

## WebGL기반 협업 3D 모델링 시스템

편혜걸\*, 안향아\*\*, 박진호\*\*\*

### 요약

본 논문에서 제시하는 협업 3D 모델링 시스템은 기존 협업 시스템의 문제점을 보완하여 보다 효율적인 작업이 가능하게 한다. 현재 3D 모델링 작업은 오프라인에서 작업하여, 파일의 형태로 주고받는 협업 시스템이 대부분이다. 이러한 협업 시스템은 즉각적인 인터랙션이 불가능하며, 피드백 사이의 시간 지연 발생, 파일을 볼 수 있는 부가적인 프로그램이 필요하다. 본 논문에서는, 웹이라는 네트워크의 특성을 이용하여 이러한 문제점을 해결하고 나아가 새로운 형태의 모델링 협업 시스템을 제안한다. 3D 웹 구현 분야에서 각광받는 기술인 웹지엘(Web Graphics Library; WebGL)을 이용하여 브라우저에서 3D 오브젝트를 생성하고, 이를 서버-클라이언트 통신을 이용하여 여러 사용자가 동시에 조작할 수 있도록 설계 하였다. 따라서 실시간 인터랙션과 피드백이 가능하며 소프트웨어 설치의 필요성을 낮추고, 브라우저가 설치된 디바이스를 사용하면 어디서든 접근을 할 수 있다.

키워드 : 웹지엘 라이브러리, 실시간 인터랙션, 웹, 협업, 3D 모델링

## Cooperative 3D Modelling System based on WebGL

Hae-Gul Pyun\*, Haeng-A An\*\*, Jinho Park\*\*\*

### Abstract

The proposed cooperative 3D modelling system enables more efficient modeling works by improving current cooperation systems. Current 3D modeling tasks work on off-line and most of works are cooperation systems which communicate by files. These cooperation system cannot be interactive, occurs time delay while feedback and must have some programs which can read the files. This paper solves these problems using web that has the characteristics of network and proposes new cooperation system form. Using WebGL(Web Graphics Library) which is proposing technology in technical realization of Web 3D modeling, we make 3D objects in web browser and these objects can be manipulated by server-client communication. Therefore, if people use this system, they can use real time interaction and feedback. Also this system lowers a software installation necessity and can access everytime if web browser is installed.

Keywords : WebGL Library, Real time interaction, Web, Cooperation, 3D Modeling

### 1. 서론

※ Corresponding Author: Jinho Park

Received: September 30, 2014

Revised: March 22, 2015

Accepted: April 10, 2015

\* Global School of Media, Soongsil University

\*\* Global School of Media, Soongsil University

\*\*\* Global School of Media, Soongsil University

Tel: +82-2-828-7040

email: c2alpha@ssu.ac.kr

■ 본 연구는 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2012R1A1A2006094)

현재 웹페이지는 대부분 텍스트와 이미지들로만 이루어져 있으며, 단순한 하이퍼링크(hyper link)들을 통해 이용자들 간의 정보 소통이 이루어진다. 그러나 오늘날 인터넷 공간 안에서 공유하는 콘텐츠가 텍스트, 이미지와 같은 단면적인 정보들에서 복잡하고 다양한 동영상, 3D 모델링과 같은 멀티미디어 정보들로 바뀌어 가는 상황에서 이러한 평면적이고 단순한 인터페이스가 사용자들에게 효과적으로 정보를 전달하기에는 한계가 있다. 따라서 웹상에서 3D 효과를 구현

한다면 사용자가 좀 더 효율적으로 정보를 받아들일 수 있을 것이다.

최근에는 제품 디자인, 캐릭터 디자인, 인테리어 디자인에서 영화나 광고와 같은 엔터테인먼트 분야까지 3D 콘텐츠가 많이 사용되고 있는데, 이러한 산업의 작업 환경에서 3D 콘텐츠를 볼 수 있는 웹 페이지는 굉장히 큰 장점을 가질 수 있다. 현재 3D 모델을 만드는 작업은 주로 마야(Maya), 맥스(3ds Max), 오토캐드(AutoCAD), 스케치 업(Sketchup)과 같은 3D 모델링 소프트웨어를 사용한 작업이 주를 이루고 있으며, 이러한 소프트웨어들은 모두 네트워크상의 공동 작업을 지원하지 않는다. 이러한 프로그램은 개인이 작업하기에는 편하지만 여러 사용자가 동시에 소통하면서 협업하기에는 상당히 불편하다. 즉, 각 분야에서 협업하는 구성원들과 즉각적인 피드백을 할 수 없으며 느리고 번거로운 과정을 거치게 된다. 만약 인터넷 상에서 서로의 3D 모델링을 서로 같이 보면서 수정할 부분을 고치고 필요한 부분을 더한다면 좀 더 빠르고 효율적인 작업을 할 수 있을 것이다. 따라서 본 논문에서는 효과적이고 즉각적인 소통이 가능한 3D 모델링에 초점을 맞추었다.

이를 위하여 본 연구에서는 웹지엘을 이용한 웹상에서의 3D 구현과 협업에 초점을 맞추어 인터넷에서 오브젝트 구현과 동시에 많은 사람들과 소통이 가능하도록 하였다. 이로써 복잡하고 실시간 소통이 불가능한 정보 공유가 아닌 이용자들 간의 간단하고 즉각적인 반응이 가능하도록 하여 그래픽 산업에서의 효과적인 협업과정 모델을 수행할 수 있다.

## 2. 관련 연구

현재 컴퓨터 그래픽 산업에서 기술적, 예술적 측면에서 여러 팀의 협업이 요구되고 있고 이를 효과적으로 해결하기 위해 여러 가지 연구가 진행되고 있다. 그러나 3D 모델링을 제작하는 과정에서 서로 다른 파일의 호환성 문제가 존재해왔고[1] 이 때문에 서로 다른 파일 포맷의 3D 오브젝트 파일을 효율적으로 공유하는 것에 대한 노력은 많았다. 현재 사용되는 3D 파일 포맷은 140여개에 이른다[2]. 산업 규격으로는 콜라

다(COLLADA)가 많이 사용되고 있지만 이를 사용하기 위해서는 콜라다 파서(COLLADA parser)를 사용해야하고 웹상에서 3D 데이터를 보는 새로운 접근방식을 제안하는 정도이다[3].

3D를 이용해서 협업의 효율성을 높이려는 시도도 많다. 기존의 양방향 협업만 가능했던 시스템에서 복합 협업을 가능하게 하고 3D 비디오 콘텐츠를 보여주는 통신집중 시스템을 만들어서 데이터 손실을 최소한으로 전달하는 방법을 제시하고 있다[4]. 또한 실시간 스테레오 시각 시스템에 초점을 맞춰, 정확한 3D 구현 환경으로서 사용자들 사이의 소통을 강조하고 있다[5]. 또 다른 방법으로는 같이 3D 화면을 공유하는 것이 아니라 새로운 3D 가상공간을 만들어서 협업을 이끄는 것이다[6]. 하지만 사용자들 사이에 가상공간을 만들어야 하므로 기술이 좀 더 복잡하고 얼굴을 맞대고 해야 할 일이 아닌 경우에는 효율성이 떨어지는 단점이 존재한다.

3D 콘텐츠를 웹 브라우저 환경에 더한다는 것은 어플리케이션에도 많은 이점을 더할 수 있고, 3D 장면들을 이용한 공동 작업을 할 수 있도록 도움을 주는데 큰 역할을 한다. 인터넷에서 3D 오브젝트를 보여주는 방법 중 하나로 웹지엘을 이용하는 방법이 있다. 웹지엘을 이용하여 간 해부학 3차원 인터넷 학습(E-learning Web3D)를 개발하여 학생들이 간의 구조를 훈련하도록 응용되어 사용된바 있다[7]. 또한 웹지엘뿐만 아니라 C3DL에 기반을 둔 모션뷰(Motionview)라는 웹 어플리케이션을 제안하여 아티스트나 모션캡처 스튜디오에서 사용자들이 서로 소통을 할 수 있도록 만든 바 있다[8]. 이 어플리케이션에서는 C3DL이라는 자바스크립트 라이브러리를 사용하여 웹지엘이나 3D 수학적 연산의 복잡성 없이 3D 콘텐츠를 웹 브라우저에 기본적인 3D 기능성을 제공하였다. [11]은 소셜 서비스를 위한 시공간 모델링 방안으로 장소와 시간, 이벤트, 사용자의 소셜 액티비티를 통해 휴먼 액티비티 그래프(Human Activity Graph)모델링 방안과 쿼드 릴레이션 팩터스(Quad Relation Factors)를 제안하고, 이를 위한 데이터 수집과 분석을 위한 하부 구조를 설계하였다.

### 3. 제안 시스템

본 논문에서는 인터넷 웹 브라우저 환경에서 서버-클라이언트 간의 3D 정보 교환을 통해 언제 어디서나 어떤 플랫폼을 이용하든 접근을 쉽게 할 수 있는 3D 협업 모델을 제안한다. 이를 위하여 웹 개발언어 중에서도 크로스 플랫폼을 지원하는 개발 언어인 HTML5와 자바스크립트(JavaScript)를 기반으로 제작했다.

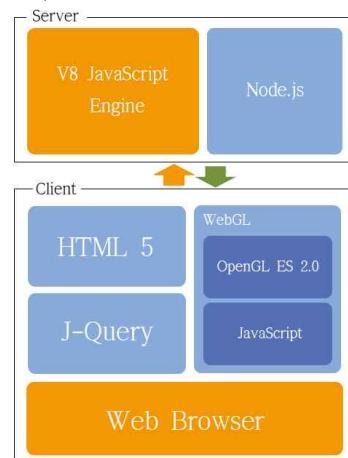
전체적인 시스템 아키텍처를 구성할 때(그림 2) 기본적인 인터페이스를 구성에는 HTML5와 JQuery로 구현했다. 웹 환경에서 그래픽을 구현할 때 캔버스(Canvas), 가변 벡터 도형 처리(SVG), 웹지엘을 사용하는 방법이 있는데 본 연구에서는 자바스크립트가 API로 제공되는 웹지엘을 사용했다. 캔버스나 가변 벡터 도형 처리는 2D 그래픽을 구현할 경우에는 문제가 없지만 3D 그래픽을 구현할 수 없다는 한계가 있다. 반면에 웹지엘은 오픈지엘(OpenGL) 라이브러리를 기반으로 작동하기 때문에 2D뿐 아니라 3D 구현하는데도 문제가 없다. 여기서 오픈지엘이란, 오브젝트를 그리는 알고리즘을 미리 구현해둔 라이브러리로 이를 이용하면 3D 그래픽을 보다 쉽게 구현할 수 있다[9]. 또한 크롬이나 파이어폭스와 같이 사람들이 많이 사용하는 웹브라우저들의 자바스크립트 엔진 성능이 향상되면서, 자바 스크립트로 구현되는 웹지엘의 성능도 많이 향상되었다[10]. 본 논문에서 제안한 시스템은 웹지엘을 이용하여 육면체(cube), 구(sphere), 원기둥(cylinder) 등 3D 오브젝트를 생성할 수 있으며, 오브젝트 변형은 웹지엘 범용 라이브러리인 쓰리제이에스(Three.js)를 사용하여 만들었다. 오브젝트는 버튼을 클릭하여 미리 정해진 기본 값으로 생성하는 방법과 마우스 클릭 후 드래그 하여 생성하는 방법 두 가지를 만들었다.

오브젝트는 기하(geometry) 정보와 재질(material) 정보로 나눌 수 있으며, 두 정보를 이용하여 메쉬 데이터(Mesh Data)를 형성한다. 두 정보는 독립적으로 전달할 수 있으므로, 어떤 꼭짓점(vertex) 좌표를 주냐에 따라 오브젝트의 모양이 결정된다. 오브젝트는 마우스를 통해 좌표 이동, 회전, 크기 변형 등을 할 수 있는데, 이는 쓰리제이에스 라이브러리의 트랜스

폼컨트롤스제이에스(TransformControlls.js)를 이용하여 구현한다. 오브젝트의 선택은 레이 캐스팅(RayCasting) 방식을 사용, 마우스로 클릭을 했을 때, 마우스 벡터의 좌표 값을 향해서 카메라에서 직선(picking ray)을 발사한다. 발사된 선(ray)은 처음 부딪히는 오브젝트의 정보를 찾아 배열에 저장, 필요한 클래스에서 사용하도록 구현하였다.

실시간 인터랙션을 위해서는 클라이언트간의 데이터를 공유해줄 서버가 필요하다. 서버 프로그래밍에는 여러 가지 방법이 있는데, 그 중에서 서버 사이드 언어로 자바스크립트를 기반으로 한 Node.js를 이용하였다. Node.js은 V8 자바스크립트 엔진 위에서 동작하는 이벤트 처리 입출력 프레임워크이며 단일 스레드 이벤트 루프를 통하여 보다 쉽고 효율적으로 처리할 수 있는 장점을 제공한다.

(그림1) 서버와 클라이언트 내부 구조

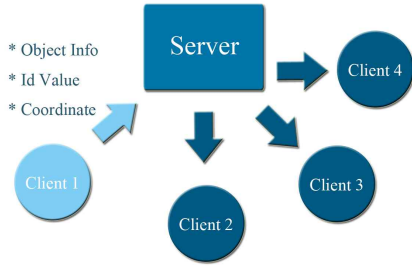


(Figure 1) Server and Client Architecture

서버통신은 기본적으로 한 클라이언트에서 서버에 데이터를 전송하면, 서버가 그 데이터를 받아 다른 클라이언트에게 전달해준다(그림 1). 마찬가지로 한 클라이언트에서 3D 오브젝트를 조작하면, 그 데이터를 서버를 통해 다른 클라이언트들이 공유한다. 본 논문에서 사용하는 Node.js에서는 소켓아이오를 이용하여 데이터 통신을 구현한다. 클라이언트에서 에미트(emit) 메서드를 이용하여 오브젝트의 정보(id값, 크기 정보, 좌표 등)와 실행시킬 메서드 이름을 전달해 주

면, 서버에서 값을 전달받은 해당 메시지를 실행시켜서 다른 서버에게 값을 전달해준다. 실질적으로 서버는 데이터 값만 전달해주고, 오브젝트를 그리는 것은 클라이언트가 담당한다(그림 2). 이와 같은 원리로 다른 클라이언트에서 동일한 화면을 볼 수 있으며, 한 오브젝트를 여러 클라이언트에서 실시간으로 동시 조작하는 것도 가능하다.

(그림2) 서버-클라이언트 데이터 교환 모델



(Figure 2) Server-Client Data Transform Model

#### 4. 시스템 구현

초기 세팅은 쓰리제이에스 API 문서를 참고하여 설정한다. 쓰리제이에스 라이브러리 패키지 파일 중 쓰리제이에스 파일, 트랙볼컨트롤스제이에스(TrackballControls.js) 파일, 트랜스폼컨트롤스제이에스 파일을 헤드 태그에 가져온다. 쓰리제이에스는 오브젝트의 생성 및 제어, 트랙볼컨트롤스는 뷰의 이동을 제어할 때, 마지막으로 트랜스폼컨트롤스제이에스는 오브젝트의 변형을 하는데 사용한다. 바디 태그에는 컨테이너(container)라는 id값을 부여한다.

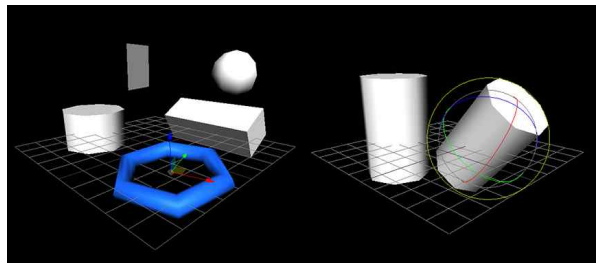
기본적인 변수 설정 방법은 오픈지엘 라이브러리와 유사하다. 본 논문에서는 컨테이너, 카메라(camera), 라이트(light), 씬(scene), 렌더러(renderer) 변수 등을 선언한 후 객체화 시켜준다. 컨테이너 변수는 3D 오브젝트가 구현될 공간을 나타내며, id값을 가져오는 메시지를 사용하여 바디 태그 안에 미리 만들어둔 태그의 id값을 가져와 저장시켜준다. 카메라는 원근 카메라(perspectiveCamera)를 사용하며 위치 좌표는 (1000,500,1000), 룩앳(lookat) 메시지를 사용하여

카메라가(0,200,0)를 향하도록 설정한다. 라이트는 기본설정에는 디렉셔널 라이트(Directional Light)를 사용했다. 씬 구성에 필요한 변수들의 객체화가 끝난 후엔 애드(add) 메시지를 이용하여 씬에 등록한다.

오브젝트는 구(sphere), 육면체(cube), 도넛(torus), 면(plane) 등을 만들 수 있는데, 기본적으로 기하(geometry) 객체와 재질(material) 객체를 만든 후 mesh 메시지를 이용하여 오브젝트를 만든다. 구의 경우 구를 생성하는 기존 메시지를 이용한다. 기하를 생성한 다음에 진행될 것은 매쉬폰(THREEMeshPhongMaterial(color)) 메시드로 재질을 설정하며, 색 값은 제이슨(JSON) 형식으로 16진수 색상 값으로 할당한다. 그 다음 매쉬 함수를 이용하여 기하와 재질 정보를 합쳐 하나의 3D 오브젝트를 만든다. 마지막으로 애드 메시드로 씬에 3D 오브젝트를 등록한다.

오브젝트의 변형(그림 3)은 헤드 태그에 등록된 트랜스폼컨트롤스제이에스 라이브러리를 이용, 키 다운(keydown) 이벤트 리스너를 등록하여 사용한다.THREE.TransformControls(camera, renderer.domElement) 메시드로 초기 설정을 할 때 TFControls 변수에 등록한 후 사용한다.

(그림 3) 오브젝트 변형



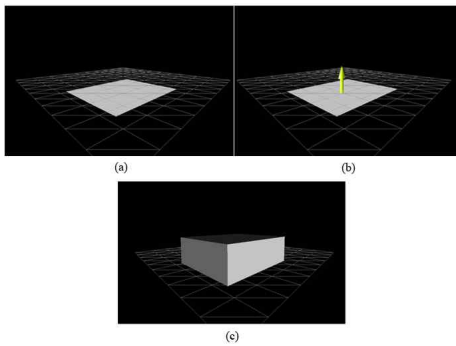
(Figure 3) Transformation of Object

W 키를 눌렀을 때 변형, E 키를 눌렀을 때 회전, R 키를 눌렀을 때 크기를 바꿀 수 있으며, 선택된 오브젝트에 좌표축 가이드라인을 표시하여 이를 이용하여 조작 할 수 있다. 어태치 오브젝트(attach(object)) 메시지를 이용하여 원하는 오브젝트에 가이드라인을 붙여 사용한다.

밀어내기 기능은 2단계로 나뉘어서 적용된다. 첫 번째 단계에서 마우스 드래그 이벤트가 처음

발생한 경우라면 오브젝트의 밑면에 대한 가로, 세로 정보로 인식하여 해당 정보를 이용하여 오브젝트의 밑면을 그린다. 가로, 세로 데이터의 길이는 마우스 업, 마우스 다운 이벤트, 레이 피킹(Ray Picking)을 이용하여 데이터의 변화량으로 계산한다. 첫 번째 단계에서 마우스 드래그 이벤트가 두 번째 발생한 것으로 인식되면 두 번째 단계로 이동한다. 두 번째 단계에서 받아들이는 정보는 높이 정보로 인식한다. 따라서 두 번째 단계에서 받은 높이 정보를 이용하여 첫 번째 단계에서 그린 밑면 위에 해당하는 높이를 이용하여 오브젝트를 완성한다(그림 4).

(그림 4) 오브젝트 밀어내기



(Figure 4) Extrude of Object

이러한 밀어내기 기능은 체크박스의 유무에 따라 버튼을 눌렀을 때 오브젝트가 바로 나오는 경우가 있고 마우스 드래그를 통해 오브젝트를 그리는 경우가 있다.

선택기능(select)의 구현은 레이 피킹 방식을 이용했다. 컨테이너 안을 클릭했을 때, 마우스 벡터 x, y값을 계산한다.

그 다음으로는 레이 캐스터(raycaster) 변수를 만든 다음에 바로 projector.pickingRay() 메서드를 이용하여 레이를 할당한다. 레이에 부딪힌 것은 raycaster.intersectObjects() 메서드로 받아 인터섹트(intersects)변수에 저장한다. 마지막으로 포문(for) 또는 와일(while)문을 이용하여 첫 번째 부딪힌 객체를 구한다(그림5).

(그림 5) 마우스 벡터와 레이 피킹 구현

```
mouseVector.x = 2 * (e.clientX / window.innerWidth) - 1;
mouseVector.y = 1 - 2 * (e.clientY / window.innerHeight);

var raycaster = projector.pickingRay(mouseVector.clone(), camera),
    intersects = raycaster.intersectObjects(objects.children);
```

(figure 5) Mouse Vector and RayPicking Realization

클라이언트 간의 데이터 교환은 소켓 통신과 이벤트 구동형 (event-driven) 방식을 이용한다. 소켓 온(socket.on()) 메서드를 이용하여 소켓 이벤트를 연결하고, 소켓 에밋(socket.emit()) 메서드를 이용하여 소켓 이벤트를 발생시킨다. client에서 이벤트 리스너를 등록한 후, 이벤트가 발생했을 당시에 사용하는 메서드로는 socket.emit ('ServerEventName')을 실행시킨다. 이를 통하여 서버에 소켓 온 메서드를 이용하여 등록된 동일한 이름(ServerEventName)을 가진 메서드가 실행된다. 실행된 메서드 안의 io.socket.emit ('ClientEventName') 메서드가 클라이언트에 특정 이름(ClientEventName)을 가진 메서드를 실행시킨다. 이런 방식으로 서버-클라이언트 간 데이터 교환이 이루어진다.

하나의 클라이언트에서 3D 오브젝트를 만들면, 그 오브젝트의 정보(반지름, x, y 좌표, 색 등)를 에밋 메서드를 이용하여 서버로 전달한다. 서버에서는 io.socket.emit 메서드를 이용하여 다른 클라이언트에게 그 데이터 값들을 전달한다. 데이터 값은 제이슨 형식으로 전달 할 수 있으며, 이와 같은 방식으로 한 클라이언트에서 생성한 객체를 모든 클라이언트에서 공유 할 수 있다. (그림 6)는 구를 만들 때 서버-클라이언트 간 코드를 나타낸 것이다.

(그림 6) 서버와 클라이언트 간 통신

```

// client code
socket.emit('draw', {
  x: x, y: y, z: z, // position
  color: 0xff0000,
  radius: radius
});

// server code
socket.on('draw', function (data) {
  io.sockets.emit('line', data);
});

// client code
socket.on('line', function (data) {
  // geometry and material object create
  var geometry = new THREE.SphereGeometry(data.radius);
  var Material = new THREE.MeshPhongMaterial({
    color: data.color
  });

  // mesh object create
  var mesh = new THREE.Mesh(geometry, material);
  mesh.position.set(data.x, data.y, data.z);
  scene.add(mesh);
});

```

(Figure 6) Server-Client Socket Communication Code

### 5. 실험 및 고찰

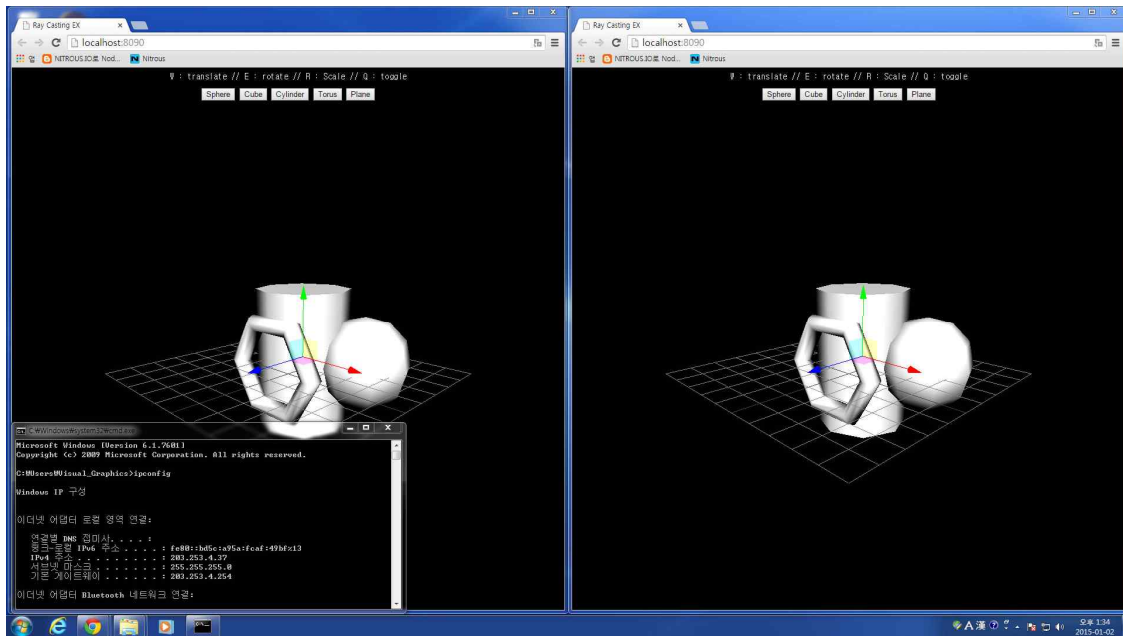
Node.js를 이용하여 컴퓨터에 서버를 구축한 후, 앞에서 언급했던 내용을 구현하였다. 먼저 로컬 호스트 내에서 제대로 작동이 되는지 점검

하였다. 로컬 호스트로 서버에 접속하여 두 개의 웹 브라우저를 실행하고 하나의 브라우저에서 조작을 하면 다른 브라우저에서도 변경된 내용이 실시간으로 반영되는지 확인하였다.(그림 7).

다음으로 외부 컴퓨터에서 서버가 설치된 컴퓨터에 접속하여 외부 컴퓨터에서도 오브젝트 조작을 할 수 있음을 확인하고, 변경된 내용을 실시간으로 다른 컴퓨터와 공유하고 있다. 만약 외부 컴퓨터에서 구를 추가했다면 서버에 접속된 모든 컴퓨터에서 구가 추가된 것을 확인 할 수 있다. 외부 컴퓨터에서는 같은 오브젝트 정보를 마우스를 이용하여 카메라 위치에 따라 다른 각도에서 볼 수도 있으며, 서버 컴퓨터와 외부 컴퓨터에서 동시에 조작도 가능함을 확인하였다(그림 8).

본 논문에서 제안한 시스템을 이용하면 실시간으로 오브젝트를 변형해가며 의견을 나눌 수 있기 때문에 좀 더 시각적이고 상호간의 소통이 가능하다. 협업할 때 자신이 만든 3D 모델을 다른 사람에게 보여주려면 스크린 샷을 찍거나, 프로그램에서 내보내기(Export)를 해준 파일을 보내주는 방법이 있다. 전자의 경우 바로 보여줄 수 있다는 장점이 있지만 오브젝트의 변형이 불

(그림 7) 로컬 호스트 내에서 조작

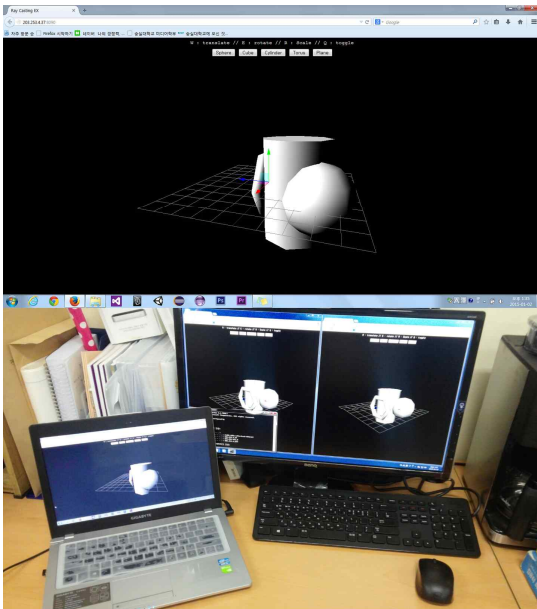


(Figure 7) Control in the Local Host Area



가능하다. 후자의 경우 반대로, 오브젝트의 변형을 통해 의견을 좀 더 자세하게 표현할 수 있지만, 툴이나 뷰어 등 프로그램 없이는 볼 없다는 단점이 있다. 따라서 본 연구에서는 여러 번 수정을 하여 파일을 보낼 필요가 없으므로, 작업 시간을 단축하여 효율성 높은 작업이 가능하다. 또한 웹 브라우저를 이용하기 때문에, 웹 브라우저가 웹지엘만 지원해준다면 OS에 종속되지 않고 간편하고 쉽게 사용할 수 있다. 현재 파이어폭스, 크롬, 인터넷 익스플로러 외에도 대다수의 브라우저가 웹지엘을 지원해주고 있다. 따라서 거의 대부분의 웹 브라우저에서 접근이 가능하다. 실제로 물리적으로 멀리 떨어진 장소에서 모델링을 해본 결과, 실시간으로 의견을 나누면서 모델링 작업하니 적은 시간 안에 서로가 원하는 결과물을 만들어 낼 수 있었다. 또 전용 프로그램을 설치하는 번거로움을 줄일 수 있고 내보내기하여 서로 주고받는 시간을 줄여 작업 시간을 크게 단축시킬 수 있었다.

(그림 8) 외부 컴퓨터를 이용한 조작



(Figure 8) Control using External PC

## 6. 결론

본 연구에서는 여러 사람들이 동시에 상호작

용이 가능한 웹 환경을 제작하여 기존의 폐쇄적인 모델링 환경을 벗어난 실시간 협업 모델을 제안하였으며 실시간 피드백이 가능하고 3D 모델링 파일을 변환하거나 따로 내보내기하여 다른 사람들에게 보내야 하는 번거로운 작업을 줄임으로써 3D 모델링 작업 환경을 개선 할 수 있음을 보였다. 본 시스템은 5가지 기본 오브젝트와 밀어내기 기능을 제공하고 있으며 개별 컴퓨터에서 여러 가지 뷰를 자유롭게 볼 수 있다. 본 논문에서 제시한 예시 모델링은 비교적 단순한 모델에 대해 테스트하였는데, 보다 세밀하고 복잡한 모델로의 확장은 기술적으로 차이가 없으므로 실제 현장에서는 복잡한 모델에 충분히 적용될 수 있으며 모델링 단계별 작업 내역을 히스토리로 저장하는 기능이 추가된다면 사용자가 원하는 단계별 진행사항을 확인 할 수 있을 것이다. 또한 사용자가 3D 모델링 전용 소프트웨어에서 작업한 파일을 바로 웹 브라우저에서 볼 수 있도록 여러 가지 파일 형식을 지원하는 가져오기 기능과 웹 브라우저에서의 협업 후 수정된 3D 모델링을 바로 내보내기하여 다시 개별 작업을 할 수 있도록 한다면 더욱 개선된 서비스를 제공 할 수 있을 것이다.

## References

- [1] T. McLaughlin, B. Ad’an Pe’na, T. A. Fechter, A. M. Pasing, J. Reitz, and J. A. Vidal, “Multi-institutional Collaboration in Delivery of Team-Project-Based Computer Graphics Studio Courses”, *Advances in Visual Computing*, pp.394-405, 2010.
- [2] K. McHenry, P. Bajcsy “An Overview of 3D Data Content, File Formats and Viewers”*Image Spatial Data Analysis Group, National Center for Supercomputing Applications, 1205 W Clark, Urbana, IL 61801, October 31, 2008*
- [3] M. Milivojević, I. Antolović, and D. Rančić “Evaluation and visualization of 3D models using Collada Parser and WebGL technology”, *Proceedings of the 2011 international conference on Computer and computing*, 2011.
- [4] W. Wu, Z. Yang, K. Nahrstedt, G. Kurillo, and R.

Bajcsy, "Towards Multi-Site Collaboration in Tele-Immersive Environments", Distributed Computing Systems, ICDCS '08. The 28th International Conference, pp.647 - 654,17-20, June 2008.

[5] R. Vasudevan , Z. Zhou, G. Kurillo, E. Lobaton, R. Bajcsy, and K. Nahrstedt, "Real-time Stereo-vision System for 3D Telemmersive Collaboration", Multi media and Expo (ICME), IEEE International Conference, pp.1208-1213, July 2010

[6] R. Bajcsy, G. Kamberova, and L. Nocera "3D Reconstruction of Environments for Virtual Collaboration" In Proc. of the 4th IEEE Workshop on Applications of Computer Vision, 2000.

[7] S. Birr1, J. M'onch1, D. Sommerfeld1, and B. Preim, "A novel Real-Time Web3D Surgical Teaching Tool based on WebGL", Bildverarbeitung für die Medizin, pp.404-409, 2012

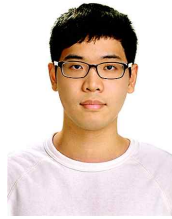
[8] C. Leung, A. Salga, "Enabling WebGL", WWW '10 Proceedings of the 19th international conference on World wide web, pp.1369-1370, 2010.

[9] J. O. Talton, D. Fitzpatrick "Teaching Graphics with the OpenGL Shading Language", ACM SIGCSE Bulletin archive, 39(1), Mar. 2007

[10] C. Leung, A. Salga "Enabling WebGL" Proceedings of the 19th international conference on World wide web, Raleigh, North Carolina, USA 2010

[11] S. H. Lee, Y. H. Park, H. W. Park, "Time and Space Modeling Method for Social Services", Journal of digital contents society, vol. 11, no. 4, pp. 571-578, 2010.

### 편 해 결



2010년 : 숭실대학교 글로벌미디어 학사

관심분야 : 컴퓨터 그래픽스, 웹 프로그래밍

### 안 향 아



2012년 : 숭실대학교 글로벌미디어 학사

관심분야 : 컴퓨터 그래픽스, 3D 모델링

### 박 진 호



1999년 : KAIST 수학과 졸업 (이학사)

2001년 : KAIST 응용수학과 졸업(이학석사)

2007년 : KAIST 전산학과 졸업 (공학박사)

2007년~2008년: KAIST 문화기술 대학원 박사 후 연구원

2008년~2009년: UCLA 전산학과 박사후 연구원

2009년~2013년: 남서울대학교 멀티미디어학과 조교수

2013년~현재: 숭실대학교 글로벌미디어학부 조교수

관심분야 : 컴퓨터 그래픽스, 물리 기반 애니메이션