

선택 트리 순회를 이용한 3D 객체선택 인터페이스

임윤호, 한덕수, 임순범*, 최윤철
연세대학교 컴퓨터과학과
*숙명여자대학교 멀티미디어학과

3D Object Selection Technique using Selection Tree Traversal

Yoon-Ho Lim, Deok-Soo Han, Soon-Bum Lim*, Yoon-Chul Choy
Dept. of Computer Science, Yonsei University

*Dept. of Multimedia Science, Sookmyung Women's University
E-mail : (dshan,limyh,ycchoy)@rainbow.yonsei.ac.kr, sblim@sookmyung.ac.kr

요 약

3D 그래픽스는 이제 전문가들만의 전유물이 아니다. 하드웨어 기술의 비약적인 발전으로 누구나 저렴한 가격으로 불과 몇 년전의 워크스테이션급 이상의 3D 그래픽스 환경을 갖출 수 있게 되었다. 그러나 3D 그래픽스 환경을 조작하기에 적합한 특수 장비는 여전히 일반 소비자가 사용할 수 없을 정도의 고가이며 대중화는 요원한 실정이다. 따라서 일반적인 2D 기반의 입력장치를 활용한 데스크탑 가상현실에서의 효과적인 인터페이스의 연구가 필요하다. 사용자가 최초로 가상 환경내의 객체를 조작하기 위해서 취하는 행동은 선택이다. 그러나 일반적인 데스크탑 환경에서 3D 객체의 선택은 정확성이 떨어지며 오류의 정정도 번거롭다. 본 논문에서는 일반적인 마우스를 가지고 3D 가상환경의 객체를 정확하게 선택할 수 있도록 돕는 인터페이스를 제안한다. 시스템은 사용자가 선택한 객체와 관련된 선택 가능한 후보 객체들로 구성된 선택 트리를 제시하며, 선택 트리 내에서의 순회를 통해 사용자가 정확한 선택과 오류 정정을 하도록 돕는다.

1. 서론

최근 3D 그래픽스 기술은 급속도로 발전하고 있다. 과거에는 고가의 워크스테이션에서만 구현 가능했던 3D 그래픽은 이제 일반 사용자가 사용 가능한 저렴한 가격대의 PC로도 가능하게 되었다. 이런 하드웨어의 발전에 힘입어 많은 사용자들은 고성능, 고품질의 3D 그래픽스 환경을 접하게 되었다. 이런 3D 그래픽스 기술의 대중화는 비단 그래픽스 관련 프로그램뿐 아니라 교육, 훈련, 엔터테인먼트, 게임을 비롯 전자상거래나 스프레드시트 같은 사무용 프로그램에 이르기까지 3D 그래픽스 기법의 수요를 증가시키고 있다. 그러나 렌더링 기술의 비약적인 발전에 비해 3D 환경에 적합한 입력장치의 발전속도는 상대적으로 매우 더디게 진행되어 특별한 전문가가 아닌 일반적인 사용자의 경우 3D 환경에 적합한 특수한 입력장치를 소유하지 못하고 있다. 일반적으로 가상현실을 지칭하면 연상되는 HMD(Head mounted display)와 데이터 글로

브드의 몰입형 장비는 물론, 3D Mouse인 Space mouse, Spaceball 등의 6DOF를 지원하는 3D 입력 장치도 상당한 고가의 제품으로 일반사용자가 구입해서 사용하기에는 무리가 있다. 현실적으로 현재 대부분의 컴퓨터 사용자들은 평범한 키보드, 마우스를 통해 2D 화면에 투영된 3D장면을 가지고 작업하는 것이 일반적이다.

사용자의 입력 환경을 개선하고자 하는 연구는 크게 두가지 방향에서 접근할 수 있다. 첫째는 고가의 장비들을 사용하거나 새로운 장비들을 개발하는 하드웨어적인 접근 방법이며 두번째는 기존의 2D 기반 장비들을 활용해서 3D 그래픽스와 인터랙션 하도록 소프트웨어적인 기법들을 활용하는 소프트웨어적인 접근 방법이다. 하드웨어적인 접근 방법은 앞서 언급된 고가의 장비가 필요하며 대중적으로 사용되기까지는 아직 시간이 많이 걸릴 것이므로 본 논문은 소프트웨어적인 접근 방법으로 기존의 2D 환경을 적극적으로

활용할 수 있는 인터페이스를 제안할 것이다.

일반적으로 가상환경에서 일어나는 일련의 행위는 크게 가상공간을 항해하는 네비게이션(Navigation)과 3D물체의 조작(Manipulation)으로 구분할 수 있다. 초창기 가상환경에서는 가상공간의 탐험이 주를 이루었으며 네비게이션 기법에 대한 연구가 활발히 이루어져왔다. 그러나 효율적인 네비게이션 기법과 더불어 점차 자연스러운 3D물체의 조작기법에 대한 요구도 증가하여 이에 부응하는 연구가 진행되고 있지만 지금까지의 연구들은 주로 다양한 하드웨어를 기반으로 하는 몰입형 가상현실 분야에 치중되어 있다. 본 논문이 주목하는 데스크탑 가상환경에서의 3D물체 조작기법은 데스크탑 환경의 특성상 입력장치가 주로 키보드와 2D 마우스에 국한되는 하드웨어적 제약과 3D공간상의 객체를 2D상에서 조작함으로써 발생하는 공간 불일치 등의 문제로 효율적인 물체조작 기법이 부족하지만 상대적으로 적은 연구만이 진행되어 있을 뿐이다.

특히 본 논문은 데스크탑 가상현실 환경에서 수행되는 3D 객체의 선택과 관련된 내용을 다룰 것이다. 사용자는 객체를 선택하는 것으로 조작을 시작하므로 선택은 모든 조작의 출발점이 되는 기본적인 조작으로 볼 수 있다. 몰입형 가상환경에서는 이미 다양한 선택 방법들에 대한 연구가 진행되어 왔다. 그러나 데스크탑 가상현실에서 3D 객체의 선택작업은 객체 위에 커서를 위치시킨 후 마우스 클릭으로 선택하는 방법이 표준처럼 굳어져 왔기 때문에 거의 연구가 진행되어 있지 않다. 그러나 그런 마우스 클릭만을 이용한 기존의 방법은 그다지 직관적이지 아니며 정확도가 떨어진다. 본 연구에서는 기존의 방법을 보다 정확도를 높이도록 보완하기 위해 객체 선택 시 선택 트리를 구성하여 선택된 객체의 주변의 연관된 객체간에 순회를 통해 선택 포커스를 이동하여 정확도를 높이는 방법을 제안하며 프로토타입을 구현하여 그 아이디어의 가능성을 알아보고자 한다.

2. 관련연구

3D객체의 조작은 여러 관점에서 분류할 수 있다. 먼저 Bowman[1]은 가상공간에서의 3D 객체조작을 크게 선택(Selection), 조작(Manipulation), 해제(Release)등 세 가지로 구분하였고, Foley, D. 등 여러 연구자들[2, 3, 4]은 컴퓨터 그래픽스 인터렉션을 선택(Selection), 배치(Position), 회전(Orient), 텍스트입력(Text), 숫자입력(Quantify)등의 다섯 가지로 구분하

였다. 이것을 토대로 생각해 보면 객체조작기능은 크게 객체의 선택, 회전, 이동, 선택해제로 이루어짐을 알 수 있다. 그 중에도 선택과정은 항상 개별 객체의 조작의 시작부분에 행해지는 조작으로 기초적이면서 중요한 조작이라 할 수 있다.

선택 작업은 몰입형 가상현실에서는 비교적 많은 연구가 진행되었다. 몰입형 가상현실에서의 대표적인 객체조작기법들은 'Simple virtual hand', 'Ray-casting', 'Go-go technique' 등의 대표적 객체조작기법들 외에도 'Flash light', 'Aperture', 'Image plan', 'World-in-Miniature' 등 다양한 기법들이 있으며 여러 기법들을 혼합하여 사용하는 형태도 존재한다[5, 6, 7].

위에 언급된 기법 중 특히 'Ray-casting' 기법을 보완하기 위해 나온 'Flash light' 기법에 주목할 필요가 있다. 'Ray-casting'은 사용자의 virtual hand에서 뻗어나가는 가상의 광선으로 멀리 있는 객체를 선택하는 몰입형 가상현실에서 사용되는 선택 기법이다. 그러나 이 기법은 멀리 떨어진 작은 물체를 선택하기가 어렵다는 단점이 있다. 단지 한 점을 선택하는 광선이라는 민감한 메타포이기 때문에 발생하는 이 문제에 대한 보완책으로 연구된 기법이 'Flashlight' 기법으로 이 기법은 마치 손전등을 비추듯 일차적으로 일정 범위의 복수의 객체를 선택하여 선택된 객체 내에서 다시 정확한 선택을 하는 방법이다. 이 경우 일차로 선택된 객체 내에서 다시 선택을 처리해야 하는 문제점이 있다. 그러나 정확도를 높이기 위해 복수의 객체를 선택하는 방법이라는 점은 본 논문에서 제시하고자 하는 기본 개념과 유사하다.

데스크탑 환경에서는 기존의 방법인 선택을 원하는 객체 위에 커서를 위치시킨 후 클릭하는 방법이 표준처럼 굳어져 큰 연구가 진행되지 않았다.

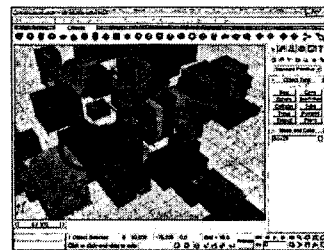


그림 1. 직접적인 객체선택

그러나 이 방법은 [그림 1]에서 원 안에 있는 가려

진 육면체처럼 화면에서 차지하는 크기가 작은 경우 한번에 정확한 선택을 하기가 힘들다. 이런 경우 시점의 이동, 확대 등의 작업이 병행되어야 하므로 사용자는 불편을 느끼게 된다. 따라서 대부분의 3D 그래픽스 프로그램들은 [그림 2]와 같은 선택 방법도 지원한다. 이 방법은 객체 자체가 아닌 객체의 이름, 색깔 등의 속성을 토대로 별도의 창을 통해 선택하는 간접적인 방법이다.

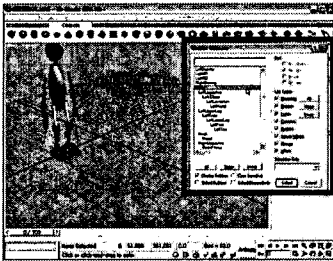


그림 2. 간접적인 객체선택

간접적인 선택방법은 정확한 선택은 가능하지만 별도의 창에서 이루어지므로 사용자의 시선이 이동되어 작업의 집중도를 떨어뜨릴 가능성이 있으며 객체의 속성을 세심하게 고려하고 기억해야하는 부담이 있다.

3. 선택 기법의 제안

본 논문에서 제안하는 선택기법은 선택의 정확도를 높이기 위해 마우스 클릭에 의한 직접적인 선택방법으로 선택된 객체 외에 그와 연관된 복수의 선택 가능 객체 후보들을 선정한다. 그 다음 그 객체들 사이에서 포커스를 이동하여 선택을 돕는 인터페이스를 제공한다.

3D 장면의 데이터는 대부분 트리구조로 저장된다. VRML을 비롯한 여러가지 그래픽 파일은 장면 그래프를 트리구조로 저장[8]하며 트리구조는 계층적인 구조를 기술할 수 있으며 탐색 속도가 빠르기 때문에 3D 장면 그래프에서 매우 유리한 데이터 구조이다. 그러므로 3D 장면 내에서 선택 가능한 후보들도 트리 형태의 데이터 구조를 갖는 것이 효율적이라고 할 수 있다. 따라서 본 논문에서는 객체 선택을 돕기 위해 원래의 장면 그래프에서 분리, 혹은 생성하는 소규모의 트리 형태로 된 장면 그래프를 선택 트리(Selection tree)라고 명명하였다. 생성된 선택 트리는 트리 내부를 순회함으로 포커스를 이동한다.

4. 선택 트리의 생성

선택 트리는 전체 장면 그래프에서 분리하는 방법과 처음부터 새로이 생성하는 방법, 크게 두 가지 방법으로 구할 수 있다. 그것은 계층적인 장면 그래프의 특성을 고려할지의 여부와 관련된다.

첫번째 방법은 전체 장면 그래프에서 선택 트리를 분리하는 것이다. 이 방법은 간단할 뿐만 아니라 전체 장면 그래프의 계층적인 구조 정보를 포함하는 선택 트리가 만들어진다는 장점이 있다. 그러나 장면 그래프를 어떻게 분리할 것인지에 대한 정책적인 결정이 필요하다. 복수의 트리로 구성된 장면 그래프인 경우는 간단히 초기 선택 객체가 포함된 트리를 선택 트리로 분리하여 그 안에서 순회를 시작하면 가장 적절할 것이다. 그러나 각각의 트리가 매우 방대하거나 단일한 트리로 구성된 방대한 장면 그래프인 경우는 트리를 분리하기 위한 정책이 필요하다. 본 논문에서는 분리 정책으로서 선택된 객체의 부모 노드를 루트로 한 서브 트리 전체를 분리하는 방법을 제안한다. 대부분의 장면 그래프에서 자식 노드는 부모 노드에 포함되어 있기 때문에 부모 노드보다 작거나 복잡하게 얽혀있을 가능성이 크다. 그리고 트리 구조의 계층적인 장점을 살리기 위해서도 선택된 객체의 자식 노드는 선택 트리에 포함되어야 한다. 그리고 부모 노드를 선택 트리에 포함시키는 이유는 선택된 객체의 형제 노드로 이동할 수 있게 하기 위한 것이다. 앞서 언급된 선택과정의 오류는 대개 형제나 자식노드간에 이루어질 가능성이 크기 때문에 이러한 정책은 사용자가 쉽게 선택을 정정할 수 있도록 돕는 유용한 가이드라인이 될 수 있다. [그림 3]에서 칠해진 부분은 이러한 정책에 의해 분리된 선택트리이다.

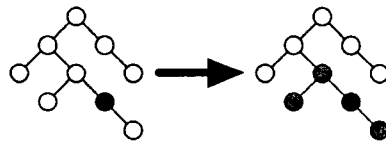


그림 3. 선택 트리 분리

두번째 방법은 선택 트리를 분리에 의해 얻는 것이 아닌 새로 생성하는 방법이다. 이 방법은 전체 장면 그래프의 구조적인 정보가 없는, 단순하게 객체들이 나열된 데이터 구조를 갖는 경우, 혹은 전체 장면 그래프의 계층적인 구조를 무시할 필요가 있을 때 사용될 수 있다.

사용자가 가장 많은 선택 오류를 범하게 되는 경우는 객체가 상대적으로 아주 작은 크기일 때, 객체들이 매우 가까이 밀착해 있는 경우, 그리고 다른 객체에 의해 가려지는 경우를 들 수 있다. 본 논문에서는 선택 트리를 생성하는 방법으로 인접성 테스트를 제안한다. 사용자는 원하는 객체 근처를 선택하고자 할 가능성이 높으며 인접해 있는 객체는 서로 연관성을 가질 확률이 크다. 그러므로 인접성은 장면그래프에서 선택 트리를 만들어 내기 위한 좋은 출발점이 된다. 본 논문에서 제안하는 인접성 테스트 방법은 [그림 4] (a)의 객체간의 인접성과 (b)의 뷰 벡터(View vector)와의 인접성이다. 여기서 k 는 상수값으로 인접성의 민감도를 조절할 수 있도록 한다. 일반적으로 3D 장면에서 거리를 구하는 방법은 다양하며 정교한 검사의 경우 매우 복잡한 연산이 필요하다. 그러므로 대부분의 경우 물체를 감싸는 경계 볼륨(bounding volume) 간의 간단한 연산을 통한 알고리즘을 많이 사용한다 [9]. 본 논문에서 제안하는 방법은 객체간의 인접도만 구하면 되므로 가장 간단한 방법인 경계구(bounding sphere)를 사용하였다. 경계구의 중심에서의 거리를 구하는 방법으로 간단히 인접도를 측정할 수 있다.

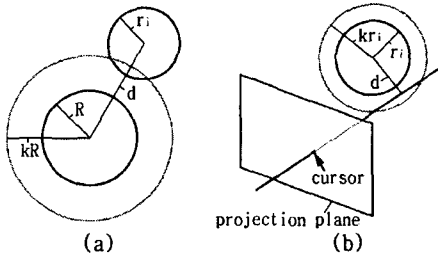


그림 4. 인접성 테스트

[그림 4] (a)의 객체간의 충돌을 이용한 경우 대부분 간단하게 인접성 테스트를 수행할 수 있다. 그러나 화면 깊숙히 가려져 있는 객체는 선택 트리에 포함시키기 어렵고 정밀도가 떨어진다는 단점이 존재한다. 그래서 그 대안으로 (b)의 뷰 벡터와의 인접성 테스트를 설정하였으며 정밀도의 경우 k 값의 조절, 혹은 다른 보다 복잡한 경계볼륨의 도입으로 해결할 수 있다.

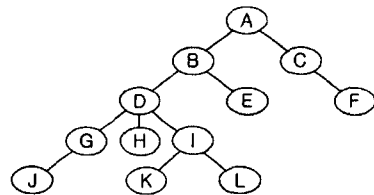
인접성 테스트를 위한 알고리즘을 선택했다면 우선 화면 내의 모든 객체들에 대해서 선택한 방법으로 인접성 테스트를 수행한다. 그리고 테스트를 통과한 객체들을 모아서 리스트로 만든다. 이 리스트를 선택 트리를 순회한 결과로 간주하면 제안한 선택 트리 순회

방법을 사용할 수 있다. 그러나 객체들이 많은 경우 비교할 회수가 늘어나며 최종 결과물의 순회 순서 객체의 내부 인덱스 번호와 같은 검색 순서로 결정되므로 직관적이 아닌 선택 트리가 생성된다는 단점이 있으므로 전처리와 후처리에 대한 보완이 필요하다.

4. 선택 트리의 순회

포커스의 이동은 선택 트리의 순회(traversal)를 통해 이루어진다. 일단 선택 트리를 생성하는 후자의 방법들에서는 인접성 테스트의 결과물 자체를 순회 결과로 대치할 수 있으므로 주로 선택 트리를 전체 장면 그래프로부터 분리하는 경우에 주로 해당된다. 트리의 순회 방법은 전위(preorder), 중위(inorder), 후위(postorder) 순회방법, 그리고 트리의 레벨 순서 순회방법이 있다[10]. 객체의 계층적인 제시가 가능한 순회방법은 전위와 깊이에 따른 순회방법이다. 다른 순회 방법들은 부모, 상위 노드가 나중에 제시되는 순회 방법이므로 사용자에게 계층적인 순회방법으로 보이지 않을 수 있다.

전위 순회는 노드를 먼저 방문하고 그 다음 한쪽 자녀의 모든 노드를 방문한다. 리프 노드에 도달하면 오른쪽 하나 옆의 자녀를 가진 가장 가까운 조상으로 돌아와 오른쪽 자식에서 재귀적으로 순회를 반복하는 방법이다. 레벨 순서 순회는 노드를 먼저 방문하고 왼쪽 자녀부터 차례로 오른쪽 자녀 순서로 방문하는 것으로 결과적으로 트리의 레벨 순서에 따른 선형 리스트를 얻게 된다. 각각의 순회 방법으로 다음과 같은 트리를 순회한다면 그 결과는 [그림 5]와 같다.



전위 순회 : A B D G J H I K L E C F

레벨 순회 : A B C D E F G H I J K L

그림 5. 트리 순회의 예

제안된 방법에서 마지막으로 고려할 점이며 가장 중요하다고 할 수 있는 점은 순회의 조작을 실제 입력장치의 어떤 조작에 매핑시킬 것인가다. 이 방법을 성공적으로 적용하려면 기존의 조작 방법과 중복되지 말아야 하며 사용자가 직관적으로 받아들일 수 있는

메타포여야 한다. 현재 하드웨어에서 사용 가능한 방법은 마우스 휠을 사용하는 방법, 마우스의 추가 버튼들을 사용하는 방법, 그리고 키보드와의 연계 및 다른 제스처를 정의하는 방법 등 세가지로 분류하였으며 각각의 장단점은 [표 1]에 나와있다. 프로토타입은 제일 추천할만한 방법인 마우스 휠을 사용하였다.

	마우스 휠	4, 5번 버튼	키보드와 연계 기타 제스처
장점	가장 직관적 조작 자체가 방향 성을 가지므로 메 타포에 부합	다른 조작과 충돌 할 가능성 희박 비교적 직관적	모든 상황에서 적용 가능한 방법 유연하게 적용 가능
단점	화면 스크롤이나 확대기능등 기존 의 조작과 충돌 가능성	특정 하드웨어에 서만 사용 가능	양손을 사용하거나 추가적인 조작 을 기억해야 함

표 1. 하드웨어 매핑

5. 프로토타입의 구현

프로토타입은 Microsoft Visual Studio 6.0 의 Visual C++와 OpenGL 라이브러리를 사용해 MFC에서 개발되었다. 프로토타입은 실험목적으로 별도의 파일 포맷을 사용하고 있지만 개념적으로는 VRML이나 X3D에 충분히 적용할 수 있도록 만들어졌다. 모든 파일은 읽어들이는 과정에서 경계구의 크기와 축에 정렬된 경계 육면체, 객체의 모델링 좌표 중심을 미리 계산해서 저장한다. 프로토타입의 시각적 피드백은 색깔과 크기, 투명도를 이용했다. 실험을 위해 모델을 읽어들이기 때 제질 정보는 무시되어 흰색으로 통일되어 표시되며 선택된 객체는 붉은 색으로, 선택 트리는 투명하게 처리된 붉은 색으로 표시하였다. 선택되지 않은 객체는 투명하게 알파블렌딩으로 처리하여 선택된 객체들을 명확하게 표시하였고 선택된 부분을 화면 중앙으로 이동하고 윈도우 크기에 맞춰 확대하여 쉽게 선택한 부분을 확인할 수 있도록 구성하였다.

프로토타입은 IETM을 가정한 것으로 선택 작업 외에 간단한 회전, 이동작업을 병행하는, 선택 이후의 작업들과의 연계에 대한 부분과, 회전량을 표시해주는 참조 객체와 객체에 대한 설명이 표시되는 부분을 가지고 있다.

테스트에 사용된 모델인 4개의 피스톤은 각각의 피스톤 마다 별도의 트리 구조를 가지고 있어서 4개의 트리로 구성된 장면이다. 우선 처음에 선택하는 과정은 기존의 방법과 동일하다. 사용자는 원하는 객체로 커서를 움직여 마우스 왼쪽 버튼을 클릭하여 선택을 수행한다. 그림에서는 피스톤의 암 부분을 클릭한 경

우로 [그림 6]과 같은 화면을 볼 수 있다. 피스톤 1개가 선택 트리로 분리되었고 현재 클릭한 객체의 포커스가 맞춰져 있다.

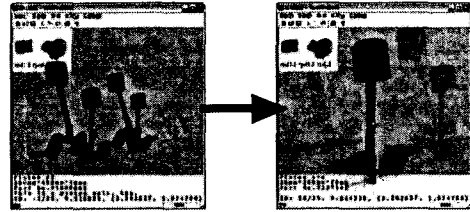


그림 6. 초기 선택

이때 마우스 휠을 1단 사용자 방향으로 돌리면 [그림 7]과 같이 전위 순회방향으로 자식 객체가 선택되게 된다.

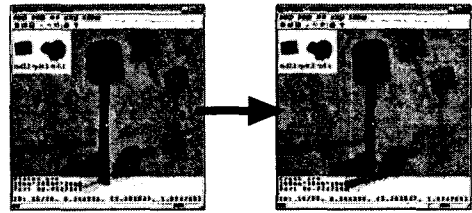


그림 7. 마우스 휠 1단 사용자 방향으로 돌린 경우

반대로 마우스 휠을 1단 사용자 반대 방향으로 돌리면 [그림 9]와 같이 전위 순회방향으로 부모 객체가 선택되게 된다.

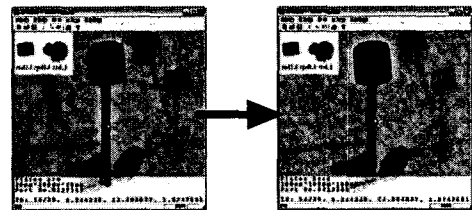


그림 8. 사용자 반대 방향으로 돌린 경우

객체들간의 인접성 테스트를 사용한 결과는 다음과 같다. 여기서는 피스톤 헤드를 클릭한 경우이다. [그림 10]에서 피스톤 헤드와 인접된 객체들이 선택트리로 선택되어진 것을 볼 수 있다.

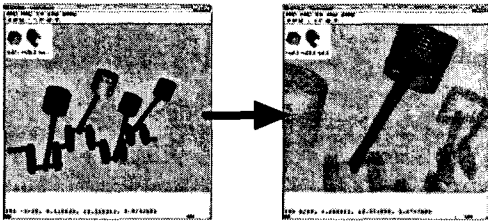


그림 10. 객체들간의 인접성

뷰벡터와의 인접성으로 선택트리를 구성한 경우는 [그림 11]처럼 깊숙하게 위치한 객체들까지 선택 트리에 포함되었음을 볼 수 있다.

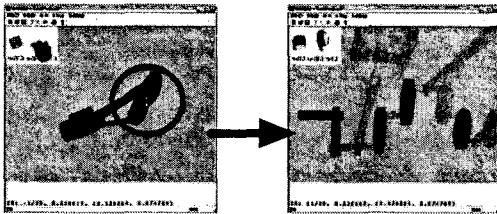


그림 11. 뷰벡터와의 인접성

6. 결론 및 향후 연구방향

본 논문에서 제안하는 정확하게 객체를 선택할 수 있는 방법은 실험 결과 긍정적으로 적용 가능하다는 결과를 얻었다. 본 기법은 기존의 방법에서 불가능한 세밀한 선택과 선택된 계층적인 객체간의 빠른 전환 및 오류의 정정이 가능하며 기존의 마우스 환경을 활용할 수 있기에 활용도가 높다고 할 수 있다. 그러므로 복잡한 객체간의 선택과 전환이 많은 작업에서 유용하게 사용될 수 있다. 특히 계층적인 정보를 활용해서 선택 트리가 구성되므로 계층적인 정보가 많은 전자메뉴얼의 기계 부품 3D 장면이나 골격 애니메이션에서의 골격과 같은 경우 사용된다면 정확한 선택과 계층적인 포커스의 간편한 이동을 통해 조작성을 크게 개선시킬 수 있을 것으로 기대된다.

현재 선택 트리의 생성은 인접성 요소만을 고려하였다. 이 경우에 선택 트리가 구조화되어 제시되지 않기 때문에 사용자에게 혼란을 초래할 가능성이 있다. 따라서 향후에는 다른 요소를 더하여 선택가능그룹 구성의 정확성을 향상시키고, 추가적인 사용자 평가를 통해 문제점을 보완할 필요가 있다.

그리고 정량적인 실험을 통해 기존 방법에 대해 어떤 장점을 갖는지 조사해야 한다. 실험은 각각의 경우

사용자가 지정한 객체를 선택하기까지의 과정을 기록하여 분석하고 사용자가 지정한 객체를 선택하기까지의 시간, 클릭 회수, 마우스 이동 거리를 측정해서 분석한다면 제안된 방법이 기존의 방법에 비해서 어떤 장점을 갖는지에 대한 구체적인 증명이 가능할 것이다. 프로토타입에 실험 기능을 추가해서 보다 체계적인 실험을 수행해야 할 것이다.

[참고문헌]

- [1] 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.
- [2] James D. Foley , Victor L. Wallace , Peggy Chan, The human factors of computer graphics interaction techniques, IEEE Computer Graphics and Applications, v.4 n.11, p.13-48, Nov. 1984
- [3] James D. Foley , Andries van Dam , Steven K. Feiner , John F. Hughes, "Computer graphics: principles and practice (2nd ed.)," Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1990
- [4] Grissom, S., Periman, G. "StEP(3D): A portable discount usability evaluation plan for 3D interaction," Ohio State University, Department of Computer Science and Information Science, Technical Report OSU-CISRC-2/93-TR7, 1993
- [5] Mine, M. "Virtual environment interaction techniques," UNC Chapel Hill Computer Science Tech. Report TR95-018, 1995
- [6] Ivan Poupyrev , Mark Billinghurst , Suzanne Weghorst , Tadao Ichikawa, "The go-go interaction technique: non-linear mapping for direct manipulation in VR," Proceedings of the ACM symposium on User interface software and technology, November 1996
- [7] 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
- [8] Chris Marrin, Bruce Campbell, 이상영 역, "Teach Yourself VRML2 in 21 Days," sams, 인포북, pp.6-9, 15, 1997
- [9] David H. Eberly, "3D Game Engine Design A Practical Approach to Real-Time Computer Graphics," 민프레스 편집부 2001
- [10] Horowitz, Sahni, Anderson-Freed, 이석호 역, "Fundamentals of Data Structures in C" Computer Science Press, pp. 191-210, 1993