2001년 한국정보처리학회 춘계 학술발표논문집 제8권 제1호

웹 기반 협동설계시스템에서 오브젝트의 효과적인 Picking

윤보열° 송승헌 김응곤
순천대학교 컴퓨터과학과
byyoon@maesan.org° song9232@shinbiro.com kek@sunchon.ac.kr

Effective Picking Object in Web Based Collaborative Design System

Bo-yul Yoon° Seung-heon Song Eung-kon Kim
Dept. of Computer Science, Sunchon National Univercity

요약

웹 기반 협동설계시스템은 인터넷망과 웹브라우저를 이용하여 공유된 가상 공간에서 협동작업이 이루어지도록 한다. 이때 공유 오브젝트는 3D 도형이 되며, 사용자가 임의의 오브젝트를 선택하여 조작하기 위하여 picking이 효과적으로 이루어져야 한다. 본 논문에서는 오브젝트의 picking이 마우스 포인터에서의 ray와 오브젝트간에 intersection을 계산하는 방법 외에 Java 3D API를 이용하여 scene graph의 노드에 picking 속성을 주는 방법, bounds를 설정하는 방법, picking test의 범위를 한정하는 방법을 사용하여 computation의 부담을 줄이고 효과적인 picking이 이루어지도록 한다.

1. 서론

전통적인 협동작업은 일정한 시간에 일정한 장소에서 함께 만나 자료를 보고 서로 의견을 말하면서 진행되었다. 오늘날에는 컴퓨터와 통신 기술의 발달로 시간과 공간의 제약 없이 공유된 가상 공간에서 상호작용을 하면서 효율으로 작업하는 새로운 시스템이 대두되고 있다[1,2].

지금까지 개발되어 이용해온 협동작업시스템은 전자우편을 비롯하여 화상회의, 공동프로그래밍, 전자결재, 원격교육, 원격 진료 등이 있다. 그러나 대부분의 시스템은 특정한 플랫폼과 그룹웨어를 사용하여 폐쇄적으로 공동작업이 이루어지는 경우가 많고, 윈도우의 탐색기 형태를 취하고 있어서 공유 오브젝트가 폴더나 문서, 메모 등에 그치고 있다[3].

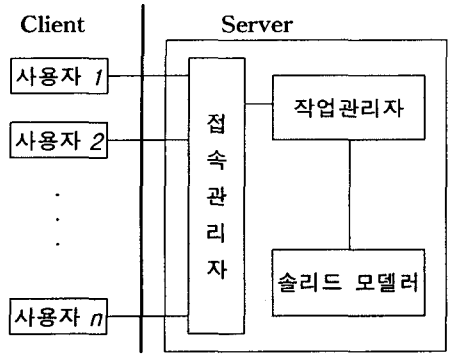
우리가 연구한 협동설계시스템은 인터넷 상에서 웹브라우저를 통해 서버에 접근하여 협동작업이 이루어지도록 되어 있으며, 가상공간에서 협동설계작업을 수행하기 때문에 공유 오브젝트가 3D 도형이 되고 있다[3]. 따라서 다수의 사용자가 임의의 공유 오브젝트를 선택(picking)하여 lock을 걸고 어떤 조작을 하게 된다.

본 논문에서는 Java 3D를 이용해 오브젝트의 picking을 단순히 마우스 포인터에서의 ray와 오브젝트간에 intersection을 계산해 내는 것 뿐 만 아니라, Java 3D API를 이용하여 computation의 부담을 줄이고 효과적인 picking이 이루어지도록 한다. 논문의 구조는 2장에서 웹기반 협동설계시스템의 구조를 소개하고, 3장에서 일반적인 intersection을 통한 오브젝트의 picking을 설명하며, 4장에서는 Java3D를 이용한 효과적인 picking방법을 제시하고, 5장에서 요약 및 결론을 말한다.

2. 협동 설계시스템의 구조

시스템은 클라이언트/서버 구조로 클라이언트는 자바 애플릿을 통해 웹 상에서 접근하고 서버는 자바 애플리케이션으로 접속을 통제하는 접속관리자, 작업 그룹의 동기화를 유지하며 공유된 작업 공간을 확보하는 작업관리자, 그리고 3차원 도형을 그릴 수

있는 솔리드 모델러로 이루어져 있다. <그림1>은 협동설계시스템 전체 구조를 보여주고 있다.

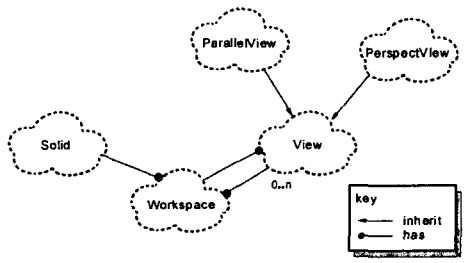


<그림1> 전체 시스템의 구조

접속관리자는 웹서버를 통하여 들어온 클라이언트의 서비스 요청을 받아 분석하여 메시지로 세션관리자에게 보낸다. 사용자의 ID에 따라 접속을 설정하거나 해제할 수 있고, 작업하는 동안 클라이언트의 접속을 계속 유지시키는 역할을 한다.

작업관리자는 클라이언트의 작업 요청에 따라 실제적인 작업을 처리한다. 개인작업공간과 공유작업공간을 확보하도록 하고 도형 객체를 생성하거나 저장된 파일을 불러와 변형시킨다.

솔리드 모델러는 Java 3D를 이용하여 개발하며, 시스템 라이브러리는 여러 가지 Java 클래스들로 구성되며 시스템 레벨의 클래스는 <그림2>와 같다.



<그림2> 시스템라이브러리

솔리드의 기본입체로는 육면체, 원기둥, 원뿔, 토러스, 피라미드, 구 등이다. 이들 기본입체에 대하여 INTERSECT, DIFFERENCE, UNION과 같은 부울리안 연산을 수행하여 솔리드 모델을 만들 수 있게 된다. 솔리드 객체에 대하여 TRANSLATION, SCALING, REFLECTION, ROTATION과 같은 변환을 수행할 수 있다. 솔리드 모델러에서 오브젝트를 만들거나 불러와서 변형을 시키는데, 여러개의 오브젝트 중 어느 것을 선택하여 조작하려고 할 때 picking이 이루어져야 한다.

3. intersection을 통한 오브젝트의 picking

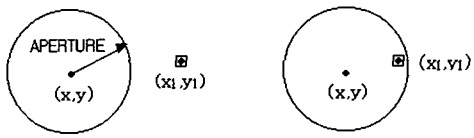
picking이란 화면 내부의 물체 혹은 물체의 일부분을 구성하는 메쉬 등을 선택하는 것이다[4]. 이는 포인팅 디바이스 등으로 어떤 오브젝트를 선택하여 인터액션이 이루어지도록 하기 위한 것으로 마우스 포인터 위치에서 내부 가상 세계에 광선을 쏘아 오브젝트와의 겹치는 부분(intersection)을 찾아내는 것이다.

마우스 포인터(x,y)와 오브젝트(x_1,y_1)와의 거리(D)를 구해 일정한 범위(APERTURE) 내에 들어있는지를 확인한다. <그림3>의 왼쪽은 점(x_1,y_1)이 i)의 경우로 picking이 안되고, 오른쪽 그림은 ii)의 경우로 점(x_1,y_1)이 picking이 된다.

$$D^2=(x-x_1)^2 + (y-y_1)^2$$

i) $(x-x_1)^2 + (y-y_1)^2 > APERTURE^2$

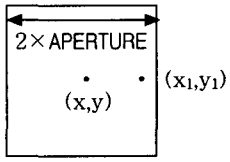
ii) $(x-x_1)^2 + (y-y_1)^2 < APERTURE^2$



<그림3> 오브젝트가 점인 경우의 picking

앞에서 살펴본 D < APERTURE 를 확인하는 것은 셈이 많고 제곱근을 계산해야 하므로 연산시간이 많이 걸린다. 따라서 <그림4>처럼 점(x,y)를 중심으로 한변의 길이가 2×APERTURE인 정사각형의 내부에 점(x_1,y_1)이 들어오는 가를 확인한다.

$$|x-x_1| + |y-y_1| < APERTURE$$



<그림4> 정방형에 의한 테스트

다음은 한 점과 선분과의 거리를 구해 일정한 영역 안에 있으면 picking한다. 주어진 한 점에서 선분에 수선을 그어 그 교점과 주어진 점과의 거리를 구한다. 한 점(x_1,y_1)과 선분 ax + by = 0 와의 거리 D는 다음과 같다.

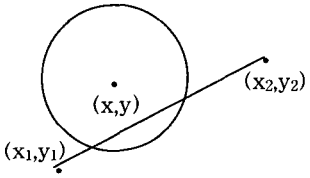
$$D=\frac{|ax_1+by_1+d|}{\sqrt{a^2+b^2}}$$

또 주어진 한 점(x,y)과 두 점 (x_1,y_1), (x_2,y_2)를 양 끝점으로 갖는 선분과의 거리 D는 다음과 같다.

$$D=\frac{|(x-x_1)(y_2-y_1)-(y-y_1)(x_2-x_1)|}{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}}$$

따라서 picking이 되기 위해서는 <그림5>처럼 선분이 원 안으로 들어와야 하므로 다음과 같은 조건을 만족하여야 한다.

$$\frac{((x-x_1)(y_2-y_1)-(y-y_1)(x_2-x_1))^2}{(x_2-x_1)^2+(y_2-y_1)^2}<APERTURE^2$$

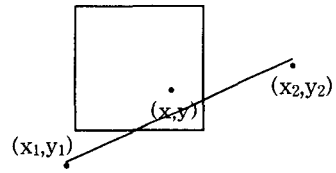


<그림5> picking된 선분

많은 곱셈 연산을 피하기 위해 <그림6>은 원 대신에 정사각형을 사용하였다.

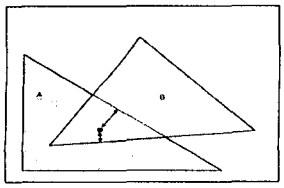
$$\min(|m(x-x_1)+y_1-y|,|\frac{(y-y_1)}{m}+x_1-x|)<APERTURE$$

$$(\text{단, } m=\frac{y_2-y_1}{x_2-x_1})$$



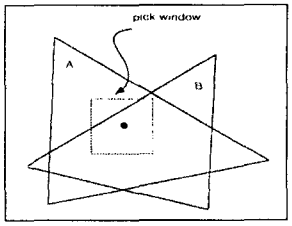
<그림6> 정사각형을 통한 picking

어떤 오브젝트의 내부에 마우스를 클릭하면 그 오브젝트가 선택되었다고 하는데, 하나의 레이어에서 두 오브젝트의의 중첩된 부분을 클릭하였을 때는 어떤 오브젝트를 picking하게 되는지 문제가 된다. 이 경우에는 클릭한 좌표에서 가장 가까운 물체가 picking되도록 한다. <그림7>에서는 점과 가까운 거리에 빗변이 놓인 B가 선택된다.



<그림7> 중첩된 부분의 picking
(출처:[4] p.65)

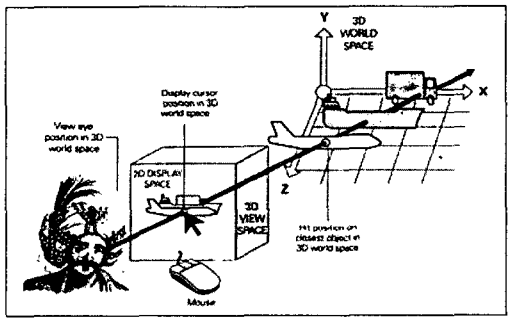
<그림8>은 pick window를 사용하여 picking하고자 하는 부분을 쉽게 볼 수 있도록 한다.



<그림8>pick window
(출처: [4] p.65)

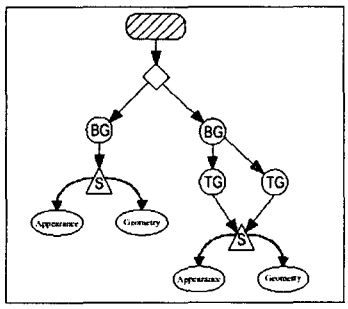
4. 3D 오브젝트의 효과적인 picking

3D 공간에서의 picking은 Z축으로 가장 앞에 놓인 오브젝트를 picking하도록 한다. <그림9>는 마우스에서 ray를 쏘아 첫 번째 비행기의 오브젝트를 picking하는 것을 보여 준다.



<그림9>3D World의 object picking (출처:[6]p.249)

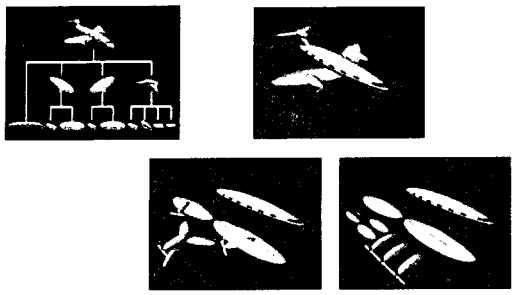
앞에서 언급한 방법들은 컴퓨터 내부적으로 많은 산술연산을 요구하고 있다. Java 3D에서는 <그림10>처럼 scene graph를 사용하므로 어떤 노드를 picking 할 수 있다.



<그림10> Java 3D의 scene graph의 예

노드를 picking하는 경우에는 많은 계산을 필요로 하지 않으며 <그림11>처럼 하위 오브젝트도 따라서

picking되는 효과를 가져오게 된다.



<그림11> 하위 오브젝트의 picking (출처:[8]p26)

Java 3D를 이용하여 3차원 가상 공간에서 효과적으로 오브젝트를 picking하기 위해 다음 과 같은 방법을 사용한다.

첫째 방법은 scene graph node의 attribute와 capability를 이용하는 것으로, 노드에 setPickable() 메소드를 사용하여 true와 false값에 따라 선택한 노드의 하위구조까지도 intersection연산을 하지 않도록 한다.

setPickable(boolean pickable)

또한 특정 Group Node에만 적용하는 capability설정이 있다.

ENABLE_PICK_REPORTING
ALLOW_BOUNDS_READ | WRITE
ALLOW_PICKABLE_READ | WRITE

둘째는 bounds를 설정하는 것으로 복잡한 도형에 대해 일정한 영역을 설정하여 연산을 빠르고 간단하게 하도록 한다.

USE_BOUNDS
USE_GEOMETRY

셋째는 pick testing의 scope를 한정하여 intersection을 찾기 위해 모든 부분에 대해 연산하는 것을 줄일 수 있다.

다음은 본 협동설계시스템의 솔리드 모델러에서 사용한 Java 3D의 picking method들이다.

Node Method
extends: SceneGraphObject
subclasses: Group, Leaf
void setBounds(Bounds bounds)
void setBoundsAutoCompute(boolean autoCompute)
setPickable(boolean pickable)

Node Capabilities
ENABLE_PICK_REPORTING
ALLOW_BOUNDS_READ | WRITE
ALLOW_PICKABLE_READ | WRITE

PickBounds
PickBounds()
PickBounds(Bounds boundsObject)
Bounds get()
void set(Bounds boundsObject)

5. 요약 및 결론

협동설계시스템은 가상공간에서 3D 도형을 공유 오브젝트로 사용하게 된다. 따라서 다수의 사용자가 임의의 공유 오브젝트를 선택하여 조작하기 위해 효과적으로 picking하는 것이 중요한 문제이다.

본 논문에서는 특정 platform과 소프트웨어에 구애받지 않고 인터넷 웹 상에서 이루어지도록 Java를 사용하였다. Java 3D API를 이용해 오브젝트의 picking을 scene graph의 노드에 picking 속성을 주는 방법, bounds를 설정하는 방법, picking test의 범위를 한정하는 방법을 사용하여 computation의 부담을 줄이고 효과적인 picking이 되도록 하였다.

감사의 글

본 연구는 과학기술부, 한국과학재단 지정 여수대학교 "설비자동화 및 정보시스템 연구개발센터"의 지원에 의한 것입니다.

참고문헌

[1] F. Faure, C.Faisstnauer, G.Hesina, "Collaborative animation over the net", IEEE 1999, p107-116

[2] Icgiro Hagiwara and Shinsuke Noda, "Homotopical Modeling as the Basis of New CAD Standard Homotopy CAD for Collaboration Engineering", IEEE 1999, p231-237

[3] 윤보열, 김응곤, "웹 기반 협동CAD시스템에 관한 연구", 한국해양정보통신학회 논문지 Vol.4. No.4, 2000

[4] 주우석, "3차원 컴퓨터그래픽스", 도서출판 그린 , 1999

[5] 심재홍, "컴퓨터그래픽스",이한출판사, 1996

[6] Jon Barrilleaux, "3D User Interfaces with Java 3D" , manning, 2001 p.247-256

[7] Denis J Bouvier, "Getting started with the Java 3D API", Sun microsystems, 2000, http://developer.java.sun.com/developer/onlineTraining/java3d/

[8] Henry A. Sowizral, David R. Nadeau, "An Introduction to Programming AR and VR Applications in Java 3D", http://www.sdsc.edu/~nadeau/Courses/VR99/