 시스템은 Kim (2012) 시스템을 기반으로 하고 있다. HTML5를 이용하여 구현을 하였기 때문에 플랫폼에 독립적이며, 데스크톱 및 스마트 장치의 웹 브라우저에서도 구동이 가능하다. 그러므로 이 연구에서 구현된 시스템을 다양한 장치 및 다양한 웹 브라우저에서 테스트 하였다. 하지만 테스트할 때 고려해야 될 사항으로 웹 브라우저가 기본적으로 HTML5의 WebGL을 지원해야 한다. WebGL을 완벽히 지원하는 브라우저로 데스크톱 웹 브라우저의 크롬 브라우저와 파이어폭스 브라우저이며, 안드로이드 스마트폰 장치에서는 안드로이드 4.1 버전의 기본 웹킷 브라우저와 파이어폭스 브라우저, 태블릿 PC에서는 파이어폭스 브라우저이다. 이 연구에서는 앞에 언급한 세 가지의 장치에 탑재된 다섯 종류의 브라우저에 대해 구동 결과를 확인 하였다. 태블릿 PC 중 애플의 아이패드에는 현재 WebGL을 지원하고 있지 않기 때문에 테스트 장치에 포함하지 않았으며, 안드로이드 기반인 태블릿 PC는 HTML5의 지원율이 스마트폰과 같은 버전이라고 해도 아직까지 기본 웹킷 브라우저는 WebGL을 지원하고 있지 않아 파이어폭스 웹 브라우저를 가지고 테스트를 하였다.

사용자 인터페이스의 경우 HTML5의 새로운 엘리먼트인 header, nav, section, footer를 이용하였다. 새롭게 생긴 엘리먼트들은 구조적인 문서 형태를 지원하며 이러한 작업은 다양한 장치에 구동하기 위한 필수 작업이다. 구조화된 웹 앱은 CSS3의 미디어 쿼리를 이용해 용이하며 장치별 사이즈가 변경 되는 사항을 판별하여 거의 모든 장치에 동일하게 구성 할 수 있다. <그림 4-1>에서 보이는 것과 같이 header 부분에는 웹 앱의 타이틀을 보여주고 있으며, 지형공간 데이터를 선택 할 수 있도록 nav 엘리먼트 내부에 메뉴를 구성 하였고, 그 외에도 모바일기기에서 해상도가 작은 것을 고려하여 왼쪽

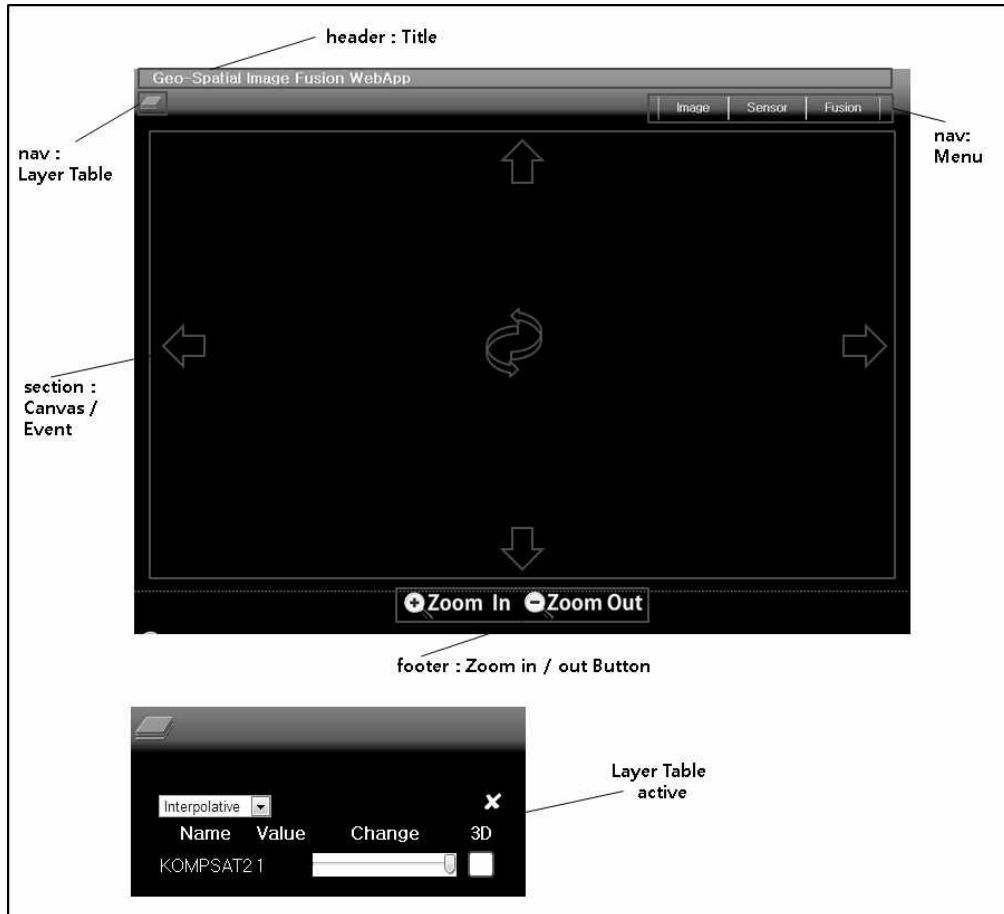
레이어 아이콘을 통해 사용자가 터치 및 클릭 하였을 때만 레이어 테이블이 보이도록 구성하였다. 메뉴 구성은 메인 메뉴와 서브 메뉴로 나뉘지며 메인 메뉴에는 지형공간, 환경정보, 융합 총 세 개로 구분 하였으며, 지형공간 메뉴의 서브 메뉴는 위성영상 이미지, 환경정보 메뉴의 서브 메뉴는 총 다섯 개의 대기 환경정보와 한 개의 방사능 정보이다. 마지막으로 융합 메뉴의 서브 메뉴로는 IHS로 구성하였으며 <그림 4-2>에 나타내고 있다.

section 엘리먼트에는 실질적인 지형공간 데이터가 시각화 되는 곳이며 HTML5의 Canvas로 되어 있다. 실제로 보이진 않지만 가장 자리 부분을 위, 아래, 왼쪽, 오른쪽으로 이동 할 수 있도록 Canvas 내부에 이벤트를 적용하였으며, 가운데 부분은 원하는 방향으로 드래그 하게 되면 지형공간이 회전 할 수 있도록 이벤트를 적용하였다. 또한 footer 엘리먼트를 통해 하단 부분에 확대 및 축소 아이콘을 넣어 시각화된 지형공간을 확대 및 축소 할 수 있도록 사용자 인터페이스를 구성하였다.

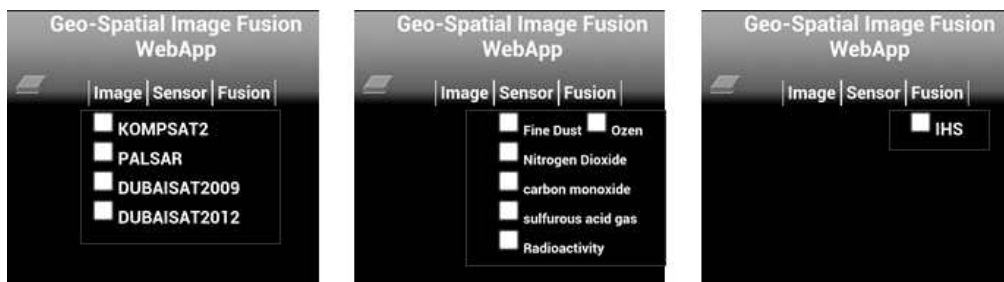
기본적으로 이번 절에서는 3D 다중 시각화 구현 결과만을 가지고 첫 번째 테스트로 광학 영상의 3D 다중 시각화로 동일한 위성의 다른 시각 때의 지형공간 데이터 두 개를 이용하여 지형 변화를 확인하였고, 두 번째 테스트로 광학 영상과 미리 생성된 임의 환경 그리드 데이터를 3D 다중 시각화 하였다. 마지막 테스트로는 광학 영상과 SAR 영상을 3D 다중 시각화 하였다. [표 4-1]은 이 연구에 구현된 시스템의 테스트 웹 브라우저에 대해 사양을 정리한 표이다.

[표 4-1] 장치별 테스트 웹 브라우저 버전

Desktop		Smartphone		Tablet PC	
Name	Version	Name	Verison	Name	Version
Chrome	23.0	Webkit	Android 4.1	Firefox	16.0.2
Firefox	16.0.2	Firefox	16.0.2	-	



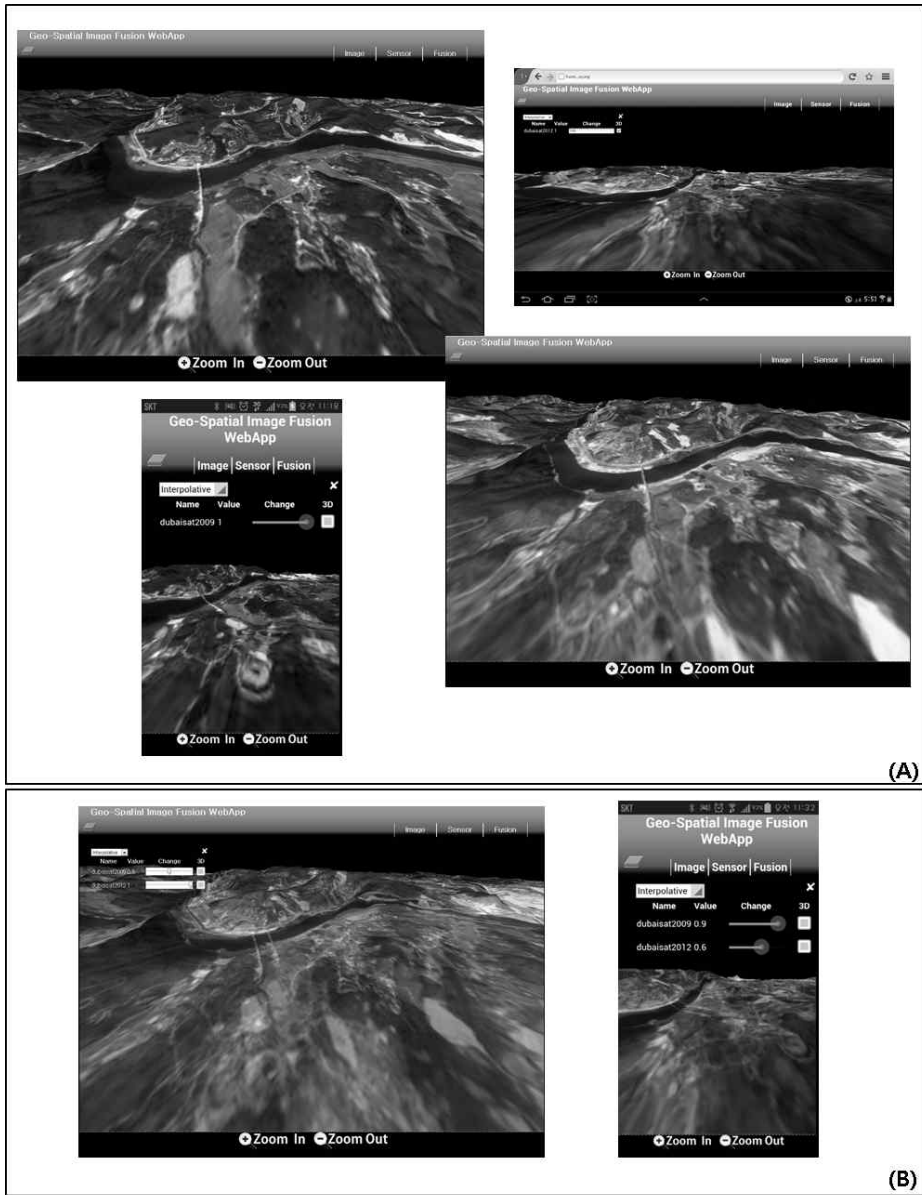
〈그림 4-1〉 사용자 인터페이스 구성.



〈그림 4-2〉 3D 지형공간 정보처리 시스템 메뉴 구성.

1. 광학 영상 3D 다중 시각화

동일한 위성의 다른 시각에 촬영된 지형공간을 동시에 시각화하면 지역 변화를 쉽게 볼 수 있다. 이 시스템에서 블렌딩 효과를 이용하여 다중 시각화를 하여 지역 변화를 웹 브라우저에서 볼 수 있도록 구현하였다. 테스트 지형공간 데이터로는 Dubaisat-1 영상 데이터의 2009년 10월 8일 영상과 2012년 4월 17일 영상의 세종시 지역이다. 세종시 지역은 행정중심복합도시로 채택이 된 후 현재까지도 공사를 계속 하고 있으므로 이러한 변화를 보기엔 적합한 지역이다. <그림 4-3>는 Dubaisat-1 2009년 데이터와 2012년 데이터를 각각 3D 시각화한 그림과 두 개의 데이터를 CubicVR.js에서 기본으로 적용하고 있는 보간형 블렌딩 기법으로 블렌딩 효과를 적용하여 3D 시각화한 그림이다. <그림 4-3>의 (A)에는 총 네 개의 구현 결과를 보여주고 있으며, 데스크톱 크롬 웹 브라우저, 태블릿PC 파이어폭스 웹 브라우저, 안드로이드 스마트폰 기본 웹 킷 브라우저 결과를 보여주고 있다.

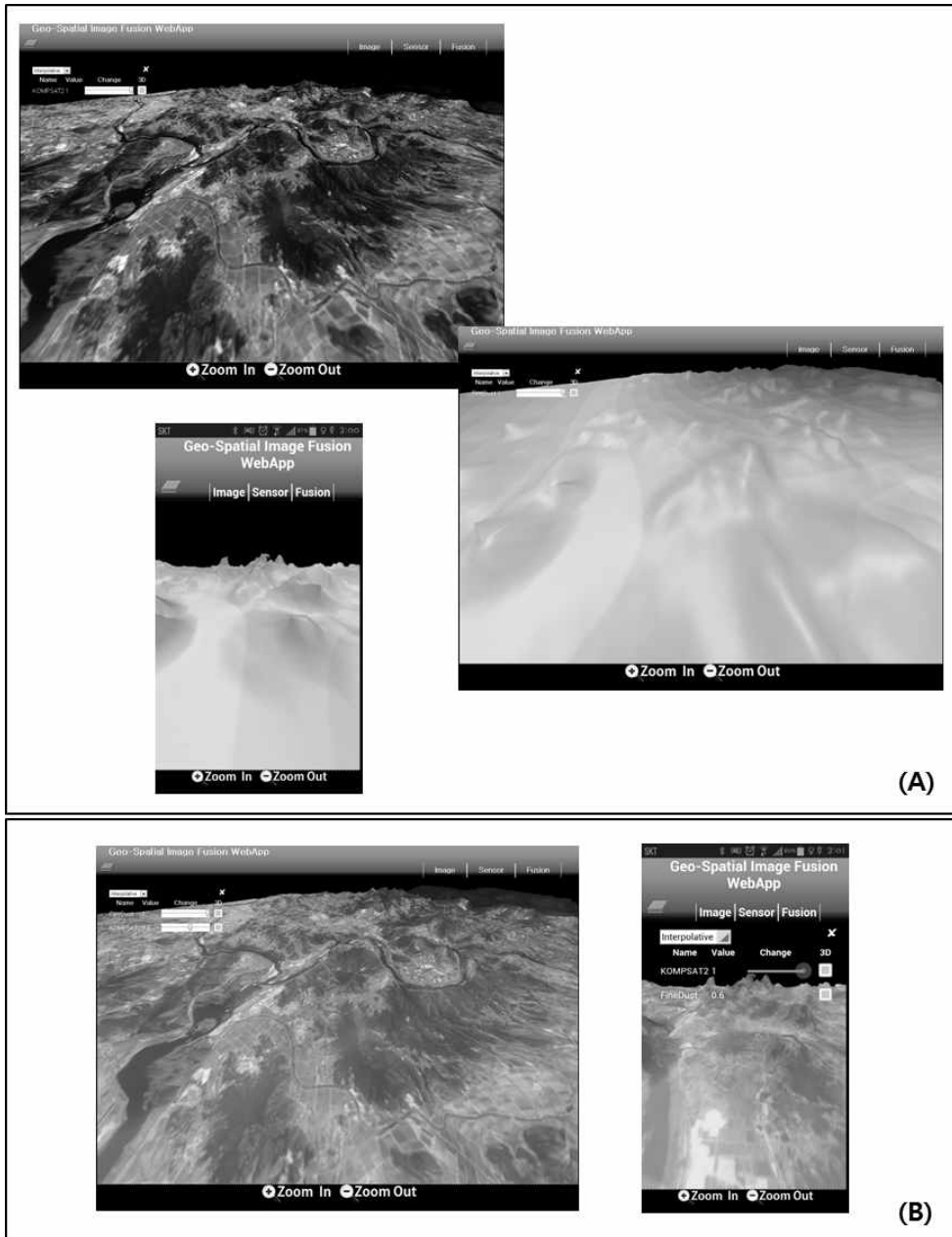


〈그림 4-3〉 데스크톱, 스마트폰, 태블릿 PC에서의 광학 영상 (Dubaisat-1) 3D 다중 시각화: (A) 촬영 시간이 다른 광학 영상 3D 시각화, (B) 광학 영상 3D 다중 시각화 결과.

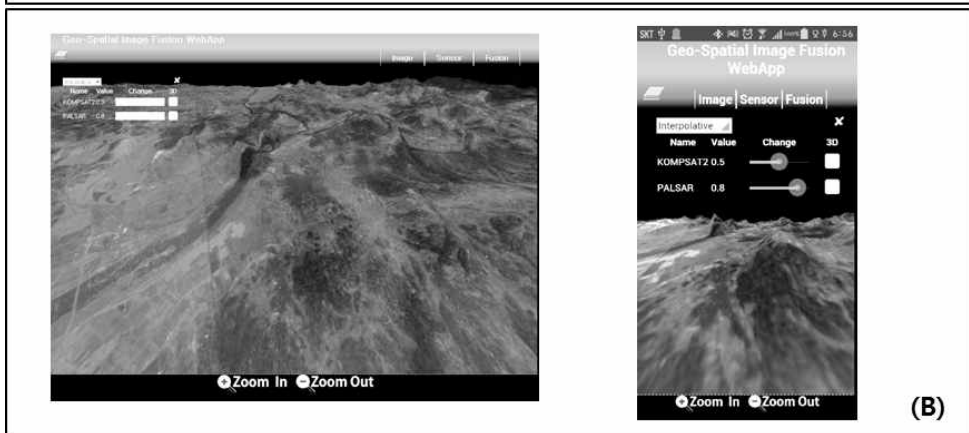
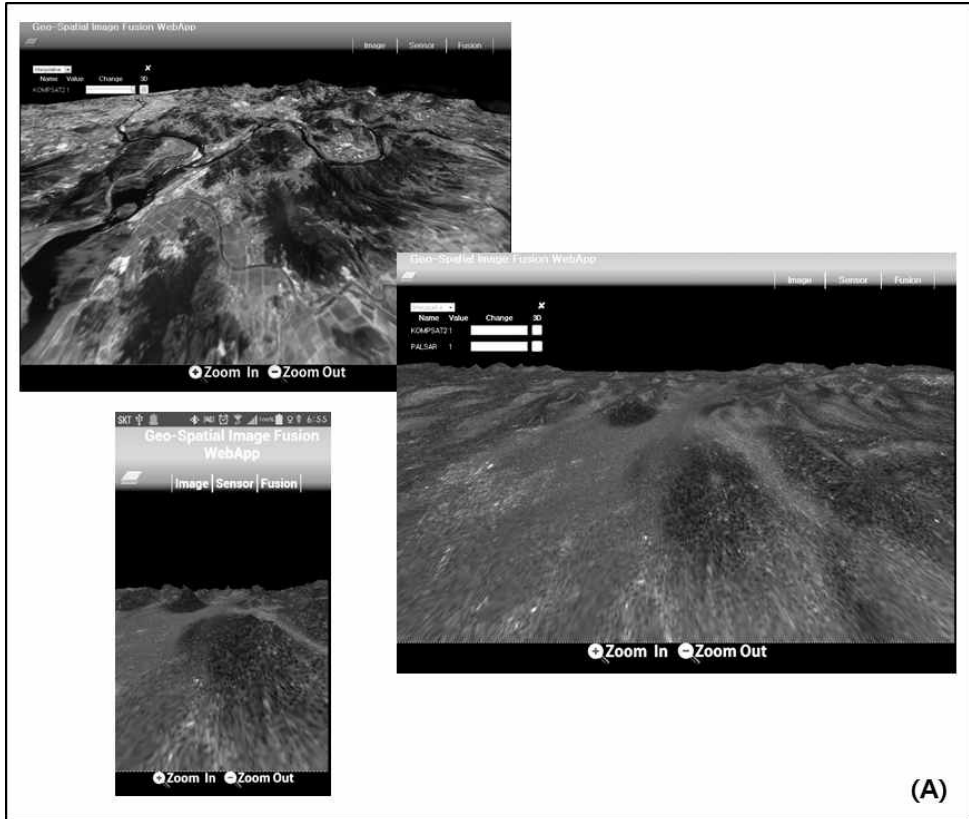
2. 광학 영상과 다양한 지형 정보 3D 다중 시각화

이 연구에서는 광학 영상 뿐 아니라 그 외의 다양한 지형공간을 다중 시각화함으로써 여러 가지 결과를 사용자가 볼 수 있다. 이 연구에서 광학 영상 외에 사용된 데이터로는 환경 정보와 SAR 데이터가 있다. 환경 정보는 일반 수치 데이터로 포인트마다 값이 나오지만 이를 3장 2절에서 설명한 처리 과정을 통해 그리드 데이터로 생성함으로써 지형공간 데이터로 생성 할 수 있다. 이러한 지형공간 데이터가 사용됨으로써 사용자 인근 지역의 환경 정보를 아는 것이 아니라 사용자가 위치하고 있는 환경 정보를 예측 할 수 있는 장점이 있다. 이 연구에서는 총 다섯 개의 환경 측정 센서인 일산화탄소, 미세먼지, 이산화질소, 오존, 아황산가스 중 미세먼지 데이터의 결과만 가지고 테스트를 하였으며 임의의 데이터를 사용하였다. 실제 데이터를 사용 할 수도 있지만 이 연구에서는 다중 시각화에 대한 정확한 구분을 확인 하기 위해 임의의 데이터를 사용 하였고 이를 실제 데이터로 변환하여 사용 할 수도 있다. 환경 그리드 데이터 외에 다중 시각화에 사용된 데이터는 SAR 데이터이다. 광학 데이터와 SAR 데이터를 다중 시각화함으로써 기존 광학 영상의 다중 시각화 외의 또 다른 정보를 볼 수 있다.

또한, WebGL은 가산형, 감산형, 곱셈형, 보간형 블렌딩 기법을 적용 할 수 있다. 이 연구에서 사용된 CubicVR.js에 기본적으로 보간형 블렌딩 기법이 적용 되어 있지만, 이 기법 외에 나머지 세 개의 기법을 적용 할 수 있도록 WebGL의 기본 함수를 이용하여 CubicVR.js에 추가하였고 이를 사용 할 수 있도록 사용자 인터페이스를 구성하였다. <그림 4-4>는 광학 영상과 환경 그리드 데이터를 3D 다중 시각화 한 결과이며, <그림 4-5>은 광학 영상과 SAR 영상 데이터를 3D 다중 시각화 한 결과이다. 또한 <그림 4-6>는 스마트폰 상에서 다양한 블렌딩 기법을 적용 하여 광학 영상 데이터와 SAR 영상 데이터를 3D 다중 시각화 한 결과이다.



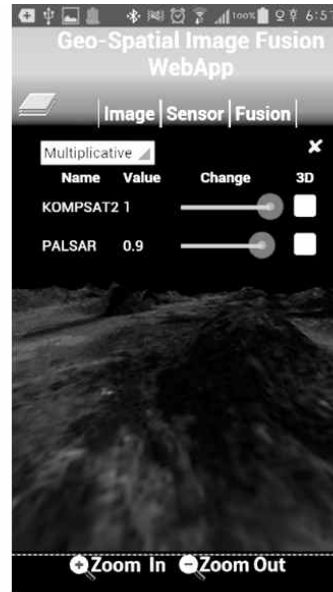
〈그림 4-4〉 데스크톱 및 스마트폰에서의 광학 영상(KOMPSAT-2) 및 환경 그리드 데이터 3D 다중 시각화: (A) 광학 영상과 임의 환경 그리드 데이터 3D 시각화, (B) 광학 영상과 임의 환경 그리드 다중 3D 시각화.



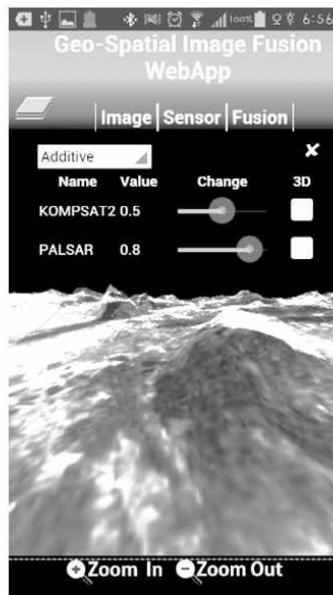
〈그림 4-5〉 데스크톱 및 스마트폰에서의 광학 영상(KOMPSAT-2) 및 SAR 영상(ALOS PALSAR) 데이터 3D 다중 시각화: (A) 광학 영상과 SAR 영상 데이터 3D 시각화, (B) 광학 영상과 SAR 영상 데이터 3D 다중 시각화.



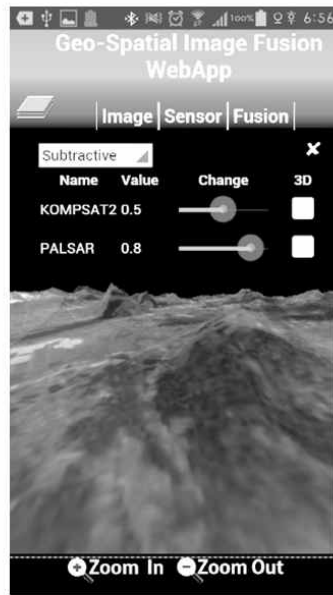
(A)



(B)



(C)



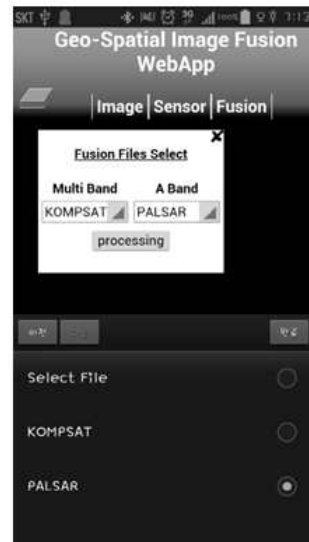
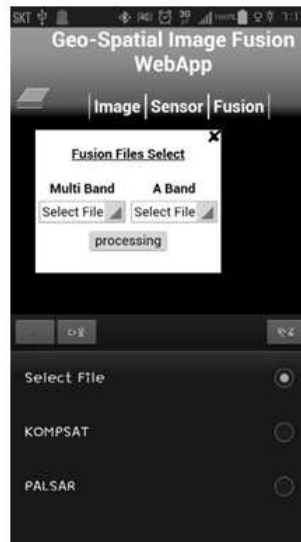
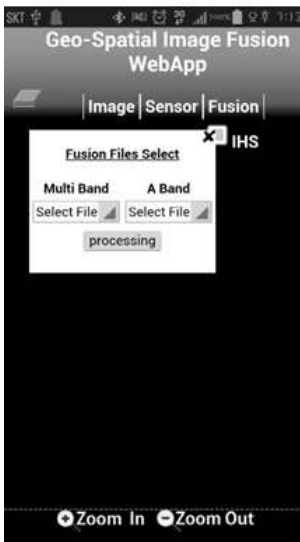
(D)

〈그림 4-6〉 블렌딩 기법에 따른 적용 결과: (A) 블렌딩 기법 선택 옵션, (B) 곱셈형 블렌딩 기법 적용 결과, (C) 덧셈형 블렌딩 기법 적용 결과, (D) 감산형 블렌딩 기법 적용 결과.

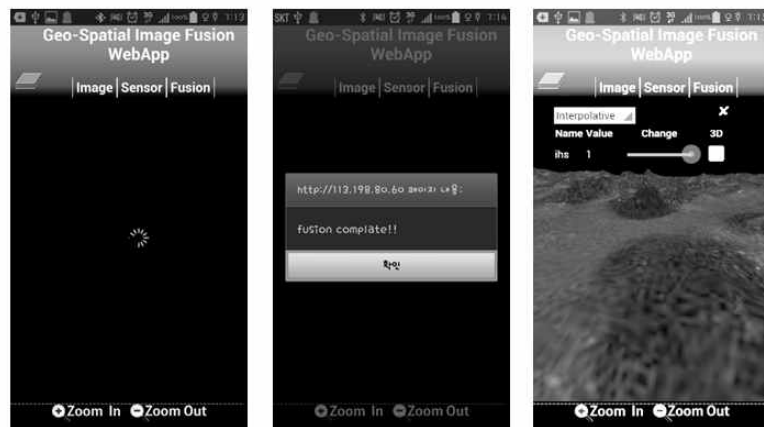
제 2 절 광학 영상과 SAR 영상 융합 구현 결과

이 연구의 마지막 구현 결과로 광학 영상과 SAR 영상의 융합 구현 결과이다. 융합 알고리즘은 앞에서 언급한 것과 같이 IHS 융합 방법을 사용 하였으며, 이 시스템은 서버에는 RGB 밴드를 가진 광학 영상(KOMPSAT-2) 데이터와 단일 밴드 SAR 영상(ALOS PALSAR) 데이터를 가진 지형공간 데이터가 GeoTiff 파일로 저장 되어 있다. 사용자가 Fusion 메뉴를 선택하여 IHS 서브 메뉴를 선택하게 되면 파일 선택 레이어가 왼쪽 윗부분에 나타나게 되며, 광학 영상 데이터 및 SAR 영상을 선택 한 후 프로세싱 버튼을 클릭하게 되면 jQuery Ajax 통신을 통해 서버에 전달되고 GDAL 라이브러리를 사용한 Python 스크립트를 통해 서버에 저장 되어 있는 광학 영상 데이터와 SAR 영상 데이터를 가지고 실시간으로 IHS 융합 프로세싱이 서버에서 수행된다. 광학 영상 및 SAR 영상은 대부분의 용량이 크기 때문에 빠른 시간에 완료가 되지 않으며, 서버에서 수행이 완료 되면 서버에는 새로운 IHS 지형공간 데이터가 특정 폴더에 저장된다. GeoTiff 저장과 동시에 웹 브라우저에서 보여지기 위해 GDAL 라이브러리를 통해 이미지 파일로 동시에 특정 폴더에 생성된다. 모든 과정이 완료되면 jQuery Ajax 통신을 통해 클라이언트에게 완료 메시지를 클라이언트에 보내주게 되고 웹 브라우저 상에 텍스트를 받아 완료 알람창이 보여준 후 IHS 융합 결과가 시각화 된다. 이렇게 시각화 된 지형공간은 다른 데이터와 마찬가지로 3D 시각화, 중첩 시각화 기능을 동일하게 적용 할 수 있다.

〈그림 4-6〉은 데스크톱 및 스마트폰에서 IHS 융합 메뉴를 클릭 및 터치하였을 때 나타나는 결과이므로 사용자가 단순히 지형공간 데이터를 선택하기만 하면 구동이 되도록 사용자 인터페이스를 구성하였고, 〈그림 4-7〉에서는 멀티 밴드 및 싱글 밴드를 선택 후 서버에서 프로세싱이 완료되기 전까지의 과정, 완료 되었을 때의 알림, 마지막은 IHS 융합 결과를 3D 시각화 한 그림이다.



〈그림 4-7〉 데스크톱 및 스마트폰의 IHS 융합 메뉴 구성.



〈그림 4-8〉 IHS 융합 과정: 과정 완료시 알림, IHS 융합 3D 시각화.

제 5 장 결 론

정보통신 분야에서는 데스크톱에서 구동되는 클라이언트 소프트웨어뿐만 아니라 스마트기기에서 구동되는 모바일 앱과 웹상에서 구동되는 웹 앱의 수요가 증가되고 있다. 하지만 공간 정보 분야에서의 지형공간 정보 처리 시스템은 아직 클라이언트 소프트웨어 중심으로 이를 이용하여 다양한 융합 알고리즘을 적용하여 지형공간을 분석하거나 처리하는 연구 및 서비스가 보편화되어 있으며, 상업적 목적이 대부분이므로 가격이 비싸고 개인이 사용하기에는 힘든 상황이다. 모바일 앱에서는 아직까지 지형공간을 배경영상으로만 사용한 연구 및 서비스가 대부분이었으며 정보처리 관련된 서비스 및 연구는 미흡한 실정이었다. 웹 기반으로 서비스가 되어있는 국내 지도 제공 업체 대부분도 지형공간 데이터를 일반 배경 영상으로 사용하거나, 플러그인을 사용하여 데스크톱 웹 브라우저에서만 구동이 가능하게 서비스 되고 있는 실정이었다.

모바일기기는 휴대성이 편리하므로 일반 생활뿐만 아니라 업무 또는 교육과 같은 전문용도로도 많이 사용되고 있다. 하지만 현재까지 공간 정보 분야의 모바일 앱의 서비스의 경우 전문가들이 사용 하는 전문 앱보다는 일반 사용자들 기준으로 서비스가 제공되어 지고 있다. 현재 모바일기기의 발전은 급속도로 성장하고 있으며, 해상도가 높은 다양한 스마트 기기들도 계속적으로 출시되고 있다. 그렇기 때문에 앞으로도 데스크톱 못지않게 발전 될 것으로 예상되므로 전문가를 위한 서비스가 필요하다.

이 연구에서는 기존에 클라이언트 기반으로 서비스 되고 있는 시스템의 기능을 오픈 소스를 이용하여 웹상에서 실시간으로 처리 할 수 있는 시스템을 설계 및 구현함으로써 공간 정보 분야에 접목하여 새로운 가능성을 보여 주었다. 이 시스템을 활용 하면 현재 사용자가 가지고 있는 다양한 지형공간 데이터를 소프트웨어 설치 없이 웹상에서 시각화 할 수 있다. 하나의 데이터만 시각화 하는 것이 아닌 두 개 이상의 데이터를 융합 하여 보거나, DEM 데이터가 있다면 3D로 시각화가 가능하다. 기술적으로 HTML5의 분석을 통해 현재 데스크톱 웹 브라우저와 스마트기기의 웹브라우저에서의 지원율을 조사하여 HTML5의 적용률을 확인한 결과 대부분의 웹 브라우저에서 기술을 적

용하거나, 적용 예정인 것을 알 수 있었으며 HTML5의 Javascript API에 대한 기술은 기존 HTML4와는 다르게 향상이 많이 되었기 때문에 클라이언트 소프트웨어 및 모바일 앱 수준의 서비스를 제공이 가능 할 것으로 예상된다. 또한, 편리하게 사용 할 수 있는 jQuery, jQTouch, SenchaTouch 등 다양한 라이브러리들이 많이 출시되고 있기 때문에 웹에 대한 활용 가치도 높아지고 있다. 특히, 이 연구에서 사용된 HTML5의 Javascript API 기술 중 하나로 OpenGL ES 2.0을 기반인 WebGL은 웹 브라우저를 통해 인터랙티브한 3D 렌더링이 가능하므로 공간정보 분야에서 사용되는 지형공간 데이터 및 건물들을 2D 렌더링 보다는 더욱더 현실적으로 서비스 할 수 있을 것이라고 예상된다. WebGL을 이용한 프레임워크도 많이 출시되고 있어 복잡한 3D 렌더링의 간단하게 구현 할 수 있도록 도와주고 있다. 물론 현재까지 WebGL을 완벽하게 지원하고 있는 웹 브라우저는 일부 밖에 되지 않지만 웹 브라우저 제공 업체들의 지속적인 업데이트로 향후 모든 브라우저가 WebGL을 지원 할 것으로 예상된다. 기존 중첩 시각화 서비스의 경우 플러그인을 사용하기 때문에 데스크톱의 웹 브라우저에서만 2D 또는 3D 렌더링을 이용하여 지형의 변화를 알 수 있었으며, 웹상에서의 지형공간 실시간 정보처리에 대한 서비스도 미흡한 실정이다. 하지만 이 연구에서는 플러그인을 사용하지 않았기 때문에 다양한 장치에서 구동이 가능하며 WebGL을 사용했기 때문에 3D 중첩 시각화가 가능하다. 또한 현재 제공되지 않은 실시간 정보처리 기능을 웹상에서 구동 가능하도록 구현하여 사용 가능성을 테스트 하였다. 실시간 정보처리 알고리즘으로는 IHS 융합 처리를 구현하였으며 웹상에서 테스트 결과 실시간으로 처리가 가능한 것을 볼 수 있었다. 이 연구의 시스템은 공간정보 분야의 새로운 유형의 웹 시스템이며 가능성을 보았고 초기 단계로 앞으로 웹 시스템의 다양한 지형공간 정보처리에 대한 연구에 기여 할 것으로 예상된다.

【참고문헌】

1. 국내문헌

- 강상구, 이기원. (2011). 위성영상정보 기반 코너 포인트 객체 추출 안드로이드 스마트폰 앱 개발, *대한원격탐사학회지*, 27: 33-41.
- 권지인. (2009). 국내외 모바일 애플리케이션 마켓 현황 분석, *Biz Comm*.
- 권진혁. (2012). 『오감(Five Senses) 인터넷 기술 동향 연구』, *정보통신산업진흥원*.
- 김광섭, 이기원. (2011(a)). 위성영상정보 공간 메타데이터 검색 하이브리드 앱 설계 및 시험 구현, *한국원격탐사학회지*, 27: 203-211.
- 김광섭, 이기원. (2012(b)). 대기환경정보와 방사능정보 위치기반 서비스 스마트폰 앱 설계 및 시험구현, *한국지리정보학회지*, 14: 161-171.
- 박세환. (2011). 3차원 GIS 및 지도 데이터 기술 동향, *정보통신산업진흥원 주간기술동향*, 1497:1-10.
- 신동형, 유미연. (2012). 『휴대폰 산업 분석』, *LG경제연구원 report*.
- 손맥, 박수만, 이윤희. (2010). 스마트 사회 구현을 위한 공간정보서비스 활용 전략, *한국정보화진흥원 CIO report Vol. 29*.
- 이은민. (2011). HTML5가 웹 환경에 미치는 영향, *정보통신정책연구원*, 23:54-65.
- 전종홍, 이승윤. (2012). HTML5 기반의 웹 플랫폼 기술 표준화 동향, *전자통신연구원 전자 통신 동향분석*, 27(4): 84-95.
- 전황수. (2011). LBS 시장 및 업체 동향, *정보통신산업진흥원 주간기술동향*, 1496:1-14.
- 채원석, 박찬우, 최완, 안세영, 노병석, 이준우. (2012). 모바일 웹 앱을 위한 HTML5 및 프레임워크 동향, *한국전자통신연구원*.

케이티 종합기술원. (2011). 모바일 애플리케이션 촉매제 스마트폰 센서, IT 지식포털 디지에도.

한국인터넷진흥원. (2012). 스마트 모바일 강국 실현 2012년 상반기 스마트폰 이용실태조사, ISIS 인터넷통계정보검색시스템.

2. 국외문헌

- Boulos, M. N. K., Warren, J., Gong, J., & Yue, P. (2010). Web GIS in practice VIII: HTML5 and the canvas element for interactive online mapping』, *International Journal of Health Geographics*, 9: 1-14.
- Bretschneider, T., & Kao, O. (2000). *Image Fusion in Remote Sensing, Proceeding of the 1st Online Symposium of Electronics Engineers*.
- Butchar, B. & King, M. (2010). *Location Based Services without the Cocoa*, Presentation at the Institutional Web Managers Workshop.
- Canga E. F. (2002). Image Fusion, *Signal & Image Processing Group*.
- Cantor, D., & Jones, B. (2012). 『WebGL Beginner's Guide』, *PACKT PUBLISHING*.
- Childs, C. (2004). 『Interpolating Surfaces in ArcGIS Spatial Analyst』, *ESRI Education Services*.
- Corcoran, P., Mooney P., Winstanley A., & Bertolotto M. (2011). Effective Vector Data Transmission and Visualization Using HTML5, *Proceedings of GISRUUK2011 Conference*.
- CubicVR 3D engine. (2012). 『CubicVR.js API Reference』, <https://github.com/cycliffe/CubicVR.js/wiki/CubicVR.js-API-Reference>.
- Johansson, H. (2010). Rich Web Map Applications: An assessment of performance, functionality and implementation of Rich Internet Application techniques in web-based GIS, *Chalmers University*.
- Jumtap. (2011). Mobile STAT Simple Targeting & Audience Trends,

- Jumtap Monthly report.
- Kim, K., & Lee K. (2011). Remote Sensing Application Based on Web Mapping API and HTML5, *Proceedings of Asian Conference on Remote Sensing*.
- Kim, K., & Lee K. (2012). 3D fusion of Remote Sensing Data On Smartphone, *Proceeding of International Symposium On Remote Sensing*.
- Kim, K., & Lee K. (2006). Development of Mobile 3D Terrain Viewer with Texture Mapping of Satellite Images, *Korean Journal of Remote Sensing*, 22: 351–371.
- Klonus, S., & Ehlers M. (2009). Performance of evaluation methods in image fusion」, *Proceeding of the 12th International Conference on Information Fusion*.
- Lee, K., & Dong W. (2008). A Prototype Implementation for 3D Feature Visualization on Cell Phone using M3G API, *Korean Journal of Remote Sensing*, 24: 246–250.
- Mitchell, H. B. (2007). 『Multi-Sensor Data Fusion: An Introduction』, *Springer*.
- Power M. (2011). 『Mobile Web Apps』, *JISC Centre for Educational Technologist & Interoperability Standards*.
- Sauerwein, T. (2010). Evaluation of HTML5 for its Use in the Web Mapping Client OpenLayers, *Fachhochschule Kaiserslautern University of Applied Sciences*.
- Stathaki, T. (2008). 『Image Fusion Algorithms and Applications』, *ELSEVIER*.
- Trevett, N. (2011). Building Markets for Advanced Devices through Open

Standards, Persentation at Game Developers Conference.

W3C Working Group, (2012). 『HTML5 differences from HTML4』 ,
<http://www.w3.org/TR/html5-diff/>.

WebGL Public Wiki, (2012). 『WebGL User Contributions』 ,
http://www.khronos.org/webgl/wiki/User_Contributions.

ABSTRACT

Design And Implementation Of 3D Geo-spatial Information Processing System based on HTML WebGL

Kim, Kwang Seob
Major in Information System Engineering
Dept. of Information System Engineering
The Graduate School
Hansung University

Smartphone has popularized and developed rapidly in information and communications technology field. Study and services are being announced in the variety field using these. Recently, smart devices were published such as tablet PC, smart TV and smart car, thus resulted in major change in mobile markets. These devices can provide mobile application by personal developers, as compared with the existing feature phone. In the spatial information field, applications of smart phone are providing map search services and location-based services etc.. However, the images of spatial information have been used for simple visualization.

With this motivation, this study is to design and implement web system. It is able to a variety of Geo-spatial data 3D visualization and data processing. Existing web-based systems are considering desktop web browser. However, this study applied HTML5 to mobile as well as

desktop web browser for show. The WHATWG (Web Hypertext Application Technology Working Group) began work on the new standard, W3C(World Wide Web Consortium) and WHATWG are currently working together on the development of HTML5. HTML5 is international standards for services without plug-in about existing HTML. Therefore, it included variety of libraries, HTML5 WebGL was used for 3D Geo-spatial information visualization and processing in this study. WebGL was based on OpenGL ES 2.0(OpenGL for Embedded Systems 2.0). Unlike Adobe Flash and Microsoft SilverLight, it can apply for dynamic 3D technology without plug-in on web browsers. Because of this, it is possible for WebGL to run mobile web browsers as well as desktop web browsers. Geo-spatial data has used optical data, SAR(Synthetic Aperture Radar) data and Digital map data and grid data sets of air environment, and area is multifunctional administrative city. these are as follows: Optical data is KOMPSAT-2 image and DUBAISAT-1 image, SAR data is ALOS PALSAR image, and grad data sets of air environment are measuring station. The technique used in information processing applied WebGL blending technique and IHS fusion algorithm. IHS fusion is often used from the spatial information field. It is expected that this thesis will contribute to the future smartphone application developments in the remote sensing developers communities.

keywords : HTML5, WebGL, Mobile Web App, 3D Geo-Spatial Fusion, Open Source