

웹 기반 가상현실 안내 시스템의 기초 구현

이신결 · 최보성 · 전희성
컴퓨터 · 정보통신 공학부

<요 약>

본 논문에서는 인터넷에서 가상세계를 구현하는 표준 언어인 VRML(Virtual Reality Modeling Language)을 사용하여 가상 세계를 구축하고 이를 안내시스템에 적용하는 방법을 제시하였다. 이를 위해 먼저 VRML 규약에 대해서 고찰을 하고, VRML과 JavaScript를 이용한 웹 기반 가상 안내 시스템의 구현 방법에 대해 상세히 설명하였다. 그리고, 시스템의 구현을 통해 얻어낼 수 있는 효과와 구현 과정에 발생된 문제점과 개선방법 및 앞으로의 연구방향에 대해 기술하였다.

Implementation of a Prototype Virtual Reality Information Desk based on the Web

Shingeol Lee · Bosung Choi · Heesung Jun
School of Computer Engineering & Information Technology

<Abstract>

In this paper, we propose a method of implementing a virtual reality information desk using VRML and Internet. We review about VRML which can construct a virtual world on the Internet. Next, we explain a basic concept and implementation details of a web-based virtual reality information desk. VRML and JavaScript are used to implement the system. Finally, the advantages of the developed system and the future research directions are described also.

1. 서 론

인터넷에서의 가상현실기술의 도입은 그 효용가치로 인해 많은 연구가 진행 중이다. 이것은 가상현실이 인터넷과 연동되어 사용자에게 다양하고 더욱 상호작용적인 정보를 제공할 수 있기 때문이다. 본 연구는 울산대학교의 안내 시스템에 최근 들어 각광 받고 있는 가상현실 기술을 접목하여 더욱 편리하고 다양한 정보를 제공하는 가상 현실 기반 안내시스템의 제작에 대한 기초를 제공하는 데 역점을 둔다.

먼저, 2절에서는 현재 인터넷에서의 가상 현실 구현을 위한 VRML(Virtual Reality Modeling Language)의 연구 동향에 대해서 알아보고, 가상현실 기반 안내 시스템을 구축하기 위한 VRML 규약에 대해서 고찰하였다. 3절에서는 안내시스템의 기초적인 설계 및 구현 방법에 대해 기술하였고 시스템의 구현 시 발생된 문제점 및 그의 개선 방법에 대해서 기술하였다. 마지막으로 결론 부분에서는 본 연구의 성과와 앞으로의 연구 과제에 대해서 기술하였다.

2. VRML

2.1 VRML의 역사와 특징

VRML은 인터넷에서 3차원 가상 공간의 장면을 정의하는 언어이다. 1994년도에 제네바에서 개최된 W3 국제회의에서 Mark Pesce가 "Cyberspace"라는 논문을 발표하였다. 그 논문에 Labyrinth라는 VRML 브라우저의 prototype이 등장하면서 웹에서의 3차원 세계의 구성에 대한 관심이 고조되었다. Labyrinth는 웹에서 3차원 세계를 렌더링(rendering)할 수 있는 첫 번째 인터페이스라는 점에서 관심의 대상이 되었다. VRML 관련 논의를 위해 메일링 리스트(mailing list)[1]가 만들어 졌고 이를 통해 VRML에 정의될 요소와 규약에 대한 논의가 진행되었다. 이러한 메일링 리스트를 통해 실리콘 그래픽스(Silicon Graphics)의 프로그래머인 Gavin Bell이 공식적인 VRML 파일 포맷(file format)에 Open Inventor를 적용하는 것을 제안하였고, 그 제안이 설득력 있게 받아들여지게 되었다. VRML만을 위한 새로운 그래픽 파일 포맷을 만들어 내는 것보다 현존하는 그래픽 파일 포맷을 통해 규약을 만들어 내는 것이 더 좋은 방법이라 여겨졌고, 여러 가지 파일 포맷이 제안되었으나, 최종적으로 SGI의 Open Inventor를 기반으로 한 VRML 1.0 규약이 최종적으로 결정되었다. 당시 발표된 VRML 1.0 규약은 단순히 가상 공간을 모델링(modeling)하는 기술에 대해서만 정의되었을 뿐, 가상 공간 내에서 참여자의 상호작용적인 활동을 구현하는 데에는 한계가 있었다. 1995년 8월에 www-vrml 메일링 리스트 중에서 가상 현실 및 인터넷 관련 회사의 선도적인 역할을 하는 사람들로 구성된 VAG(VRML Architecture Group)가 결성되었고, 이후 논의되는 VRML에 대한 제안과 개선사항을 제정하는 기구로 설립되었다. 이후 VRML 1.0 규약은 수정을 거듭하였는데, 가상 환경에서 가상 공간내의 사용자의 투영인 아바타(avatar)[2]를 통한 상호작용의 개념을 도입한 VRML+와 여러 종류의 문자, 음성, 기하학적인 확장을 제공하는 VRML 1.1 등이 등장하였다. 그리고 1995년 10월, VAG는 더욱 개선된 기능을 제공하는 새로운 VRML 규약을 공모하였고, 1996년 8월에 실리콘 그

래픽스에서 제안한 Moving World로 불리는 VRML 2.0 규약을 채택하여 발표하였다. 이러한 VRML 2.0은 VRML 1.0 규약에서 보다 발전된 개념을 채택하였는데, 그 특징은 다음과 같다[3].

- 정적 가상공간의 확장
- 상호작용
- 애니메이션과 behavior scripting
- 사용자의 정의를 통해 새로운 VRML object를 정의할 수 있는 prototyping

이상과 같은 VRML 2.0의 특징은 다음과 같은 의미를 지닌다. VRML 1.0의 정적 가상 공간 구현에서 더욱 발전한 비디오와 오디오 등의 멀티미디어 자원의 사용이 가능하게 되었고 가상공간을 더욱 세밀하고 화려하게 구성할 수 있는 elevation grid, background, fog, extrusion 등의 노드(node)가 추가로 정의되었으며 또한, 가상 현실에서 sensor node를 통하여 참여자와 상호작용이 가능하게 되었다. 그리고, interpolator node를 통한 keyframe 애니메이션으로 가상 세계에서 애니메이션이 가능하게 되었으며, scripts(Netscape's JavaScript 또는 Sun Microsystem의 Java)를 통한 복잡한 형태의 시뮬레이션이 가능하게 되었다. 그리고, VRML에서 제공되는 노드라 불리는 일정한 형태의 객체를 더욱 확장하여 사용자가 노드의 형태를 직접 정의하여 사용할 수 있는 prototyping이 추가되었다.

2.2 Network에서의 VRML 전송 과정 및 VRML의 구조

클라이언트(client) 컴퓨터가 서버(server)에 있는 특정 VRML 파일을 요청하면 서버는 그 VRML 파일을 클라이언트에게 전송하고 그 파일이 클라이언트 컴퓨터의 웹 브라우저(web browser)의 VRML 플러그인(Plug-in) 프로그램에 의해 클라이언트 컴퓨터에서 렌더링 된다[그림 1].

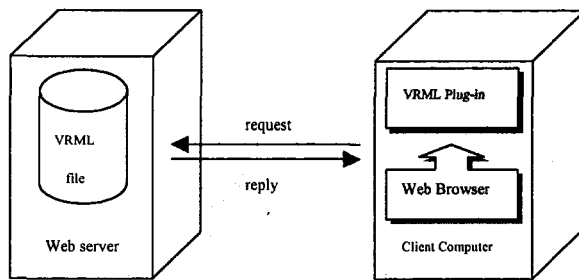


그림 1. VRML의 서버-클라이언트 구조

VRML 파일을 전송하기 위해서는 새로운 형식의 MIME(Multipurpose Internet Mail Extensions) 타입이 정의되어야 한다. MIME 타입은 인터넷을 통하여 전송되는 파일의 내용을 정의하여, 각 웹 브라우저가 어떠한 방법을 통해 파일을 화면에 표시할 것인지를 결정할 수 있게 해준다. 즉, VRML파일은 확장명이 wrl이고, 이러한 형태의 파일이 웹 서버에 요청되거나 클라이언트 컴퓨터에 전송되어 웹 브라우저에 의해 읽히면 VRML 파일임을

인식할 수 있게 된다. 이러한 과정이 수행되기 위해서는 웹 서버와 클라이언트 컴퓨터의 웹 브라우저에 VRML의 MIME 타입이 정의되어 파일 타입의 매핑(mapping)이 이루어져야 한다.

VRML은 아래와 같은 기본적인 형태를 가지고 있다.

```
#VRML V2.0 utf8
Node {
  field value
}
```

모든 VRML 2.0 파일은 VRML 파일임과 VRML 파일의 버전(version), 그리고 ISO 10646 표준의 UTF-8 인코딩(encoding)을 사용함을 알려주는 첫 번째 행으로 시작된다. 노드는 크게 Grouping nodes, Common nodes, Sensor nodes, Geometry nodes, Interpolator nodes, Bindable nodes로 나뉘고, 이러한 노드의 reference는 Public interface로 VRML 2.0 규약에 정의되어 있다[4].

각 노드는 세부적인 표현을 위한 field와 그에 따른 value를 가지고 있으며, 이러한 field와 value를 통해 각 node의 속성을 변경하여 가상 세계를 만들게 된다.

3. 가상 현실 안내 시스템의 구현

3.1 가상 현실 안내 시스템의 기본 개념

현재 실험적으로 구현된 가상 캠퍼스의 건물은 본관, 상정탑, 7호관 및 8호관 이다. 그리고, 가상 안내 시스템을 만들기 위한 기본 개념은 HTML로 구현된 홈페이지와 VRML로 제작된 가상 캠퍼스를 함께 구현하여 안내시스템을 제작하고, 이들 간에 정보를 주고받을 수 있는 방법을 정의함으로써 가상 현실 등의 더욱 다양한 정보를 제공할 수 있는 안내시스템을 만드는 것을 목표로 하고 있다.

가상 현실 안내 시스템을 위한 구조는 크게 다음과 같은 부분을 가지고 있다[그림 2].

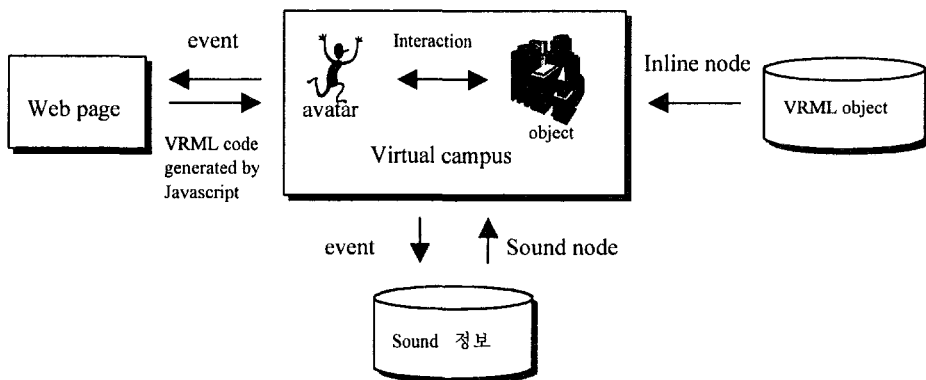


그림 2. 가상 현실 안내 시스템의 구성

- 가상 공간을 구성하기 위한 가상 건물과 물체
- 안내 정보를 위한 음성 정보
- 가상 현실 정보를 추가로 보조하기 위한 웹 페이지를 통한 정보
- 가상 공간 내에서의 물체와 아바타의 상호작용에 대한 구현
- 상호 작용에 의한 이벤트(event)의 처리
- 웹 페이지에서 사용자 선택에 의한 가상 공간의 변화

이러한 안내시스템을 구현하기 위하여 각 구조에 대한 구현은 다음과 같이 이루어 졌다.

3.2 가상 캠퍼스내의 물체 제작

현재 구현된 가상 건물 및 가상 물체는 그림 3에 보인 바와 같이 본관, 7호관, 8호관, 상징탑, 안내 지도판과 나무 등으로 구성되어 있다.

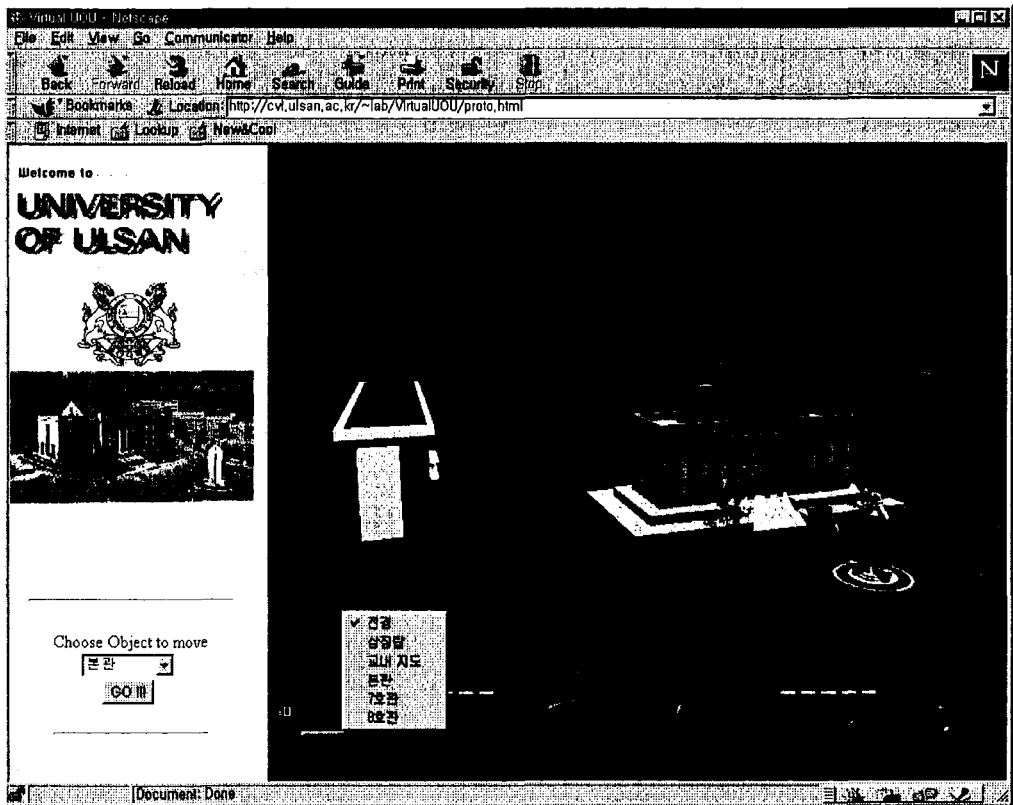
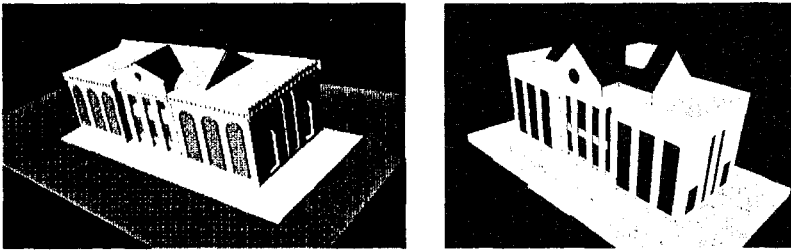


그림 3. 가상 현실 기반 안내 시스템

가상 캠퍼스내의 물체인 본관, 7호관, 8호관, 상징탑은 3차원 모델링 프로그램인 3D Studio MAX를 사용하여 제작한 후, 이것을 Kinetix사에서 제작된 3D Studio MAX 플러그인 프로그램을 사용하여 VRML 1.0 파일로 변환하였다. 이는 VRML에서 제공되는 물체

생성 노드의 사용만으로는 물체를 세밀히 구현하는 데 한계가 있기 때문이다.

연구를 수행하면서 VRML 물체를 제작하는 과정에서 각 물체를 사실감 있게 묘사하기 위해 물체의 세밀한 부분까지 3D Studio MAX를 통하여 제작하였으나 이는 가상 물체의 파일 크기를 크게 하여 결과적으로 인터넷에서의 전송시간을 지연시키는 결과를 가져왔다. 즉, 이러한 문제를 해결하기 위하여 물체의 폴리곤(polygon) 수를 줄이거나 복잡한 폴리곤을 대신하여 텍스처 매핑(texture mapping)을 사용해 파일 크기를 줄이는 방법을 택할 수 있는데, 폴리곤 수를 줄이는 데는 한계가 있으므로 최적의 폴리곤 수와 텍스처 매핑을 사용하여 물체를 제작하였다. 그림 4의 (a)는 연구 초기에 제작된 것으로서 텍스처 매핑을 사용하지 않고 3D Studio MAX를 사용하여 제작한 본관을 VRML 파일로 변환한 본관의 가상현실 물체이고, 그림 4의 (b)는 텍스처 매핑을 사용한 물체이다. 이번 연구에서는 Pentium 200MMX CPU와 32M bytes Main memory, 그리고 Windows 95 운영체제를 사용하는 시스템을 사용하였는데, 그림 4의 두 물체가 렌더링 되는 시간은 각각 25초와 17초가 소요되었다. 그리하여 이번 연구에서 제작된 각 물체는 3D Studio MAX에서 제작될 때 최소한의 폴리곤으로 제작되었고, 이것은 실제 인터넷을 통해 가상 캠퍼스 파일이 전송되는 시간을 단축시켜 주었다.



(a) 3D Studio MAX로만 구현된 object (b) 3D Studio MAX 구현과 texture mapping에 의한 object

그림 4. 두 가지 구현 방법에 의한 본관의 가상 현실 모습

가상 현실내의 안내 지도판의 제작은 VRML 2.0 의 Shape 노드를 사용하여 제작되었고, 지도는 실제 본관 앞에 있는 안내 지도를 디지털 카메라로 캡처(capture)하여 제작된 안내 지도판에 텍스처 매핑을 하였다.

가상 공간의 물체는 인터넷을 통하여 서로 공유하는 추세로 발전하고 있으므로, 가상 공간을 만들기 위한 물체는 인터넷의 여러 곳에서 획득할 수 있다. 본 가상 캠퍼스내의 나무도 Orc inc.의 가상 물체 라이브러리[5]를 통하여 획득하였으며, 이것을 Inline 노드를 통해 가상 공간 내에 배치하였다. 이와 같이 인터넷을 통한 가상 물체의 획득과 사용은 VRML inline 노드를 통해서 가능하며 이 노드의 사용법은 아래와 같다.

```

Inline {
  url "http://www.ocnus.com/models/Plants/pinetree.wrl.gz"
  bboxCenter 0 0 0
  bboxSize -1 -1 -1
}

```

위에서 url 필드(field)는 인터넷에서의 가상 현실 물체의 URL(Uniform Resource Location)을 나타내며 인터넷의 가상 현실 물체의 위치를 기술함으로써 자신의 가상 세계에 물체를 포함시킬 수 있다.

3.3 추가 안내 정보를 위한 sound 정보의 제작 및 가상 현실 내의 적용

가상 현실 내에서의 다양한 멀티미디어 정보를 포함하여 참여자에게 제공할 수 있는 기능이 VRML 2.0에 제공되어 있다. 본 연구에서는 더욱 효율적인 정보 전달을 위해 가상 현실 내에서 각 건물을 아바타가 선택하게 되면 그 건물에 대한 정보가 음성으로 표현될 수 있도록 구현하였다. 이를 위해 각 건물의 설명을 음성 파일로 제작하였고, 이를 아바타가 가상 건물을 선택하여 이벤트를 발생시킬 때 음성 파일을 재생하도록 구현하였다. 이를 위해 각 건물에 대한 음성 파일을 Group 노드로 제작하여 건물을 아바타가 누를 때 발생된 이벤트를 sound group으로 보내어 음성 정보를 표현할 수 있도록 하였다[그림 5].

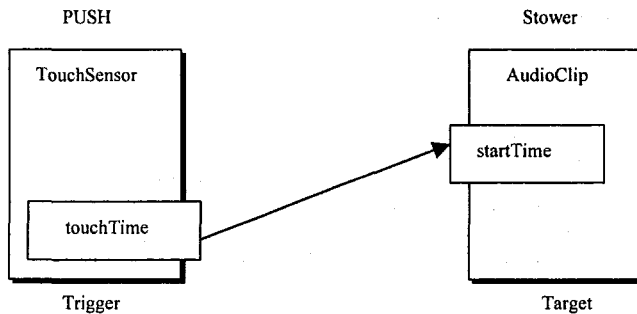


그림 5. 음성 정보의 재생을 위한 이벤트의 흐름

3.4 가상 현실 정보를 추가로 보조하기 위한 웹 페이지를 통한 정보

구현된 가상 캠퍼스는 상호작용적이고 다양한 정보를 제공하기 위하여 웹 페이지와 연동된 구조를 취하였다[그림 3].

먼저, HTML 태그(tag)를 사용하여 두 개의 윈도우를 가진 웹 페이지를 만들고, 각 윈도우는 HTML 파일과 VRML 파일을 읽게 했다. 이러한 구조를 구현한 것은 가상 현실에서 부족한 문자 정보를 HTML 파일에서 보여주기 위한 것이다. 이러한 구조를 위해서 가상 캠퍼스에서 발생된 이벤트(가상 세계에서 정보를 알고자 하는 물체로 마우스 커서를 이동해 마우스 버튼을 눌렀을 때 발생된 이벤트)를 어떻게 HTML 파일에게 보내어 웹 페이지에서 내용을 변경하는 가에 대한 구조가 정의되어야 한다.

이것은 아래의 VRML 코드와 같이 VRML의 Anchor 노드에서 파라미터(parameter) 필드를 사용함으로써 하나의 프레임을 변경하기 위해 필요한 HTML 코드의 파라미터를 전송할 수 있다. 아래 코드는 Anchor 노드에서 이벤트를 발생시키기 위해 삽입된 Sensor 노드를 보이고 있고, 그림 6은 가상 캠퍼스와 웹 페이지간의 이벤트와 파라미터의 흐름을 보인 그림이다. 그리고, 그림 7은 가상 세계내의 아바타가 건물을 선택하여 발생된 이벤트로 웹 페이지의 내용이 변경된 것을 보이고 있다.

```

Anchor {
  children [
    DEF TOWER Transform {
      translation 0 0 -130
      scale 0.55 0.55 0.55
      children Inline {
        url "Tower.wrl"
        bboxCenter -1 -1 -1
        bboxSize 0 0 0
      }
    },
    DEF PUSH1 TouchSensor { }
  ]
  bboxCenter 0 0 0
  bboxSize -1 -1 -1
  url "Tower.html"
  description "상징탑설명"
  parameter "target=Contents"
}
    
```

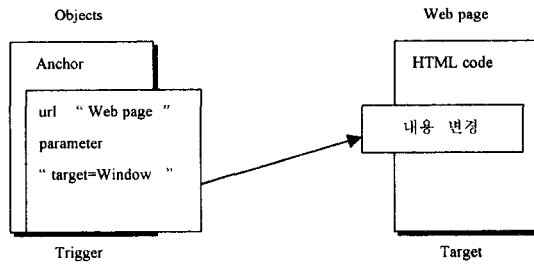


그림 6. 가상 캠퍼스와 웹 페이지간의 이벤트와 파라미터의 흐름

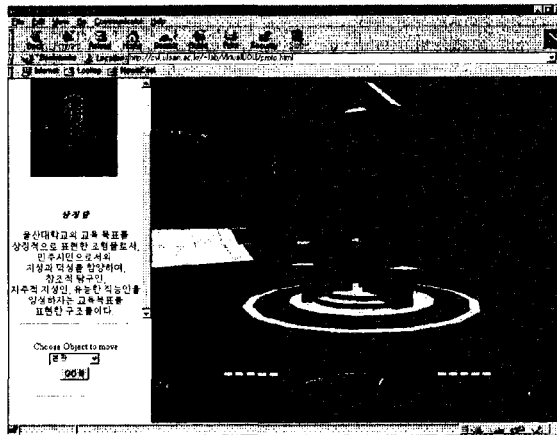


그림 7. 가상 환경 내에서 아바타의 선택에 의한 웹 페이지 변경

3.5 가상 현실의 세부적 구성을 위한 Common 노드와 Bindable 노드의 구성

가상 현실 내의 물체를 사용자에게 보여 주기 위해서는 아래와 같은 몇 가지의 노드가 정의되어야 한다.

- 가상 세계를 조명하는 광원의 정의
- 가상 세계 내의 물체를 관찰하기 위한 시점의 정의
- 가상 세계 내의 아바타의 정의

먼저, 광원을 배치하기 위해서 PointLight 노드를 사용하였는데, 가상 캠퍼스내의 모든 물체를 여러 곳의 시점에서 볼 수 있도록 하기 위하여 4 개의 점 광원을 두었다. 현실에서 태양 광에 의한 효과와 유사하게 하기 위하여 PointLight의 위치를 가상 공간내의 높은 지점에 위치시켰고, 각 광원의 세기 조절과 빛이 미치는 범위를 넓게 하기 위하여 radius 필드를 최대로 설정하였다. 아래 코드에서 radius, intensity, ambientIntensity, color 필드는 중복되므로 이하 생략하였다.

```
DEF LIGHTS Group {
  children [
    DEF SUN1 PointLight {
      on TRUE
      location -300 700 200
      radius 900
      intensity 0.3
      ambientIntensity 0.2
      color 1 1 1
    },
    DEF SUN2 PointLight {
      on TRUE
      location 300 700 200
    },
    DEF SUN3 PointLight {
      location -300 700 700
    },
    DEF SUN4 PointLight {
      location 300 700 -700
    }
  ]
}
```

다음은 각 물체를 관찰하기 쉬운 지점에 시점을 설치하고, 6개의 ViewPoint 그룹을 구성하였다. 이것은 CosmoPlayer의 control panel의 viewpoint 리스트를 통해 이동하면서 관찰할 수 있도록 하였다.

```

DEF VIEWS Group {
  children [
    Transform {
      rotation 0 1 0 -0.35
      children Viewpoint {
        description "전경"
        position -130 90 80
        orientation 1 0 0 -0.15
        fieldOfView 0.785398
        jump TRUE
      }
    },
    Viewpoint {
      description "상징탑"
      position -45 10 -60
      orientation 0 1 0 -0.5
      fieldOfView 0.785398
      jump TRUE
    },
    Viewpoint {
      description "교내 지도"
      position 14 10 -166
      orientation 0 0 1 0
      fieldOfView 0.785398
      jump TRUE
    },
    Viewpoint {
      description "본관"
      position -70 10 -120
      orientation 0 1 0 -0.5
      fieldOfView 0.785398
      jump TRUE
    },
    Transform {
      rotation 1 0 0 -0.05
      children Viewpoint {
        description "7호관"
        position 70 70 -200
        orientation 0 1 0 -0.6
        fieldOfView 0.785398
        jump TRUE
      }
    }
  ]
}

```

```

    }
  },
  Transform {
    rotation 1 0 0 -0.05
    children Viewpoint {
      description "8호관"
      position -70 70 -200
      orientation 0 1 0 0.6
      fieldOfView 0.785398
      jump TRUE
    }
  }
]
}

```

그리고, 아바타 크기와 관찰자가 가상 현실을 살펴보는 환경을 정의하기 위해서 NavigationInfo 노드를 다음과 같이 정의하였다.

```

NavigationInfo {
  type "FLY"
  speed 0.5
  avatarSize [0.25, 1.6, 0.75]
  headlight FALSE
  visibilityLimit 0.0
}

```

위에서 type 필드의 정의는 관찰자의 시점을 자유롭게 하기 위하여 "FLY"로 지정함으로써 관찰자의 가상 세계의 어느 곳에서나 관찰을 할 수 있도록 하였고, 아바타의 크기를 가상 세계와 비교하여 인체의 크기와 유사한 비율로 조정하였다. 이것은 가상 세계를 이동하면서 각 물체와의 충돌을 검출하는 데 사용되는 정보이고, 충돌로 인한 이벤트의 발생이나 물체와의 상호작용을 일으키는 중요한 정보가 된다. 그리고 headlight 필드는 가상 세계의 아바타가 이동하면서 정면에 보이는 물체에 조명을 비치는 효과를 조정하는 필드인데 가상 세계 내에 광원을 지정하여 두었기 때문에 값을 "FALSE"로 지정하였다.

3.6 JavaScript를 통한 VRML 코드의 생성

HTML과 JavaScript를 이용한 변경된 VRML 코드의 생성을 통해 사용자에게 더욱 상호작용적인 시스템을 만들 수 있다. 웹 페이지에서 사용자가 안내를 원하는 건물을 선택하면, viewpoint behavior scripting이 변화된 새로운 VRML 코드를 생성한다[그림 2]. 이는 사용자의 시점을 가상 세계 내에서 변화시켜 마치 사용자가 가상 환경 내에서 움직이는 것과 같이 해주어 더욱 효과적인 안내시스템이 될 수 있게 하였다[그림 8].

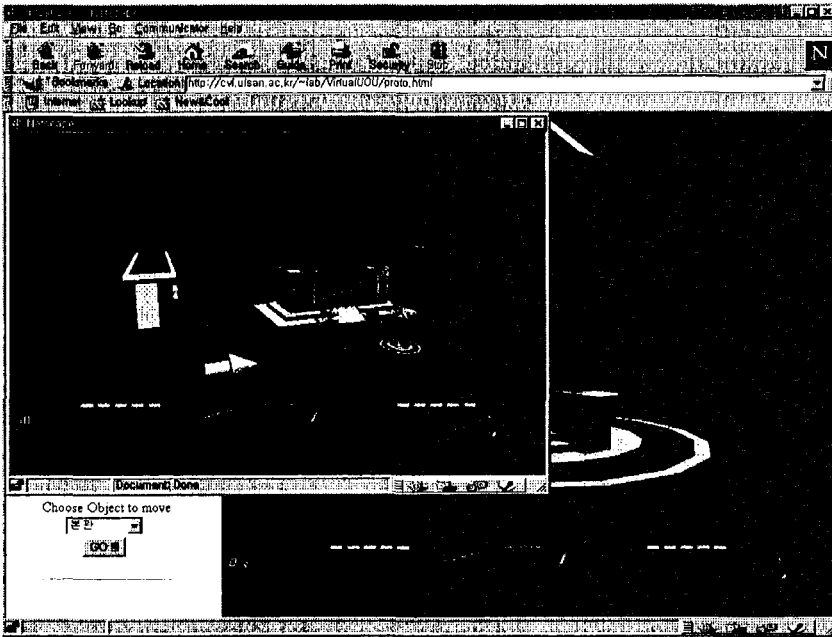


그림 8. Viewpoint behavior scripting이 변경된 새로운 VRML 코드에 의한 윈도우

3.7 안내시스템의 사용자 인터페이스

본 연구에 사용된 웹 브라우저는 Netscape Communicator 4.04 [6]이고 Cosmo Player 2.0[7]이 VRML 플러그인 프로그램으로 사용되었다.

본 연구에서 가상 현실로 구현된 안내시스템은 6개의 시점을 가지고 있다. 그리고, 6개의 시점사이의 전환은 Cosmo Player의 Viewpoint List에서 원하는 시점을 선택함으로써 이루어진다. 또한, 한정된 시점만으로 가상 세계를 볼 수 있는 것이 아니라, Cosmo Player에서 제공되는 기능을 사용하여 사용자가 가상 캠퍼스 안을 자유롭게 움직일 수 있으므로, 실제 학교를 방문한 것과 같은 체험을 인터넷을 통하여 할 수 있게 된다. 그리고, 다양한 형태의 정보를 제공하여 사용자에게 더욱 효과적인 안내 정보를 제공할 수 있도록 하였다. 구현된 가상 물체들을 마우스 커서를 움직여 선택하면 브라우저의 왼쪽 프레임에 관련 정보가 표시되고 컴퓨터의 스피커를 통해 안내 음성이 나오게 된다. 이것은 발생한 이벤트를 VRML 파일의 Anchor 노드와 Sound 노드를 통해 처리할 수 있도록 하여 구현되었다.

본 연구에서 구현된 가상안내 시스템의 URL은 <http://cvl.ulsan.ac.kr/~lab/VirtualUOU/proto.html> 이다.

4. 결론

본 논문에서는 사용자에게 더욱 친근하고 이해하기 쉬운 사용자 인터페이스를 제공할 수 있고 인터넷의 범용성으로 인해 쉽게 접근할 수 있는 가상 현실 기반 안내시스템을 제안하였다.

가상 현실 기반 안내시스템을 구현하는 데 필요한 VRML과 네트워크에서의 VRML 동작 구조를 고찰함으로써 인터넷에서 가상 현실을 이용한 안내시스템의 구현에 대한 기초를 제공하였다. 그리고, 가상 현실 기반 안내시스템의 구현을 위해 HTML, VRML과 JavaScript를 사용하였는데, 이 세 가지 언어로 구현된 웹 페이지 사이에서의 정보 흐름 구조를 정의하여 상호 연동 할 수 있는 구조를 정의하였다. 이러한 구조를 통하여 사용자에게 더욱 다양한 멀티미디어 정보와 사용하기 쉬운 인터페이스를 구현하였다. 다음은 실제로 가상 현실 기반 안내시스템을 구현하는 데 있어 필요한 가상 현실내의 물체 제작에 대한 방법과 texture mapping을 이용하여 가상 현실 물체 제작을 위한 최적화 기술에 대한 기초를 제공하였다.

현재 인터넷의 대역폭과 client 컴퓨터의 rendering 속도를 생각해 볼 때 가상 현실을 인터넷에 구현하는 것이 시간적인 제약이 따르므로 본 연구에서는 texture mapping을 사용하여 가상 공간 내의 물체를 제작하는 방법으로 물체의 polygon 수를 줄이고 file 크기를 감소시켜 전송시간과 rendering 속도의 효율을 증가시켰으나 아직 이러한 전송 시간 단축에 관해 더욱 연구가 필요하다.

<참고문헌>

- [1] "VAG Mailing list", <http://vag.vrml.org/www-vrml/>
- [2] Andrea L. Ames, David R. Nadeau, John L. Moreland, "VRML 2.0 Sourcebook", John Wiley & Sons, Inc. 1996, p525
- [3] Jed Hartman, Josie Wernecke, "The VRML 2.0 Handbook", Silicon Graphics, 1996, p7
- [4] "VRML 2.0 Specification", VRML consortium, <http://www.vrml.org/Specifications>
- [5] "VRML Models", Orc inc., <http://www.ocnus.com/models/>
- [6] Netscape Communicator, <http://home.netscape.com/download/>
- [7] Cosmo Player 2.0, Silicon Graphic inc., <http://cosmosoftware.com/>