

Java3D를 이용한 3D 사용자 인터페이스 기법의 개발

오태철^o 고명철 최윤철
연세대학교 컴퓨터과학과
(nim, zoo, ycchoy)@rainbow.yonsei.ac.kr

Development of a 3D User Interface Technique with Java3D

Tae-Cheol Oh^o Myeong-Cheol Ko Yoon-Chul Choy
Dept. of Computer Science, Yonsei University

요 약

최근 인터넷과 비몰입형 가상현실 기술을 기반으로 하는 3D 사이버스페이스 분야에 대한 관심이 고조되고 있다. 그 동안 3D 그래픽의 처리나 네트워크 상의 데이터 전송기술과 같은 하드웨어 분야에서는 많은 개선이 있었으나 가상환경 내부의 다양한 상호작용을 처리하는 부분에 있어서는 연구가 미진한 실정이다. 특히 3D 사이버스페이스 상에서 참여자의 의사결정 및 상호작용의 수단이 되는 3D 사용자 인터페이스에 대한 연구는 성공적인 연구사례가 보고되지 않았다. 현재 실험적인 수준의 3D 사이버스페이스가 앞으로 좀더 활성화되고 일반화 되기 위해서는 사용자 관점을 지원하는 친숙하고 자연스러운 3D 사용자 인터페이스에 대한 연구가 필요하다. 본 논문에서는 현재 가장 일반적인 입력 장치로서 사용되는 2D 마우스를 이용하여 3D 사이버스페이스 상의 다양한 객체들을 자연스럽게 조작할 수 있게 하는 3D 조작 인터페이스를 제안한다. 제안된 3D 조작 인터페이스를 실시간 환경에 적용하여 검증한 결과 3D 사이버스페이스 상에서의 상호작용을 매우 효과적으로 지원할 수 있음을 확인하였다.

1. 서 론

인터넷이 인간의 생활 공간과 기존의 상거래에 대한 개념을 바꾸는 등 그 응용 분야가 다변화되고 확장되는 것과 더불어, 최근에는 인터넷을 기반으로 하는 가상현실(VR: Virtual Reality) 기술에 대한 관심이 고조되고 있으며 차세대 핵심기술의 하나로 인식되고 있다. 가상현실은, 초기에 단순히 3D그래픽을 통한 시각적 효과를 지원하는 수준이었으나 최근에는 사용자와의 상호 작용성을 중요시 하여 단순한 사이버스페이스가 아닌 실제로 사용자가 사이버스페이스 내에서 객체를 움직이고 조작할 수 있는 수준으로 발전하고 있다. 그러나 그 동안 사용자 인터페이스 기술은 고가의 하드웨어 장비를 이용하는 몰입형 가상현실(Immersive VR) 분야 위주로 연구되어져 왔으며 비 몰입형 가상현실(Non-immersive VR) 분야에서의 연구사례는 거의 보고되지 않았다. 또한 3D 객체의 조작 인터페이스와 관련하여 3D 모델링 툴과 같은 오프-라인 시스템에서는 그 동안 많은 기능들이 연구되었으나 3D 사이버스페이스와 같은 실시간 환경에 적용된 사례는 알려져 있지 않다. 따라서 대중적인 데스크탑 환경 하에서 사이버스페이스와 같은 실시간 3D 공간의 특성을 자연스럽게 반영하는 사용자 인터페이스에 대한 연구가 필요하다.

본 논문에서는 2D 마우스를 이용하여 3D 사이버스페이스 환경에서의 조작과 관련된 일련의 상호작용을 자연스럽게 처리할 수 있는 3D 조작 인터페이스를 제안하고자 한다.

2. 관련 연구

비몰입형 가상현실 분야에서의 사용자 인터페이스에 대한 기존 연구를 살펴 보면 Nielson[1]은 2D 마우스를 이용하여 3D 객체를 조작하는 방법을 제시하였는데 기하학적인 형상 모델링이나 기타 다른 어플리케이션에서 사용될 수 있는 점과 선의 위치지정과 이동에 관한 새로운 기술을 제안하였고,

K.Shoemake[2]는 객체를 회전 시키기 위해 2D 마우스가 객체 위에서 눌린(pressed) 위치와 해제된(released) 위치를 계산 하여 객체를 회전시키는 방법을 제시하였으며, Terri Stein[3]은 3개의 좌표 면으로 이루어진 마우스를 이용하여 2D 스크린 상의 3D 객체를 조작하는 방법(Metric Cursor)을 각각 제안하였다. 그러나 이러한 방법들은 일반 사용자들이 대중적으로 사용하는 인터넷 상의 가상환경을 위한 인터페이스가 아니라 3D 객체 제작을 위하여 객체를 구성하는 점이나 선을 조작하는 일련의 인터페이스 기능에 국한되어져 있다. 본 논문에서는 이들이 제안하는 방법을 바탕으로 기존 3D 객체 모델링 툴에서 제공하는 인터페이스 기능을 개선하여 실시간 3D 사이버스페이스에서 사용자가 원하는 대로 객체를 쉽게 조작할 수 있는 인터페이스 기법을 제시한다. 3D 객체에 대한 조작은 먼저, 객체를 만들어 내는 'Spatial rigid object manipulation'과 객체를 선택하고 움직이거나 돌리는 'Basic 3D manipulation' 등 두 가지 과정으로 구분할 수 있다[4]. 즉, 전자는 3D 객체 모델링 툴의 일반적인 기능이라고 볼 수 있고, 후자는 본 논문에서 다루고자 하는 실시간 3D 사이버스페이스 상에서의 일련의 객체 조작 기능이라고 할 수 있다.

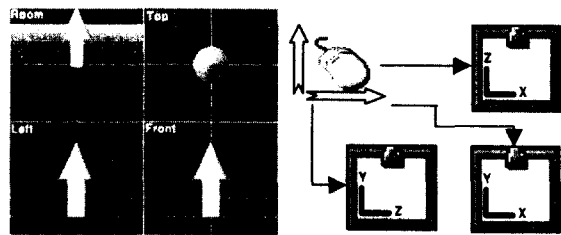


그림 1. 저작도구 편집 창

현재 대부분의 3D 객체 모델링 툴에서는 객체 편집 시에 통상 3개 또는 4개의 창을 이용한다. 그 이유는 마우스를

입력장치로 하는 2D 환경 하에서 3D 객체를 조작하기 위해서는 마우스가 가지는 2개의 방향성을 3D로 매핑 할 수 있는 3개의 면(face)이 필요 하기 때문이다[그림1]. 이러한 방법은 각각의 면을 이용하여 2D에서 3D로 효과적으로 매핑 시키고 있지만, 각 면을 관리하는 창을 따로 둬으로써 사용자가 객체를 조작할 때마다 각각의 창을 선택하고 객체를 다시 선택해야 하는 불편함이 있고, 하나의 객체에 대해서 3개 또는 4개의 창을 제공 한다는 것은 본 논문에서 다루고자 하는 실시간 사이버스페이스 환경에는 적합하지 않다. 만일 이러한 방법을 인터넷 상의 사이버스페이스에 적용할 경우 사이버스페이스는 상당히 복잡해질 것이기 때문이다. 따라서 이러한 면(face)을 하나의 창에 효과적으로 표현하고 사용자가 이를 자연스럽게 이용할 수 있게 하는 3D 조작 인터페이스가 필요하다.

3. 3D 객체의 조작

3.1 정의

Bowman[5]은 3D 객체의 조작을 크게 선택(Selection), 조작(Manipulation), 해제(Release) 등 세 가지로 구분하고 각각의 기능에 해당하는 요소 기술들을 나열하였다. 즉, 객체를 조작하기 위해서는 먼저 어떤 특정한 방법을 통해 객체를 선택하여야 하고 선택 후에는 객체를 돌리거나 움직이는 조작과정을 거친 후 다시 객체가 선택되지 않은 상태가 되어야 한다는 것이다. 이러한 과정에는 객체가 선택되었는지 어떻게 조작하여야 하는 지를 사용자에게 보여주는 피드백 기술이 포함된다. 이러한 분류는 특수한 장비를 이용하는 몰입형 가상 현실분야를 전제로 한 것이지만 2D 디스플레이 장치와 마우스를 사용하는 3D 데스크 탑 가상 환경분야에도 충분히 적용이 가능하다. 따라서 본 논문에서는 Bowman이 제안한 분류에 기반 하여 사이버스페이스와 같은 실시간 환경에서 특수하게 요구되는 추가적 기능들을 다룬다.

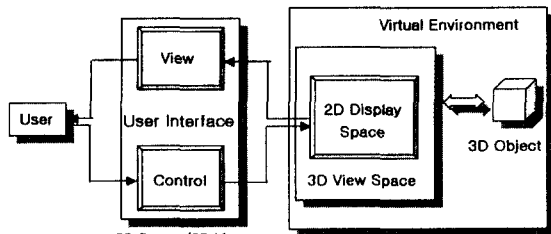


그림 2. 3D 사용자 인터페이스 개념도

또한 2D 화면상에서 2D 마우스를 이용하여 3D 객체를 선택하고 조작하는 일은, 2D와 3D간의 공간개념의 불일치로 인하여 참여자로 하여금 많은 혼란과 조작의 부자연스러움을 유발 할 수 있다[그림 2]. 이러한 문제는 단순히 자유도(DOF)의 증가만으로는 해결 될 수 없다. 본 논문에서는 이러한 공간개념의 불일치 문제를 해결 할 수 있는 사용자 인터페이스 기능들도 함께 다룬다.

3.2 객체의 선택 및 조작

2D 마우스를 이용한 3D 객체의 선택과 조작은 [그림3]에서와 같이 면(face) 피드백 시각정보를 이용한다. 이는 기존 3D 모델링 툴에서 여러 개의 편집 창을 이용한 객체조작 기능을 사이버스페이스와 같은 실시간 환경에 적합하도록 개선

한 기법으로서, 단일 창 내에서 세 평면(X-Y, X-Z, Y-Z)에 대한 시각적인 피드백 정보를 마우스의 세 버튼 각각에 매핑 하여 적절히 제시한다. 즉, [그림1]에서와 같이 Top, Left, Front 등 세 개의 창을 의미하는 좌표공간에 대한 시각적인 정보를 마우스의 각 버튼에 따라 적절히 제시하는 것이다. 사용자는 마우스를 이용하여 객체를 선택한 후에 마우스의 세 버튼을 이용하여 원하는 좌표평면 상에서 객체에 대한 이동, 회전 등 다양한 조작 기능을 수행 할 수 있게 된다.

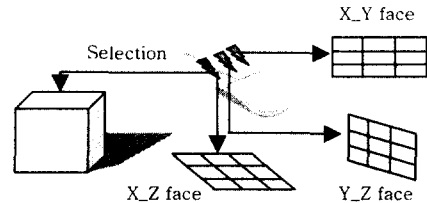


그림 3. 3D 좌표평면과 마우스 버튼의 매핑

4. 인터페이스 구현

4.1 Scene Graph의 구성

본문의 구현은 Sun사의 Java3D를 이용하여 구현 하였다. Java3D는 계층구조를 가지는 언어로 그 구조가 VRML과 매우 유사하지만 더욱 광범위한 기능을 가지고 있다. 장면 그래프(Scene Graph)라는 구조로 되어 있기 때문에 삼차원 그래픽스 프로그래밍이 편리하고, 네트워크에 강한 자바의 장점을 그대로 가지고 있어 네트워크 환경에서 삼차원 그래픽을 이용할 수 있으며, 플랫폼에 독립적인 특징으로 인하여 성능 확장이 쉽다.

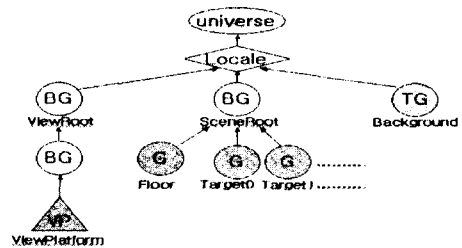


그림 4. Scene Graph

본 논문에서 제안하는 사용자 인터페이스 구현의 전체적인 장면그래프(Scene graph)를 간략하게 살펴보면[그림 4], 최상위 노드에는 Java 3D 프로그램의 계층구조의 특징이라고 볼 수 있는 Virtual universe를 두었고 그 밑에 뷰 관련 Branchgroup 과 targets 및 floor등을 관리하는 Scene관련 Branchgroup, 전체 Universe의 배경색이나 기타 환경속성을 설정해줄 수 있는 Transformgroup을 두었는데 SceneRoot Branchgroup에는 java3D로 제작한 object는 물론, VRML로 제작한 object도 load할 수 있도록 하였다.

4.2 2D/3D 매핑

3D 저작 도구에서 사용하고 있는 3개의 창 기능을 마우스 세 개의 버튼으로 처리하기 위해서는, 각각의 버튼을 누른 상태에서 발생하는 드래그를 구분하고, 구분된 2D의 드래그 값을 3D로 매핑시키는 것이 필요하다. Awt Event의 Mouse Event와 화면상의 마우스 point를 참조하여 마우스 버튼에 따른 X축 Y축으로의 드래그를 각각 다르게

구분하고[그림 5], 구분된 값 들을 각각의 면(face)으로 매핑될 수 있도록 하였는데, 마우스의 X축 값과 Y축 값을 원하는 면(face)으로 매핑시키기 위해서 [그림 6]의 매핑 방법을 사용하였다.

먼저 Boolean값을 가지는 두 개의 배열 _mapX[3], _mapY[3]를 선언하고 각각 X값과 Y값을 true로 초기화 한 후 아래 함수에서와 같이 매핑하고자 하는 두개의 축 값을 받게 되면 2D 마우스의 X축과 Y축의 움직임이 각각 X_Z, X_Y, Y_Z의 3개의 면(face)으로 각각 매핑되고, Rotation시에는 고정하고자 하는 축과 마우스의 움직임을 값을 반영하고자 하는 축 값을 받게 되면 하나의 축은 고정되고 다른 하나의 축의 드래그 값이 반영되는 방법이다.

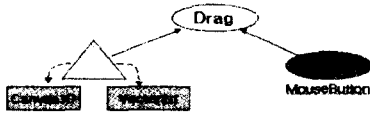


그림 5. 마우스 드래그

```

    X   Y   Z   W
_mapX[]: [ T   F   F   F ]
_mapY[]: [ F   F   T   F ]
public void setSourceMap(int mapX, int mapY) {
    _mapX[0] = ((mapX & Mapper.DIM_X) == 0) ? false : true;
    _mapX[1] = ((mapX & Mapper.DIM_Y) == 0) ? false : true;
    _mapX[2] = ((mapX & Mapper.DIM_Z) == 0) ? false : true;
    _mapX[3] = ((mapX & Mapper.DIM_W) == 0) ? false : true;

    _mapY[0] = ((mapY & Mapper.DIM_X) == 0) ? false : true;
    _mapY[1] = ((mapY & Mapper.DIM_Y) == 0) ? false : true;
    _mapY[2] = ((mapY & Mapper.DIM_Z) == 0) ? false : true;
    _mapY[3] = ((mapY & Mapper.DIM_W) == 0) ? false : true;
}

X_Z face :
_mapX==DIM_X → _mapX[]: [ T   F   F   F ]
_mapY==DIM_Z → _mapY[]: [ F   F   T   F ]
X_Y face :
_mapX==DIM_X → _mapX[]: [ T   F   F   F ]
_mapY==DIM_Y → _mapY[]: [ F   T   F   F ]
Y_Z face :
_mapX==DIM_Y → _mapX[]: [ F   T   F   F ]
_mapY==DIM_Z → _mapY[]: [ F   F   T   F ]
    
```

그림 6. 2D/3D의 매핑함수

4.3 객체의 조작 과정

본 논문에서 제안하는 인터페이스는 객체를 옮기고 이동하는데 필요한 'Translation' 모드와 객체를 돌려보고 회전시킬 수 있는 'Rotation' 모드 등 2가지 모드를 제공한다.

이를 구분하기 위해서 불편한 팝업 메뉴형태의 인터페이스를 사용하지 않고 흔히 사용하는 방법인 'single click'과 'double click'으로 구분하였다. 즉, 'single click'은 객체 이동을 위한 모드이고, 'double click'은 객체회전을 위한 모드이며 객체 선택 후에는 마우스에 있는 세 개의 버튼을 이용, 원하는 면과 축으로 움직이거나 회전시키면 된다[그림 7]. 사용법을 쉽게 알 수 있도록 하기 위해 객체를 선택하였을 때 사용자가 마우스를 어떻게 사용해야 하는가를 알 수 있도록

특 하는 문장을 보여주고, 객체를 선택하면 객체가 선택되었는지를 알 수 있게 객체 주위에 바운딩 박스(bounding box) 처리를 하였다. 객체 선택 후에 객체 위에서 각각의 마우스 버튼을 눌렀을 때 해당된 면과 축 방향으로 화살표를 보여줌으로써 객체가 어떤 방향 어떤 축으로 이동 및 회전이 가능한 지 알 수 있도록 하는 피드백 기술들을 적용하였다.

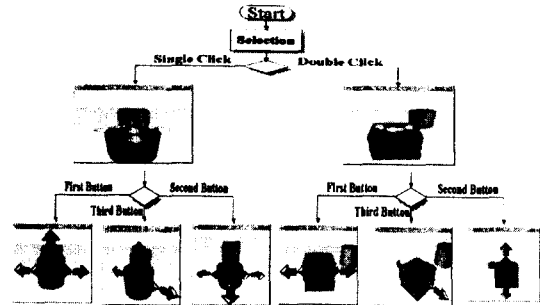


그림 7. 실행화면

5. 결론

본 논문에서는 데스크 탑 환경의 비몰입형 가상환경 하에서 2D 마우스를 이용하여 3D 공간에서의 객체조작을 위한 사용자 인터페이스를 제안하였는데, 2D 마우스가 두 개의 축으로만 이동하는 것을 고려하여 3D 환경에서의 3개의 축을 가지고 만들 수 있는 3개의 면에 이러한 평면의 움직임을 매핑시키는 방법으로 새로운 사용자 인터페이스를 제안하였다. 아직 많은 사용자들이 단순한 2D 마우스와 인터페이스를 선호하고 있는 상황에서 본 논문에서 제안하는 사용자에 친숙한 3D 사용자 인터페이스는 다른 여러 응용분야에 효과적으로 적용될 수 있을 것이다.

6. 참고문헌

- [1] G.M.Nielson and D.R.Olsen,Jr. Direct manipulation techniques for 3D objects using 2D locator devices. Proc. of the 1986 Workshop on Interactive 3D Graphics, pp. 75-182,1986.
- [2] K.Shoemake. ARCBALL : A user interface for specifying three-dimensional orientation using a mouse. Proc. of the Graphics Interface'92, pp. 151-156,1992.
- [3] Terii Stein and Sabine Coquillart, The Metric Cursor. Proc. of the Eighth Pacific Conference. IEEE 2000. 2000.
- [4] Doug A. Bowman, Ernst Kruijff, et al. 3D User Interface Design: Fundamental Techniques, Theory, and Practice, Proc. of the SIGGRAPH 2000 Course Notes, 2000.
- [5] Doug A. Bowman, Donald B. Johnson and Larry F. Hodges; Testbed evaluation of virtual environment interaction techniques; Proc. of the ACM symposium on Virtual reality software and technology, pp. 26-33. 1999.