# 비몰입형 가상환경에서 효과적인 3D객체선택 인터페이스

한덕수[†] · 임윤호[††] · 최윤철[†††] · 임순범[††††]


## 요 약

3D 가상환경에서의 상호작용기법은 가상공간 사용자의 몰입감과 사실감에 결정적인 영향을 미치는 매우 중요한 요소이다. 특히 데스크탑환경의 전자매뉴얼과 같이 정교한 객체조작이 요구되는 분야에서는 효과적이고 자연스런 객체조작을 위한 상호작용기법이 절실히 요구되고 있다. 본 논문에서는 객체를 구성하는 장면 그래프를 사용자의 선택에 따라 내부적으로 분할, 생성된 선택 후보 리스트를 이용하여 객체의 세부 단위요소를 정확하게 선택하고, 이동 및 회전 등의 조작을 하며 선택을 해제할 수 있는 일련의 조작 기법을 제안한다. 여기엔 각 조작단계마다 다양한 시각피드백 기능을 지원함으로써 객체조작을 보다 효과적으로 수행할 수 있는 기능도 포함된다. 제안된 기법은 정교한 객체의 조작을 필요로 하는 분야의 3D 사용자 인터페이스 구축에 효과적으로 적용될 수 있을 것이다.


# Effective 3D Object Selection Interface in Non-immersive Virtual Environment

Deok-Soo Han[†], Yoon-Ho Lim[††], Yoon-Chul Choy[†††] and Soon-Bum Lim[††††]


## ABSTRACT

Interaction technique in 3D virtual environment is a decisive factor that affects the immersion and presence felt by users in virtual space. Especially, in fields that require exquisite manipulation of objects such as electronic manuals in desktop environment, interaction technique that supports effective and natural manipulation of object is in demand. In this paper, 3D scene graph can be internally divided and reconstructed to a list depending on the users selection and through moving focus among the selection candidate objects list, the user can select 3D object more accurately. Also, by providing various feedbacks for each manipulation stage, more effective manipulation is possible. The proposed technique can be used as 3D user interface in areas that require exquisite object manipulation.


**Key words:** 3D UI, Scene Graph, 3D Manipulation, Desktop VR

## 1. 서 론

급속도로 발전하는 3D 컴퓨터 그래픽스 기술은 과거 고가의 워크스테이션에서나 가능했던 3D 그래픽스 작업을 이제는 일반사용자도 개인용 컴퓨터를 사용하여 할 수 있게 하였다. 이와 같은 3D 그래픽스 기술의 대중화는 3D 그래픽스 기술의 보다 폭넓은 활용을 가능하게 하였다. 과거에는 CAD, 3D 그래픽스 렌더링과 같은 일부 전문적인 분야에서나 사용되던 3D 그래픽스 기술이 점차 교육, 협업, 엔터테인먼트, 전자상거래 그리고 게임과 같은 일반적인 분야에도 사용되고 있다. 최근에는 3D 그래픽스와는 무관한 것으로 여겨지던 일반 사무용 프로그램들에서도 3D 그래픽스가 사용되고 있다[1].

이처럼 점점 많은 사용자들이 3D 그래픽스를 접하게 되었으나 개인용 컴퓨터의 기본적인 입력장치

"이 논문은 2002년도 문화관광부 문화콘텐츠기술지원사업에 의하여 연구되었음"(1-02-2002-001-1008-00-0040)
접수일 : 2002년 12월 2일, 완료일 : 2003년 1월 17일
[†] 정회원, 육군3사관학교 전산정보처리학과 조교수
[††] 준회원, 연세대학교 컴퓨터과학과 박사과정
[†††] 종신회원, 연세대학교 컴퓨터과학과 교수
[††††] 종신회원, 숙명여자대학교 멀티미디어학과 조교수

들은 크게 변화하지 않았다. 물론 특화된 여러 하드웨어들이 개발되어있고 그러한 장치들만의 특유의 조작은 진정한 3D 그래픽스와 상호작용을 가능하게 한다. 그러나 그러한 장치들은 대부분 고가장비들이 대부분이기 때문에 일반적인 개인용 컴퓨터 사용자들이 쉽게 접할 수 없다. 가상현실분야에서도 이와 같은 하드웨어의 제약으로 인해 가상현실을 사용자가 HMD(Head Mounted Display)나 데이터 글러브 등 여러 복잡한 고가의 장비를 장착하고 체험하는 몰입형 가상현실과 그렇지 못한 비몰입형 가상현실로 구분하고 있다. 비몰입형 환경 하에서도 제한적이나마 6DOF(Dgree of Freedom)을 지원하는 장비[2-4]를 사용할 수는 있으나 이러한 장비들도 극히 일부의 전문가들만이 사용할 수 있는 고가의 장비들로서 대중화되지 못하고 있다. 현실적으로 현재 대부분의 컴퓨터 사용자들은 2D 기반의 장치들을 사용하고 있는데 평범한 키보드, 마우스를 통해 2D Display에 투영된 3D장면을 가지고 작업하는 것이 일반적이다. 지금까지의 3D 조작기법관련 연구들을 보면 주로 다양한 하드웨어를 기반으로 하는 몰입형 가상현실 분야에 치중되어있는 현실이다. 그러나 데스크탑 가상환경에서의 3D객체 조작기법은 데스크탑환경의 특성상 입력장치가 주로 키보드와 2D 마우스에 국한되는 하드웨어적 제약과 3D공간상의 객체를 2D상에서 조작함으로써 발생하는 공간 불일치 등의 문제로 효율적인 물체조작 기법이 매우 부족한 현실이다.

따라서 본 논문에서는 일반적인 데스크탑 가상환경에서 수행되는 3D 객체의 선택과 조작에 관련된 내용을 다룬다. 사용자는 최초 어떤 객체를 선택하는 것으로 조작을 시작하므로 객체선택은 모든 조작의 출발점이 되는 기본적인 조작으로 볼 수 있다. 그러나 데스크탑 가상현실에서 3D 객체선택작업은 기존의 마우스 클릭으로 선택하는 방법이 거의 표준처럼 굳어져 왔기 때문에 거의 연구가 진행되어 있지 않다. 그러나 마우스 클릭만을 이용한 기존의 객체선택 방법은 그다지 직관적이지 못하며 정확도도 떨어진다. 본 연구에서는 기존의 방법을 보완하기 위해 객체 선택 시 선택된 객체주변의 연관된 객체들로 선택 후보리스트를 구성하여, 사용자가 선택 포커스를 이동하면서 원하는 객체를 선택할 수 있도록 함으로써 정확도를 높이는 방법을 제안하며 그 아이디어의 타당성을 검증하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서 기존의 3D 객체조작기법들에 대한 소개와 제안하는 기법과 관련된 배경 지식에 대해 서술한다. 3장에서는 본 논문에서 제안하는 3D객체선택기법에 대해 구체적으로 설명한다. 4장에서 제안된 기법을 적용하여 제작된 프로토타입으로 실시한 실험내용을 다루며 5장에서 결론과 향후연구방향에 대해 논한다.

## 2. 관련 연구

### 2.1 3D 객체조작 개요

3D 객체조작은 여러 관점에서 분류할 수 있다. 먼저 Bowman[5]은 가상공간에서의 3D 객체조작을 크게 선택(Selection), 조작(Manipulation), 해제(Release)등 세 가지로 구분하였고, Foley, D. 등 여러 연구자들[6-8]은 컴퓨터 그래픽스 인터렉션을 선택(Selection), 배치(Position), 회전(Orient), 텍스트입력(Text), 숫자입력(Quantify)등의 다섯 가지로 구분하였다. 이를 살펴보면 객체조작기능은 크게 객체의 선택, 회전, 이동, 선택해제로 이루어짐을 알 수 있다. 그 중에서 객체선택기능은 항상 개별 객체조작의 출발점으로 가장 기본적이고 중요한 기능이다.

객체선택 기법은 몰입형 가상현실 분야에서 비교적 많은 연구가 진행되어왔다. 이 몰입형 가상현실 분야에서의 대표적인 객체조작기법들은 'Simple virtual hand', 'Ray-casting'[9], 'Go-go technique'[10] 등이 있고 이외에 'Flash light', 'Aperture', 'Image plan', 'World-in-Miniature' 등 다양한 기법들이 있으며 여러 기법들을 혼합하여 사용하는 형태도 존재한다[10]. 또한 증강현실(AR: Augmented Reality)에서는 각종 센서들, HMD, 데이터 글러브, 각종 트래킹 장비 등 대부분 몰입형 가상현실에서 사용되는 장비들을 이용한 기법들이 주를 이룬다.

위에 언급된 여러 기법들 중에서 특히 'Ray-casting'기법을 보완하기 위해 개발된 'Flash light' 기법에 주목할 필요가 있다. 'Ray-casting'기법은 사용자의 가상손(Virtual hand)에서 뻗어나가는 가상의 광선으로 멀리 있는 객체를 선택하는 기법인데 이는 멀리 떨어진 작은 물체를 선택하기 어렵다는 단점이 있다. 이 는 단지 한 점만을 선택할 수 있는 광선이라는 민감한 메타포이기 때문에 발생하는 문제인데 이 단점에 대한 보완책으로 연구된 기법이

'Flashlight'이다. 이 기법은 일차적으로 마치 손전등을 비추듯 일정 범위내의 여러 객체를 선택한 후 선택된 객체그룹 내에서 다시 한번 정확한 선택을 하는 방법이다. 이 기법은 두 번의 선택을 해야 하는 번거로움이 있으나 객체 선택의 정확도를 높이기 위해 일정범위내의 여러 객체를 일차적으로 선택한 후 그 그룹 내에서 원하는 객체를 정확히 선택을 하는 방법이라는 점은 본 논문에서 제시하고자 하는 기본 개념과 유사하다.

이처럼 몰입형 가상현실에서의 선택 혹은 조작 기법은 다양한 방향에서 연구가 진행되었으나 비몰입형 가상환경, 특히 하드웨어의 제약으로 소프트웨어적인 방법에 의존하는 데스크탑 가상현실 분야에서의 객체선택과 조작에 관련된 연구는 상대적으로 미미한 수준이다. 2D 마우스를 이용한 3D 객체조작에 관련된 기존연구는 Nielson[11]이 2D 마우스를 사용한 객체의 이동에 관한 기술을 제안하였고 K. Shoemake[12]는 객체의 회전 방법을 제시하였으며 Terii Stein[13]은 독특한 매트릭 커서를 이용한 새로운 객체조작기법을 제시하였다. 그러나 이들 모두는 개발자가 객체 모델링 단계에서 필요한 점과 선을 다루는 기술들로 정확한 선택보다는 2D 입력장치의 부족한 축을 매핑해 6DOF로 조작하는데 중점을 둔 기법들로써 본 연구의 주제와는 다소 거리가 있다.

객체조작에서 또 하나의 중요한 요소는 객체조작 상태를 상호 대화적으로 사용자에게 알려주는 피드백기능이다. 사용자는 자신이 하는 행동의 결과를 실시간으로 확인하면서 조작하므로 정확한 조작을 하기 위해서는 적절한 피드백이 필수적이다[17]. Bowman[14]은 피드백의 종류를 'graphical', 'force/tactile', 'audio'로 구분하였다. 이들 중 데스크탑 가상환경에서는 'graphical'에 해당되는 시각피드백이 적절하다. 객체선택 시에도 현재 어떤 객체가 선택되었는지를 사용자에게 알려줄 필요가 있으며 이를 위해, 선택된 객체의 색깔을 바꾸거나 하이라이팅, 혹은 경계상자를 출력하는 방법 등이 대표적인 객체선택과 관련된 'graphical' 피드백이다.

### 2.2 3D 장면 파일의 구조와 장면 그래프(Scene Graph)

본 논문에서 제안하는 3D 선택기법은 3D 파일의 계층적인 구조를 활용하는 경우가 있으므로 대표적인 3D 파일에 대해서 언급할 필요가 있다. 대표적인 표준 3D 파일 포멧으로는 VRML(Virtual Reality Modeling Language)[15,16]이 있다. VRML은 3D 파일 포멧의 표준을 위한 노력으로 제정된 것으로 VRML에 적용된 파일구조는 일반적인 3D 장면 파일의 구조를 조사하는데 기초가 된다. VRML 파일은 장면 그래프(scene graph)라는 형태로 표현된다.

3D 장면은 복잡한 물체들로 구성되며 이런 장면의 복잡도(scene complexity)를 관리하기 위해서는 장면 그래프(scene graph)라는 내부적인 자료 구조와 코드의 생성이 필요하다. 장면 그래프 내의 객체들은 각각 다른 객체와 관계를 가지고 있다. 이러한 계층적인 객체를 나타내기 위해서는 보통 트리(tree)라는 자료구조, 혹은 좀더 일반적인 의미로 그래프(graph)라는 자료구조를 갖는다[16]. 그러나 현실적인 구현에서 장면 그래프는 대부분 트리 구조를 갖는다. VRML의 장면 그래프는 매우 구조적으로 나타나며 트리 구조로 표현된다[17].

VRML이외에도 X3D나 JAVA 3D 등 대부분의 3D 관련 파일 포맷 혹은 라이브러리는 트리구조를 기반으로 한 장면 그래프를 생성한다[18]. 이처럼 복잡한 3D 작업에 있어서 계층적인 구조를 표현하고 속도상의 이점을 얻기 위해 많은 경우 장면 그래프는 트리 형태의 자료 구조를 갖고 있다.

물론, 때에 따라서는 트리 구조를 갖지 않는 형태의 3D 장면도 있을 수 있다. 본 논문에서 제안하는 방법에서 장면 그래프는 트리 구조를 가정하고 그에 따라 최적화되어 있지만, 트리 구조가 아닌 경우, 혹은 장면 그래프의 형식이 전혀 다른 경우라도 복수의 3D 객체들로 구성된 장면이라면 이후에 언급될 제안을 적용할 수 있다.

### 2.3 기존 시스템에서의 선택 방법

기존의 많은 3D 그래픽스 프로그램들 역시 객체선택 기능을 필요로 한다. 특별한 하드웨어를 사용하지 않는 경우 대부분의 객체선택은 마우스에 의존하게 된다. 기존 시스템에서는 크게 두 가지 방법으로 선택을 수행한다. 첫 번째 방법은 직접적인 방법이다. 이것은 직접 3D 객체가 표시된 화면상에서 객체를 클릭하면 클릭된 객체가 선택되는 방법이다. 그러나 간단한 마우스 클릭만으로 객체를 선택하는데는 다음과 같은 문제가 따른다.

우선 사용자가 접하게 되는 문제는 3D 장면이 2D 화면상에 투영된 상태로 작업을 하기 때문에 발생하는 정보의 부족이다. 투영 과정에서 은면제거 알고리즘에 의해 사용자의 시점에 따라 화면의 물체는 서로 겹치거나 가려지고 사용자는 항상 전체 3D 장면의 일부만을 보고 작업을 한다. 사용자가 가려진 물체를 선택하기 위해서는 시점을 바꾸거나 원하는 객체를 가리고 있는 객체를 옮기는 작업을 한 후에야 원하는 객체에 도달할 수 있다.

보다 더 근본적인 문제는 3D 장면과 입력장치간의 오차이다. 실제 3D 장면의 데이터는 통상 정수값이 아니다. 대부분의 경우 float이나 double 등의 부동소수 형태의 실수 값으로 위치정보들이 정의된다. 혹 정수로만 이루어진 물체라 하더라도 회전, 투영 등과 같은 여러 변환과정에서 적용되는 각종 함수들의 영향으로 가 데이터가 정수로만 표현되는 것은 극히 이례적인 경우이다. 하지만 실제 사용자가 사용하는 일반적인 데스크탑 가상현실의 입력장치들은 부동소수 형태의 실수 값을 입력 데이터로 제공하지 않는다. OS나 장치 드라이버 차원에서 지원되는 입력장치의 입력 값은 대부분 정수이다. 마우스의 경우 OS 차원에서 마우스의 입력 값은 마우스 커서가 위치한 곳의 픽셀 좌표이며 드라이버 차원에서도 정수값으로 된 축의위치만을 얻을 수 있다[19]. 그렇기 때문에 전통적인 마우스를 가지고 하는 객체선택의 경우 부동소수점 형태의 3D 장면 데이터를 입력장치가 지원하는 정수 형태의 데이터만으로 처리하는 과정에서 소수점 이하의 정보를 잃어버리게 되어 정확성이 떨어지게 된다. 이는 상당히 밀착되어 있는 객체, 혹은 래스터화 과정에서 단지 픽셀 몇 개 이하를 차지하는 극히 작은 객체의 경우 정확한 선택을 어렵게 한다. 이 경우 화면을 확대하여 실제 객체가 화면에서 차지하는 크기를 늘려서 선택하는 방법을 사용할 수 있지만 이때 사용자는 불편을 느낄 수 있다.

많은 3D 그래픽스 프로그램들은 직접적인 마우스 클릭 외에도 간접적인 선택 방법도 사용할 수 있다. 간접적인 선택 방법이란 객체 그 자체를 선택하는 것이 아니라 객체가 가지고 있는 번호, 이름, 색깔 등 다른 정보를 이용해서 선택하는 방법이다. 이 방법은 정확성의 문제는 해결할 수 있다. 그러나 객체 생성 시에 정확하게 객체의 속성을 부여해야 하며 별도의 다이얼로그 창에서 선택이 이루어지므로 사용자의 시선이 객체에서 떨어져 작업 집중도를 떨어뜨리는 단점이 있다.

## 3. 선택 후보 리스트를 이용한 객체선택기법

본 논문에서 제안하는 3D 객체 선택 방법은 다음과 같다. 우선 초기 객체선택과정은 기존의 방법과 동일하게 일단 사용자가 원하는 객체를 클릭하는 것이다. 그러나 이때 한번의 클릭만으로 단일한 3D 객체가 선택되는 것이 아니라, 시스템이 내부적으로 클릭된 객체를 중심으로 선택 가능한 후보객체들의 집합을 구성하고 그 후보들 사이에서 포커스를 옮길 수 있도록 해준다. 첫 번째 선택과정에서 원하지 않는 객체가 선택된 경우, 사용자는 다시 선택을 위해 커서를 움직여 클릭 할 필요 없이 포커스를 이동시키면서 피드백을 참고로 선택을 원하는 객체에 도달할 수 있다. 이 방법은 선택하기 어려운 가려진 객체나 작은 객체를 쉽게 선택할 수 있다. 하지만 이 아이디어를 적용시키려면 선택 가능한 후보들의 선정 방법과 포커스이동 방법을 결정해야 한다.

### 3.1 선택 후보 리스트

우리가 제안하는 객체선택 방법은 일단 선택된 객체와 더불어 연관된 객체들을 선택 후보로서 전체 장면에서 분리하고 그 객체들 간에 선택 포커스를 이동하게 된다. 이 때 분리된 객체들 사이에 포커스 이동이 이루어지므로 후보 객체들은 전후관계를 가진다. 내부적으로 이들 선택 후보 객체들은 리스트 형태로 관리되며 이 리스트를 선택 후보 리스트라고 정의하였다. 선택 후보 리스트는 장면 그래프의 분리 및 순회와 별도의 선택 후보 리스트의 생성이란 두 가지 방법으로 얻을 수 있다.

VRML, X3D 그리고 Java 3D 등의 많은 3D 장면 파일의 구조, 그리고 그런 파일들이 가지고 있는 장면 그래프 역시 트리 구조를 기반으로 하고 있다. 이처럼 트리 구조를 갖는 계층적인 장면 그래프는 단순히 작은 서브 트리로 분리하는 작업만으로 쉽게 계층적으로 연관된 선택 가능한 후보들의 집합을 얻을 수 있다. 장면 그래프가 트리 형태를 갖는 경우는 분리된 서브 트리를 순회함으로 각 객체에 한번씩 접근하는 선형 리스트가 얻어지며[20] 이것이 바로 선택 후보 리스트가 된다. 장면 그래프가 리스트 형태로

된 경우는 한 쪽 자식 노드만 갖는 트리와 마찬가지로 생각할 수 있으므로 동일한 방법으로 분리가 가능하다.

장면 그래프가 분리하기에 적합하지 않은 구조를 갖는 경우, 혹은 구조를 무시하고자 하는 경우에 선택 후보 리스트는 새로 생성하는 방법으로 얻어지며 이 때는 인접성 등의 다른 요소를 고려해서 새로이 생성하게 된다.

### 3.2 선택 후보 리스트 생성 기법

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

#### 3.2.1 트리 형태로 구조화된 장면 그래프에서 분리 생성

이 방법은 트리 구조의 전체 장면 그래프에서 서브 트리를 분리해서 선택 후보들을 얻어내는 것으로 매우 간단할 뿐만 아니라 전체 장면 그래프의 계층적인 구조 정보를 포함시킬 수 있다는 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 후보 리스트로 만들면 가장 적절할 것이다. 그러나 매우 방대한 트리로 구성된 장면 그래프의 경우는 장면 그래프의 일부분을 분리해서 관리하기 위한 정책이 필요할 수 있다. 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 후보에 포함되어야 한다. 그리고 부모 노드를 선택 후보에 포함시키는 정책을 세운 이유는 선택된 객체의 형제 노드로 이동을 용이하게 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 1에서 칠해진 부분은 이러한 정책에 의해 분리된 선택 후보들로 이루어진 서브 트리 이다.

이러한 장면 그래프로부터 선택 후보 트리를 분리해서 리스트로 만드는 방법은 장면 그래프의 구조적인 정보를 얻어올 수 있으므로 3D 장면이 구조적이고 계층적인 정보를 담고 있는 경우 유용하게 사용될 수 있는 선택 방법이다.

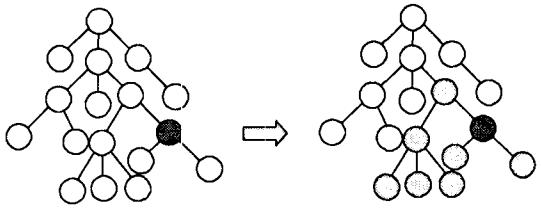


그림 1. 선택 후보들로 이루어진 서브 트리의 분리

#### 3.2.2 인접성 테스트에 의한 생성

이 방법은 선택 후보 리스트를 분리에 의해 얻는 것이 아닌 새로 생성하는 방법이다. 전체 장면 그래프가 분리하기에 적합하지 않거나, 구조적인 정보가 없는, 단순하게 객체들이 나열된 데이터 구조를 갖는 경우, 혹은 전체 장면 그래프의 계층적인 구조를 무시할 필요가 있을 때 사용될 수 있다. 이러한 경우 선택 후보 리스트를 구성하는 방법은 선택작업에서 오류가 발생하는 상황을 토대로 접근하고자 한다.

사용자가 가장 많은 선택 오류를 범하게 되는 경우는 앞서도 언급되었듯이 객체가 상대적으로 아주 작은 크기일 때, 객체들이 매우 가까이 밀착해 있는 경우, 그리고 다른 객체에 의해 가려지거나 다른 객체 안에 들어있는 경우를 들 수 있다. 본 논문에서는 선택 후보 리스트를 생성하는 방법으로 인접성 테스트를 제안한다. 사용자는 원하는 객체 근처를 선택하고자 할 가능성이 높으며 인접해 있는 객체는 서로 연관성을 가질 확률이 크다. 그러므로 인접성은 장면 그래프에서 선택 후보들을 추출해 내기 위한 좋은 출발점이 된다.

**(1) 객체간의 충돌**

가장 간단히 생각해 볼 수 있는 방법은 선택된 객체에 근접한 다른 객체들을 선택 후보에 포함시키는 방법이다. 3D 상에서 객체간의 충돌을 검사하는 방법은 충돌하는 객체 유형에 따라 여러 가지가 있을 수 있지만 정교한 충돌 검사는 매우 비용이 많이 드

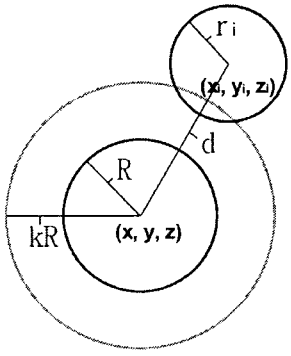
는 처리이며 본 논문의 경우 대략적인 인접 정도만 측정하는 것이므로 모든 점이나 면에 대한 충돌 검사는 필요하지 않다. 충돌 검사는 계산의 복잡성으로 인해 시뮬레이션과 같은 아주 정교한 충돌 검사가 필요하지 않은 경우는 간략화 된 경계 볼륨과의 충돌을 검사하는 방법이 주로 사용된다. 경계 볼륨은 구, 상자, 실린더, 타원체 등 여러 종류가 존재할 수 있으며 각각의 경우 다양한 알고리즘들이 이미 연구 되어 있다[1]. 본 논문에서는 가장 간단한 방법인 경계구(Bounding sphere)끼리의 충돌을 사용해서 인접성 테스트를 수행하는 프로토타입을 작성하였다. 경계구간의 충돌을 검사하는 방법은 매우 간단하다.

경계구간의 충돌은 간단하게 점과 점 사이의 거리로 측정될 수 있다. 두 객체 사이의 중심점의 거리는 다음 식(3.1)에 의해 구해진다.

$$d=\sqrt{(x-x_i)^2+(y-y_i)^2+(z-z_i)^2} \quad (3.1)$$

이 때 $kR+r_i \geq a \quad (k \geq 0)$를 만족하면 두 객체의 경계구는 충돌한 것이다. 그러나 반드시 객체의 경계구가 충돌할 때만을 처리한다면 사용자가 의도한 결과가 나오지 않을 수 있기 때문에 k라는 상수를 추가하여 경계구의 크기를 조절하여 선택 후보 리스트 생성의 민감도를 조절하도록 하였다.

경계구의 충돌은 가장 간단하게 충돌을 검사할 수




- (x, y, z) : 현재 클릭 된 지점의(최초 선택된) 객체의 중심
- R : 선택한 객체의 Bounding sphere의 반지름
- $(x_i, y_i, z_i)$ : 인덱스가 i 인 장면내의 객체의 중심
- $r_i$ : 인덱스가 i 인 객체의 Bounding sphere의 반지름
- d : 처음 선택된 객체와 i 인 객체와의 Bounding sphere의 중심간의 거리
- k : 상수

그림 2. 경계구 간의 충돌 검사

있는 방법인 반면, 가장 부정확한 결과를 갖는 방법이다. 실제 길고 가는 형태의 객체의 경우 긴 쪽에 맞춰서 경계구의 반지름이 정해지므로 실제 크기는 작은 객체라 하더라도 매우 큰 경계구를 가질 수 있어 의도하지 않았던 결과가 나올 수 있다. 그런 경우 k의 값을 작게 조절하면 부분적으로나마 해결할 수 있으나 근본적으로는 보다 복잡한 충돌 검사를 수행하는 알고리즘들을 수행하여 보완할 수 있다. 축에 정렬된 직육면체, 혹은 물체를 감싸는 최소의 직육면체와 같은 보다 복잡하지만 정확한 충돌검사[1]를 수행한다면 좀더 정확한 결과를 얻을 수 있다. 그러나 이러한 객체간의 충돌 검사를 이용한 방법은 다른 객체에 깊숙하게 가려진 객체를 선택하려 한 경우 원하는 객체가 선택 후보에 포함되지 않을 수 있다. 따라서 객체간의 충돌검사를 이용한 선택 후보 리스트 생성법은 객체가 그렇게 많지 않은 경우에 사용하기 적절하다.

**(2) View vector와의 충돌**

깊숙하게 가려져 있는 객체에 접근하기 위해서는 단순한 객체간의 인접성 테스트로는 불가능하다. 물론, 상수 k를 큰 값으로 대치하면 깊숙하게 위치한 객체도 선택 후보에 포함시킬 수 있으나 그런 경우 선택 후보들의 집합이 방대해져 원하지 않는 객체들이 선택 후보에 포함될 수 있으므로 비효율적이다. 그에 대한 대안을 마련하기 위해 사용자의 행동을 분석할 필요가 있다. 우선 사용자는 커서가 위치한 곳의 객체를 선택하기를 원한다. 이것은 정확히 말하면 사용자의 커서가 있는 투영면의 위치로 투영된 객체를 선택하기를 원하는 것이다. 그러므로 그 지점에 투영된 객체들은 선택하고자 하는 객체일 가능성이 높다. 따라서 두 번째 인접성 테스트 방법으로 커서에서 화면 안쪽으로 뻗어나가는 직선 성분과 객체간의 충돌 검사에 의한 방법을 제안한다. 이 방법의 경우 View vector와 객체간의 거리를 구할 필요가 있다. 그 방법을 설명하면 다음과 같다.

View vector를 $\vec{A}$ 라고 하고 i번째 객체의 중심점 $(x_i, y_i, z_i)$까지를 $\vec{B}_i$ 로 표현한 경우 중심점 $(x_i, y_i, z_i)$에서 $\vec{A}$ 에 수직으로 그은 직선과의 교점 bk $\vec{p}_i$ 는 View vector와 객체사이에 가장 가까운 점이다. $\vec{p}_i$ 는 다음 식(3.2)으로 구할 수 있다.

$p_i$를 구했으면 이제 $B_i$와의 거리를 구하여 인접성을

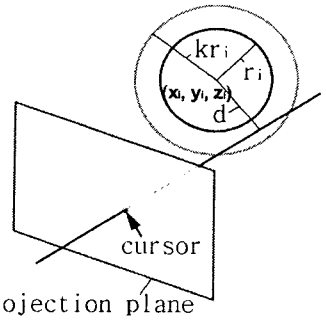



그림 3. View vector와 객체간의 충돌 검사

$$\vec{p}_i = \left(\frac{\vec{A} \cdot \vec{B}_i}{\vec{A} \cdot \vec{A}}\right) \vec{A} \qquad (3.2)$$

테스트할 수 있다. 이 경우는 $kr_i \geq a$ 를 만족하는 경우 인접성 테스트를 통과한 것으로 볼 수 있다. k는 첫 번째 방법과 같은 상수로 민감도를 설정하는 것이다. 물론 설명한 방법은 직선과 경계구간의 충돌이며 보다 정확한 검사를 하고자 한다면 직선과 객체간에 보다 복잡한 형태의 충돌 검사법들[1]을 동원하면 가능하지만 본 논문의 경우 간단한 인접 여부만 검출하는 것이므로 보다 복잡한 방법을 사용할 필요는 없다고 본다. 이 방법은 첫 번째 방법에 비해 복잡하고 많은 객체들이 존재하는 경우 유효하다.

위와 같은 방법에 의해 선택과 관련된 인접성을 테스트할 수 있다. 인접성 테스트를 위한 알고리즘을 선택했다면 우선 화면 내의 모든 객체들에 대해서 선택한 방법으로 인접성 테스트를 수행한다. 그리고 테스트를 통과한 객체들을 모아서 리스트로 만들면 이것이 선택 후보 리스트가 된다. 그러나 이 방법은 객체들이 많은 경우 비교할 회수가 늘어나며 최종 결과물의 순회 순서 객체의 내부 인덱스 번호와 같은 검색 순서로 결정되므로 직관적이 아닌 순서를 갖는 리스트가 생성된다는 단점이 있다. 이러한 단점을 극복하기 위해서 전처리로 인접성 검사를 할 가능성이 전혀 없는 객체들을 미리 분리해 내고, 후처리로 선택 후보 리스트의 순서를 사용자가 예상할 수 있도록 조작해 줄 필요가 있다. 선택 후보 리스트의 순서를 완전히 자동적으로 생성하는 것은 어렵지만 비교적 직관적으로 보이게 표현할 수 있는 방법은 인접성 테스트를 통과한 객체들을 z축 방향으로 정렬하여 제시하는 것이다. 이것은 사용자의 시점에서 가까운 것부터 포커스를 맞추도록 순서를 조정하므로 사용자에게 보다 직관적으로 보일 수 있다.

### 3.3 포커스 이동 법

#### 3.3.1 리스트의 객체 순서의 결정

우선 전체 장면 그래프에서 분리해서 선택 후보 리스트를 얻는 경우 Focus의 이동 순서의 결정은 선

표 1. 각 선택 후보 리스트 구성 방법의 특징

| | 장면 그래프에서 분리 | 인접성 테스트 | |
|---|---|---|---|
| | | 객체간의 충돌 | View vector와의 충돌 |
| 적용 예 | 장면 그래프의 구조적인 특성을 이용할 필요가 있는 경우<br>- IETM, 애니메이션 데이터 | 객체간의 계층적 관계가 미비하거나 무시해야 하는 경우 - 3D 객체 모델링 작업 등 | |
| 장점 | 계층 간의 구조를 이용할 수 있다.<br>단순히 분리만 하면 되므로 간단하다.<br>각 객체간의 구조가 명확히 드러난다. | 객체간에 계층적인 구조가 필요 없다.<br>가장 간단하게 구현 가능하다. | 객체간에 계층적인 구조가 필요 없다.<br>깊게 가려진 객체를 선택후보리스트에 포함 가능하다. |
| 단점 | 반드시 장면 그래프가 계층적으로 만들어져 있어야 한다.<br>선택 후보 리스트를 분리하는 정책이 필요하다. | 오차가 심하다.<br>깊게 가려진 객체를 선택하기 어렵다.<br>정교한 인접성 테스트 알고리즘 적용시 속도가 느리다.<br>객체의 순회 순서가 구조적이지 못하다. | 객체간의 충돌 방법에 비해 상대적으로 복잡하다.<br>객체의 순회 순서가 구조적이지 못하다. |
| 보완책 | 장면 구성 시 체계적인 구조화 | 상수 k의 조절<br>다른 충돌 검사 알고리즘의 적용 | |

택 후보들로 이루어진 서브 트리의 순회(traversal)를 통해 얻을 수 있다. 이 방법은 선택 후보들의 트리를 전체 장면 그래프로부터 분리하는 경우에만 고려대상이 된다. 트리 구조인 경우 순회 방법은 전위(preorder), 중위(inorder), 후위(postorder) 순회방법, 그리고 트리의 레벨 순서 순회방법이 있다[20]. 이 경우 객체의 계층적인 제시를 가능하게 해 주는 순회방법은 전위와 깊이에 따른 순회방법이다. 다른 순회 방법들은 부모, 상위 노드가 나중에 제시되는 순회방법이라서 사용자에게 계층적인 순회방법으로 보이지 않을 수 있다.

전위순회방식은 항상 부모 다음에 한쪽 자식 노드를 전부 둘러 본 뒤 부모의 다른 자식 노드 방향으로 순회하므로 계층적인 구조를 쉽게 알 수 있게 해 준다. 그러나 자식 노드나 형제 노드의 수가 많은 경우 부모로 돌아가기가 번거로울 수 있다. 상위로 올라갈 때는 깊이에 따른 방법을 이용해서 바로 부모 노드로 돌아간다면 보다 빠르게 탐색할 수 있다. 그러나 어떤 종류의 순회를 사용할 것인지 여부는 사용자가 상황에 따라 설정할 수 있도록 배려하는 것이 좋을 것이다. 결정된 방법으로 순회를 마치면 그 결과는 선택 후보 리스트가 된다.

#### 3.3.2 하드웨어 매핑

제안된 방법에서 마지막으로 고려할 점이며 가장 중요하다고 할 수 있는 점은 과연 그런 순회의 조작을 실제 입력장치의 어떤 조작에 매핑시킬 것인가 하는 문제이다. 이 방법을 성공적으로 적용하려면 기존의 조작 방법과 중복되지 말아야 하며 사용자가 직관적으로 받아들일 수 있는 메타포여야 한다. 사용자에게 지나치게 복잡하거나 새로운 조작을 요하거나 아예 별도의 창에 리스트의 정보를 제시하는 간접적인 방법은 제안된 방법의 장점을 살리지 못하는 것이다. 과거의 마우스는 단순한 2D 위치 지정 기능과 1, 2개의 버튼으로 이루어진 것이 전부였으나 현재 유통되는 많은 마우스에 해당하는 입력장치들은 보다 융통성 있는 입력을 받을 수 있다. 실제 Windows에서 Direct Input API를 통한 마우스 접근 시 기본적으로 3개의 축과 8개의 버튼을 지원하고 있으며[19], 최근의 마우스들은 제3의 축과 버튼인 휠이 기본적으로 탑재되어 있고 고급기종의 경우 4번째, 5번째 버튼이 마련되어 있어 이를 사용하는 방법이 가장 효과적이라 할 수 있다. 각각의 방법에 대한 장단점을 정리하면 표 2와 같다.

표 2. 조작법에 따른 장단점

| | 마우스 휠 | 4, 5번 버튼 | 키보드와 연계 기타 Gesture |
|---|---|---|---|
| 장점 | · 가장 직관적<br>· 현재 많은 마우스에서 지원됨<br>· 조작 자체가 방향성을 가지므로 메타포에 부합 | · 다른 조작과 충돌할 가능성 희박<br>· 비교적 직관적이나 새로운 개념 | · 모든 상황에서 적용 가능 |
| 단점 | · 화면 스크롤이나 확대기능 등 기존의 조작과 충돌 가능성 | · 특정 하드웨어에서만 사용 가능 | · 양손을 사용하거나 추가적인 조작을 기억해야 함 |

## 4. 프로토타입의 구현 및 실험

본 논문에서 제안한 기법을 적용한 간단한 프로토타입을 작성하여 실험하였다. 프로토타입은 OpenGL과 Visual C++을 사용하여 개발되었으며 3장에서 제안된 3가지의 선택 후보 리스트 구성 방법을 모두 테스트해 볼 수 있다.

### 4.1 프로토타입의 구현

본 논문에서 제안한 선택 시스템의 프로토타입 구현은 OpenGL[16,21,22]과 Windows 환경의 Visual C++와 MFC[19]를 이용하였다. 실험에 사용된 3D 모델은 최종적으로는 3가지로 구조가 간단한 엔진의 피스톤 모델, 자동차 엔진[19] 그리고 테스트 목적의 수십 개의 Box로 이루어진 3D 장면이 사용되었다. 프로토타입은 실험목적으로 별도의 파일 포멧을 사용하고 있지만 개념적으로는 VRML이나 X3D에 충분히 적용할 수 있도록 만들어졌다.

프로토타입의 시각적 피드백은 색깔과 크기, 투명도를 이용했다. 실험을 위해 모델을 읽어들일 때 재질 정보는 무시되어 흰색으로 통일되어 표시되며 선택된 객체는 붉은 색으로, 선택 후보 리스트는 투명하게 처리된 붉은 색으로 표시하였다. 선택되지 않은 객체는 투명하게 처리하여 선택된 객체들을 명확하

그림 4. 프로토타입

게 표시하였고 선택된 부분을 화면 중앙으로 이동하고 윈도우 크기에 맞춰 확대하여 쉽게 선택한 부분을 확인할 수 있도록 구성하였다.

4.1.1 장면 그래프에서 분리하는 경우

이 경우 사용된 피스톤은 각각의 피스톤마다 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. 그림 5는 피스톤의 암 부분을 클릭한 경우로 피스톤 1개가 선택 후보 리스트로 분리되었고 현재 클릭한 객체에 Focus가 맞춰져 있다. 이때 마우스 휠을 1단 사용자 방향으로 돌리면 다음과 같이 분리된 서브 트리의 전위 순회방향으로 자식 객체가 선택되게 된다.

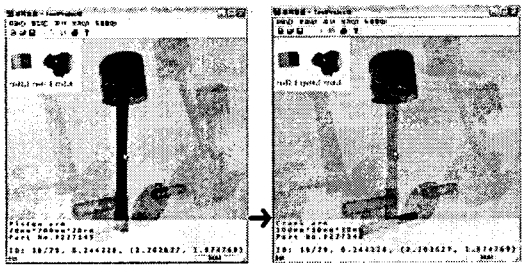


그림 5. 선택 후 마우스 휠 1단 사용자 방향으로 돌린 경우

반대로 마우스 휠을 1단 사용자 반대 방향으로 돌리면 다음과 같이 분리된 서브 트리의 전위 순회방향으로 부모 객체가 선택되게 된다.

이러한 선택 후보들의 트리를 전체 장면 그래프에서 분리해서 선택 후보 리스트를 생성하는 과정은 앞서 설명된 대로 구조가 명확한 장면의 객체를 선택

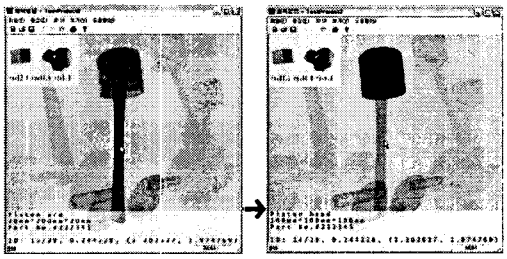


그림 6. 선택 후 마우스 휠 1단 사용자 반대 방향으로 돌린 경우

하는 경우 매우 유용하다. 사용자는 객체의 포커스 순서를 통해 자신이 선택한 선택 후보들 내에서의 관계를 쉽게 인지할 수 있으며 선택 과정만을 통해서도 객체의 구조를 상당부분 파악할 수 있다.

4.1.2 인접성 테스트를 통해 선택 후보 리스트를 생성하는 경우

객체들간의 경계구의 충돌 검사를 사용한 인접성 테스트를 사용한 결과는 다음과 같다. 여기서는 피스톤 헤드를 클릭한 경우이다. 피스톤 헤드와 인접된 객체들이 선택 후보 리스트로 선택되어진 것을 볼 수 있다.

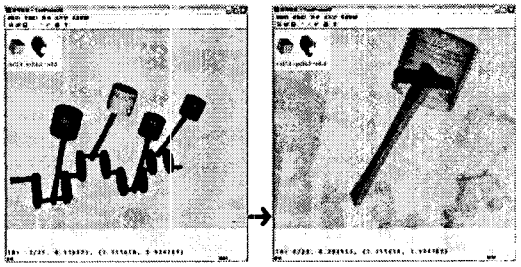


그림 7. 객체간 충돌에 따른 선택 후보 리스트 구성 결과

그러나 이 방법에 의한 단점은 충돌 검사의 오류이다. 그림 8은 피스톤 헤드를 클릭한 경우인데, 피스톤 암의 경계구가 객체의 형태로 인해 매우 크게 나타나 예기치 않은 좌우의 다른 피스톤의 암도 선택 후보 리스트에 포함되었다.

View vector와의 충돌 검사를 사용한 경우는 다음과 같이 나타났다. 이 방법은 여러 객체가 겹쳐있는 경우 사용하기 위한 방법으로 다음과 같은 시점에서 선택한 경우 그림 9와 같은 결과를 보여 깊숙하게 위치한 객체들까지 선택 후보에 포함되었음을 볼 수 있다.

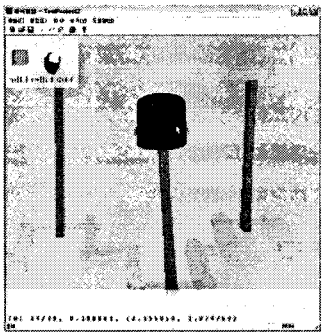


그림 8. 경계구 충돌 검사의 오류

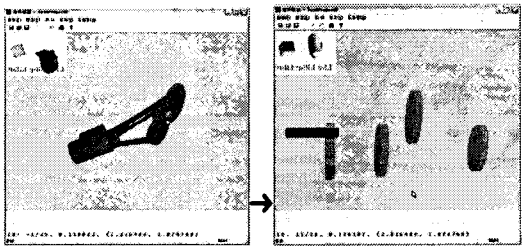


그림 9. View vector와의 충돌 검사에 의한 선택 후보 리스트 구성 결과

이 방법은 경계구 충돌 검사보다 복잡하지만 장면에 따라서 유용하게 사용될 수 있다.

### 4.2 실험

본 논문에서는 제안된 방법에 대한 프로토타입을 사용자 질의응답을 통해 평가하였다. 10명의 3D 그래픽스 경험자와 5명의 미경험자로 구성된 총 15명의 평가자들에게 프로토타입을 5분의 설명을 거쳐 사용법을 숙지시킨 후 테스트를 수행하였다. 테스트는 기존의 직접 선택 방법과 본 논문에서 제안한 방법을 차례로 수행하였으며 평가자들은 다음 문항에 대해 답하였다.

설문 결과는 다음 표 4와 같다. 평균은 작을수록 긍정적인 평가를 내린 것이다. 평가자들의 제안된 인터페이스에 대한 만족도는 전체적으로 비교적 높게 측정되었다. 1, 2, 6번 문항이 비교적 만족도가 낮게 측정되었으며 2, 7번 문항이 긍정을 답한 비율이 가장 낮았다. 3, 4, 5번 문항은 평균 2이하의 높은 만족도를 보이고 있다. 문항 3, 4, 5인 조작단계마다 적절한 피드백의 여부와 작업의 시작과 종료의 명확성, 조작의 단순성은 높은 긍정을 보여주어 제안된 인터페이스가 명확하며 단순한 조작을 보이는 것으로 분석되었다. 피드백 여부는 프로토타입의 구현과 관련된 문제로 프로토타입은 크기, 색깔, 투명도의 3가지 요소를 동시에 제시하였기 때문에 높은 긍정을 보인다. 조작 방법을 익히는데 대한 추가적인 기억 부담, 조작 기능 수행상의 일관성, 잘못 조작한 경우 복구의 용이성에 대한 1, 2, 7 번 문항의 경우 긍정을 답한 비율이 상대적으로 낮았다. 이는 기존의 3D 그래픽스 프로그램 경험자와 그래픽스 프로그램을 처음 사용해본 평가자 사이에 의견이 다소 다르게 나왔기

표 4. 설문 결과

| 질문 번호 | 평균 | 2 이하의 긍정을 답한 비율 |
|---|---|---|
| 1 | 2.13 | 73% |
| 2 | 2.13 | 67% |
| 3 | 1.67 | 80% |
| 4 | 1.87 | 80% |
| 5 | 1.87 | 73% |
| 6 | 2.13 | 73% |
| 7 | 2.00 | 67% |

표 3. 설문 문항

| 번호 | 질 문 | 점 수 |
|---|---|---|
| 1 | 조작방법을 익힐 때 추가적인 기억의 부담이 없는가? | 그렇다 ① ② ③ ④ ⑤ 아니다 |
| 2 | 조작기능을 수행하는 과정은 일관성이 있는가? | 그렇다 ① ② ③ ④ ⑤ 아니다 |
| 3 | 조작단계마다 적절한 피드백이 제공되었는가? | 그렇다 ① ② ③ ④ ⑤ 아니다 |
| 4 | 목적한 작업의 시작과 종료를 명확히 알 수 있는가? | 그렇다 ① ② ③ ④ ⑤ 아니다 |
| 5 | 조작은 충분히 단순하다고 생각되는가? | 그렇다 ① ② ③ ④ ⑤ 아니다 |
| 6 | 잘못 조작할 가능성이 적다고 생각되는가? | 그렇다 ① ② ③ ④ ⑤ 아니다 |
| 7 | 잘못 조작한 경우 복구하기가 쉽다고 느껴지는가? | 그렇다 ① ② ③ ④ ⑤ 아니다 |

때문으로 분석되었다. 이미 3D 그래픽스 프로그램을 경험해본 사용자는 이미 알고 있는 방법에 추가적으로 간단한 보완이 더해진 본 논문에서 제안하는 인터페이스에 긍정적인 답변을 한 반면에 처음 3D 그래픽스 프로그램을 다룬 평가자들은 다소 혼란스러워 하는 것으로 분석되었다.

하지만 전체적으로는 높은 만족도를 보였기 때문에 실험 결과 기존 방법으로 선택하기 어려운 객체에 보다 효율적으로 접근할 수 있는 방법이라는 가능성을 얻었다. 만일 이 방법이 적용된다면 데스크탑 가상환경에서의 3D 객체를 다루는 모든 응용프로그램에서 사용할 수 있을 것이다. 특히 활용이 기대되는 분야는 기존의 문서형태로 된 기술교범들을 디지털 형태로 제작, 관리, 활용하여 정비사가 작업에 필요한 정보를 컴퓨터를 통해 운영, 정비, 교육하기 위한 전자매뉴얼(IETM)이나 웹 기반 교육 및 훈련(WBI, WBT)분야이다. 이런 분야에는 복잡하고 정교하게 3D로 모델링된 3D 객체가 포함될 수 있다. 또 다른 분야는 애니메이션 분야이다. 요즘 대부분의 애니메이션은 Seamless한 캐릭터의 Skin밑에 골격을 삽입하는 골격 애니메이션 방식이 많이 사용된다. 캐릭터의 골격은 인간의 경우 골반을 루트로 한 트리 형태의 구조를 갖는다. 이 경우도 본 논문에서 제안하는 선택기법을 사용한다면 보다 정확하고 빠르게 계층적인 골격 선택을 가능하게 하여 애니메이션을 제작하는데 편의를 도모할 것으로 기대된다. 그 외에도 일반적인 3D 모델링을 위한 어플리케이션도 많은 선택 작업이 이루어지므로 본 기법을 적절히 적용한다면 효율적인 작업을 수행하는데 많은 도움이 될 것으로 생각된다.

## 5. 결론 및 향후 연구 방향

본 논문에서 제안한 기법은 데스크탑환경에서의 새로운 객체 선택 및 조작방법으로, 복잡한 장면의 작은 객체까지도 손쉽게 선택하여 조작할 수 있는 장점이 있다. 기존의 선택 기법의 경우 작거나 밀착해 있는 객체를 선택하는데 어려움이 따르며 정확한 선택을 위한 간접적인 선택방법 역시 작업 집중도를 떨어뜨리는 단점이 있다. 그러나 본 기법은 기존의 선택 기법들의 그러한 단점들을 보완하여 장면 내에서 직접적으로 객체를 정확하게 선택할 수 있고, 선택 포커스를 빨리 이동할 수 있으며 구조적인 정보를 이용한 구조적인 선택이 가능하다. 앞서 언급된 정밀한 조작이 필요한 IETM이나 구조적인 선택이 필요한 캐릭터 애니메이션 데이터 조작과 같은 응용 분야의 경우 특히 조합하게 사용될 수 있을 것이다.

특히 기존의 일반적인 2D 입출력장치들에 본 기법을 매핑시킬 수 있으므로 특별한 추가적인 하드웨어 없이도 이 방법을 적용할 수 있다. 사용자는 기존의 마우스, 키보드와 2D 디스플레이 환경에서 본 제안된 기법을 원활하게 사용할 수 있으며 보다 향상된 조작성을 제공받을 수 있다. 개발자의 입장에서도 새로운 하드웨어를 지원하는 대신 입력 부분의 간단한 기능 추가로 기존의 응용프로그램에서 지원 가능성을 갖게 되어 보다 향상된 조작성을 사용자에게 제공할 수 있다. 하드웨어 매핑은 앞서 언급된 대로 다양한 방법이 존재한다. 본 논문에서 제작한 프로토타입의 경우 마우스 휠을 사용하고 있는데 휠이 가장 추천할 만한 방법이라고 할 수 있다. 만일 휠을 사용하지 못하는 경우라면 다른 제스쳐를 정의해야 할 것이다. 그러나 이 경우 제스쳐를 사용자가 기억해야 하므로 이 기법의 장점을 살리기 힘들 것이다.

현재 선택 후보 리스트의 생성은 인접성 요소와 z좌표의 크기만을 고려하였다. 이 경우에 선택 후보 리스트가 정확히 구조화되어 제시되지 못할 가능성이 있다. 따라서 향후에는 인접성 요소에 객체크기요소를 더하여 선택가능그룹 구성의 정확성을 향상시키고, 추가적인 사용자 평가를 통해 문제점을 보완할 필요가 있다. 객체크기요소의 경우는 단순히 경계 볼륨의 반지름과 같은 길이 값을 택한다면 본 논문에서 테스트한 경계구의 충돌 검사와 비슷한 심한 오차를 갖게 되므로 그보다는 경계 볼륨을 최소화하여 부피로 크기를 결정짓는 것이 적절하다. 프로토타입의 경우 축에 정렬된 경계 직육면체를 이미 계산해 두고 있으므로 그것의 부피를 사용한다면 비교적 효과적인 객체간의 크기비교가 가능할 것이고, 객체의 크기를 고려해서 선택 후보 리스트를 구성하고 선택 후보 리스트의 순회 순서를 결정한다면 사용자에게 보다 직관적으로 느껴질 수 있을 것이며 이에 대한 검증이 필요하다.

그리고 현재의 방법은 다중객체의 선택을 고려하지 않았으나 향후 다중객체 선택 방법에 이를 응용하는 방안 역시 연구가 필요하다. 더 나아가 이를 다른

객체조작 기법과 접목하여 보다 직관적이고 자연스런 3D 사용자 인터페이스로 개선시킴으로써 다양한 분야에 적용시킬 수 있도록 발전시킬 필요가 있다. 그리고 이 방법을 보다 확장한다면 객체의 선택만이 아닌 점, 선, 면 등의 3D 기초물 선택에 응용하거나 3D 메뉴와 같은 다른 분야에 적용할 수 있을 것이다.

## 참 고 문 헌


[1] David H. Eberly, "*3D Game Engine Design A Practical Approach to Real-Time Computer Graphics*," 민프레스 편집부, 2001.

[2] Spaceball, Spacemouse, http://www.alsos.com

[3] ErgoPoint 3D, ITU Research, http://www.ituresearch.com, 2001.

[4] Bernd Froehlich and John Plate, "The Cubic Mouse A New Device for Three-Dimensional Input," CHI 2000, pp. 526, 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.

[6] James D. Foley, Victor L. Wallace, and Peggy Chan, "The human factors of computer graphics interaction techniques," IEEE Computer Graphics and Applications, v.4, n.11, p.13-48, Nov. 1984.

[7] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes, "Computer graphics: principles and practice (2nd ed.)," Addison-Wesley Longman Publishing Co., Inc. Boston, MA, 1990.

[8] Grissom, S. Periman, and 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.

[9] M. Mine, "Virtual environment interaction techniques," UNC Chapel Hill Computer Science Tech. Report TR95-018, 1995.

[10] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and 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.

[11] 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.

[12] K. Shoemake, "ARCBALL : A user interface for specifying three-dimensional orientation using a mouse," Proc. of the Graphics Interface '92, pp. 151-156, 1992.

[13] Terii Stein and Sabine Coquillart, "The Metric Cursor," Proc. of the Eighth Pacific Conference, IEEE 2000, 2000.

[14] Doug A. Bowman, etc al, "3D User Interface Design: Fundermental Techniques, Theory, and Practice," SIGGRAPH 2000 Course Notes, 2000.

[15] VRML, http://www.vrml.org/

[16] Richard S. Wright and Jr. Micheal Sweet, "*OpenGL Super Bible 2nd Edition,*" 2001.

[17] Chris Marrin and Bruce Campbell, 이상영 역, "*Teach Yourself VRML2 in 21 Days,*" sams, 인포북, pp.6-9, 15, 1997.

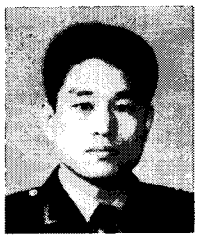
[18] Jon Barrilleaux, "*3D User Interface with JAVA 3D,*" Manning, pp.217, 2001.

[19] Microsoft Developer Network, http://msdn.microsoft.com

[20] Horowitz, Sahni, and Anderson-Freed, 이석호 역, "*Fundermentals of Data Structures in C,*" Computer Science Press, pp. 191-210, 1993.

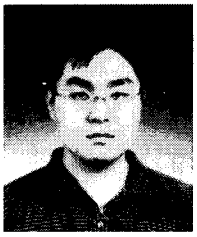
[21] Kevin Hawkins, Dave Astle, and Andre LaMothe, 류광 역, "*OpenGL Game Programming,*" 정보문화사, 2002.

[22] Nehe Production OpenGL Tutorials, http://nehe.gamedev.net



**한 덕 수**

1988년 금오공과대학교 전자공학과 (공학사)
1998년 대구가톨릭대학교 전산통계학과 (이학석사)
2003년 연세대학교 컴퓨터과학과 (공학박사)
1992년~현재 육군3사관학교 전산정보처리학과 조교수
관심분야 : 가상현실, HCI, Web Based instruction



**임 윤 호**

2000년 연세대학교 기계전자공학부 졸업(공학사)
2002년 연세대학교 컴퓨터과학과 (공학석사)
2002년~현재 연세대학교 컴퓨터과학과 박사과정
관심분야 : 컴퓨터그래픽스, 가상현실, 게임

**최 윤 철**

1973년 서울대학교 전자공학과 (공학사)
1975년 Univ. of Pittsburgh (공학석사)
1979년 Univ. of California Berkeley Dept. of IE & OR (공학박사)
1979년~1982년 Lockheed사 및 Rockwell International 사 책임연구원
1982년~1984년 Univ. of Washington 전산학과 박사과정
1990년~1992년 Univ. of Massachusetts 연구교수
1984년~현재 연세대학교 컴퓨터과학과 교수
관심분야 : 멀티미디어 문서처리(SGML/XML), 컴퓨터 그래픽스, 가상환경, Web Based instruction

**임 순 범**

1982년 서울대학교 계산통계학과(학사)
1983년 한국과학기술원 전산학과(석사)
1992년 한국과학기술원 전산학과(박사)
1989년~1992년 (주)휴먼컴퓨터 이사/연구소장
1992년~1997년 (주)삼보컴퓨터 부장
1997년~2001건국대학교 컴퓨터과학과 조교수
2001년~현재 숙명여자대학교 멀티미디어학과 조교수
관심분야 : 컴퓨터 그래픽스, 멀티미디어 응용, 전자출판 (폰트, 전자책, 멀티미디어 출판)

교 신 저 자

한 덕 수 120-749 서울시 서대문구 신촌동 134번지 연세대학교 컴퓨터과학과 멀티미디어/그래픽스 연구실