테이블 탑 디스플레이 인터페이스에서 물체의 상호작용 표현을 위한 물리 엔진 구현

정종문, 김만선, 오진식, 김정식, 양형정
전남대학교 전자컴퓨터공학부
e-mail :amiris@nate.com, kms9688004@nate.com, js21cfriend@hanmail.net
wind1105@lycos.co.kr, hjyang@chonnam.ac.kr

Implementation of Physis Engine for Interactions Representation of Object at Table Top Display Interface

Jong-Mun Jeong, Man-Sun Kim, Jin-Sik Oh
Jeong-Sik Kim, Hyung-Jeong Yang
Dept of Computer Science, Chonnam National University

요 약

테이블 탑 디스플레이는 인간과 컴퓨터간의 자연스러운 상호작용을 위하여 개발된 인터페이스 중 하나이다. 이것은 인간의 직관적인 도구인 손을 이용하여 컴퓨터와 상호작용을 하기 때문에 기존의 마우스를 이용하는 시스템에 비해 사용자의 흥미를 더욱 유발시킬 수 있으며 따라서 현재 이러한 시스템을 활용하기 위한 많은 컨텐츠들이 개발되고 있다. 본 논문에서는 테이블 탑 디스플레이 인터페이스에서 자연스러운 물체의 상호작용을 지원하는 물리엔진을 구현하였다. 이를 위해 테이블 탑 평면상에서 물체의 선택을 가능하게 하는 2차원 관점의 3차원 변환을 지원하고, 물체가 이동할 때와 물체와 물체의 충돌 시 나타나는 물리현상을 벡터연산을 통해 구현하였고, 네트워크를 통해 다중 사용자 환경에서 물리엔진이 구동되도록 하였다. 본 논문에서는 테이블 탑에서 이와 같은 기능들의 구현을 에어하키 게임을 통해 보인다. 에어 하키는 테이블 위에 퍽을 놓고 라켓으로 퍽을 쳐서 상대방의 골문에 넣어 점수를 얻는 게임이다. 본 논문에서 제안한 물리엔진을 이용함으로써 사용자는 보다 실감나는 인터페이스를 느낄 수 있다.

키워드: 테이블 탑 디스플레이, 컨텐츠, 물리엔진

1. 서론

최근에 인간과 컴퓨터간의 자연스러운 상호작용을 위하여 시각을 기반으로 한 사용자 의도 및 행위를 인식하기 위한 연구가 활발히 진행되어 왔다. 그 중에서도 손을 이용한 제스처 인식은 시각 기반 인식 분야에서 핵심 기술 분야로 계속 연구되어 왔으며, 이를 이용한 인간과 컴퓨터의 상호작용에 관한 연구가 활발하게 진행되고 있다.

테이블은 인간이 커뮤니케이션을 하는 장소로 많이 이용 되고 있다. 이러한 테이블 위에서 컴퓨터와 상호작용을 할 수 있다면 좀 더 자연스럽게 컴퓨터를 이용할 수 있을 것이다. 테이블 탑 디스플레이는 인간의 직관적인 도구인 손을 이용하여 쉽게 컴퓨터와 상호작용을 할 수 있는 적합한 시스템이다. 테이블 탑 디스플레이는 출력과 입력을 함께 처리하므로 인간은 컴퓨터와의 상호작용에서 더욱 자연스러움을 느끼게 된다. 테이블 탑 디스플레이의 이러한 장점을 활용하기 위해서 많은 컨텐츠들이 개발되고 있다[6][8].

본 논문에서는 테이블 탑 디스플레이 인터페이스에서 자연스러운 물체의 상호작용을 지원하는 물리엔진을 구현하였다. 물리 엔진은 컴퓨터·비디오 게임이나 실시간 그래픽 표시 기능을 갖춘 상호작용 어플리케이션을 구현하는 핵심 소프트웨어 구성 요소를 말한다[7]. 테이블 평면상에서 물체의 선택을 위해 2차원 관점을 3차원으로 변환 하였으며, 벡터연산을 통해 물체의 이동과 충돌 시 나타나는 물리현상을 구현하였고, 이를 여러 대의 테이블 탑에서 이용할 수 있도록 네트워크를 설정하였다.

본 논문에서는 테이블 탑에서 이와 같은 기능들의 구현을 에어하키 게임을 통해 보인다. 에어 하키는 테이블 위에 퍽을 놓고 라켓으로 퍽을 쳐서 상대방의 골문에 넣어 점수를 얻는 게임이다. 본 논문에서 제안한 물리엔진을 이용함으로써 사용자는 보다 실감나는 인터페이스를 느낄 수 있다.

2. 관련 연구

협동적 상호작용이란 공동 작업을 하는 구성원들이 직간접적인 의사소통을 통하여 어떤 현상을 일으키거나 시스템을 조작하는 것을 말한다. 크게 두 가지로 나눌 수 있다[1].

1)정전 용량 방식을 이용한 상호 작용

정전 용량 방식을 이용한 상호 작용 스크린 밑에 Force-Sensitive-Resistors의 정전 용량의 변화량을 측정할 수 있는 장치를 장착하여 손의 위치나 접촉정보를 감지할 수 있도록 한다. 정전 용량 기반 방식은 시스템 복잡도를 크게 낮추는 효과가 있으며, 스크린에 손의 접촉여부와 더불어 얼마나 접근했는지의 접근성 여부를 판별할 수

본 연구는 문화관광부 및 한국문화콘텐츠진흥원이 주관하는 전남대학교 문화콘텐츠기술연구소 육성사업의 연구비 지원에 의해 수행되었음

있다는 장점이 있다. 이러한 장점을 활용하여 최종적으로 상호작용이 가능한 표면을 현실세계로 만들 수 있다. 그러나 이 시스템들은 스크린 속에 안테나 및 센서 등의 전자적 장치가 내장되어 있으므로 거의 대부분이 상부투영(top-down projection) 방식을 채택하고 있다. 따라서 손과 투영된 이미지가 겹치는 현상이 자주 발생하여 사용자는 겹치는 부분과 스크린 위에서 손이 움직이는 것을 자주 응시하게 되어 주의 집중을 방해하는 결과를 초래한다. 또한 높은 해상도를 구현하기 위해서는 아주 많은 센서 연결을 필요로 하므로 제작비용이 매우 증가한다.

2)시각 기반 방법을 이용한 상호 작용

시각 기반 감지 방법은 비디오 카메라를 이용하여 스크린 위에 놓여진 물체를 검출 및 인식하여 입력신호를 만드는 방법이다. 그러나 시각 기반 방법은 카메라만을 이용하여 인식해야 하기 때문에 정확도나 인식률이 떨어진다.

본 연구에서는 이러한 문제점을 해결하기 위하여 테이블 탑 디스플레이의 자연스러운 인터페이스와 물리엔진을 사용한 가상현실 게임을 융합한 대결 게임 시스템을 구현하였다.

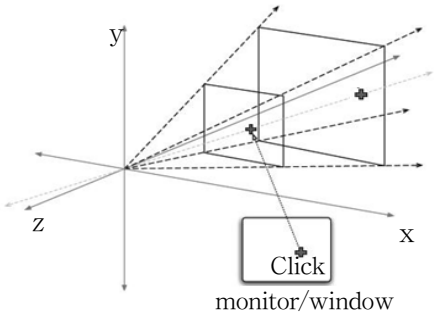
3. 물체 상호작용 표현을 위한 물리엔진

테이블 탑 디스플레이에서 물체를 마우스 대신 손으로 선택하기 위하여 좌표 값을 3D로 변환하는 과정이 필요하다. 또한 터치 포인터의 움직임에 따라 물체에 가해지는 힘의 크기가 달라지고 이에 따라 속도를 결정하는 부분이 필요하다.

3.1. 3D 객체 선택

3차원 공간의 물체를 선택하는 것은 화면에 보이는 좌표 값만으로는 쉽지 않다. 이 문제를 해결하기 위하여 2D인 좌표 값을 3D로 변환하는 과정이 필요하다.

(그림 1)의 원점은 카메라가 있는 위치이고 -z축 방향으로 바라보고 있다고 가정하자. 첫 번째 사각형은 카메라에 의해 찍힐 수 있는 가장 가까운 거리의 화면이고 두 번째 사각형은 카메라에 의해 찍힐 수 있는 가장 먼 거리의 화면이다. 첫 번째와 두 번째 사각형 사이의 공간이 모니터를 통하여 보여 지게 된다. 모니터 평면상에 보이는 물체를 선택하기 위해 손으로 터치하면 그 좌표와 대응되는 카메라와 가장 가까운 사각형의 한 점과 카메라와 가장 먼 사각형의 한 점을 구한다. 두 점을 연결하는 직선과 만나는 물체가 바로 손으로 터치한 물체가 된다.



(그림 1) 원근투상의 가시적인 부피

```
GLint gluUnProject( GLdouble winX,
                    GLdouble winY,
                    GLdouble winZ,
                    const GLdouble *model,
                    const GLdouble *proj,
                    const GLint *view,
                    GLdouble* objX,
                    GLdouble* objY,
                    GLdouble* objZ )
winX, winY - 윈도우 좌표
winz - 0.0일 경우 첫 번째 스크린의 좌표, 1.0이면 두 번째 스크린의 좌표
model - 모델뷰 매트릭스 (glGetDoublev 함수를 이용)
proj - 프로젝션 매트릭스 (glGetDoublev 함수를 이용)
view - 뷰포트 매트릭스 (glGetIntegerv 함수를 이용.)
objX, objY, objZ - 구한 좌표 값
```

(그림 2) 변환된 두 점을 찾는 함수

화면의 마우스 좌표를 이용하여 실제 3D공간의 좌표를 구하기 위해서는 openGL에서 제공하는 (그림 2)의 gluUnProject라는 함수를 사용한다. gluUnProject함수는 (그림 1)의 두 개의 사각형에 보이는 점을 찾는데 사용하는 함수이다.

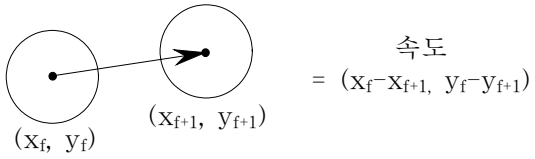
gluUnProject 함수로 두 점을 구했으면 두 점을 지나는 직선의 방정식을 구해서 테이블과 만나는 점을 구한다. 두 점의 좌표를 각각 (x_1, y_1, z_1), (x_2, y_2, z_2)라고 하면 공간에서 두 점이 만드는 직선의 방정식은 다음의 수식(1)과 같다.

$(obj_x, obj_y, obj_z) = (x_1, y_1, z_1) + t(x_2, y_2, z_2)$ - 수식(1)

이 직선이 테이블과 만나는 점을 구해야 하는데 테이블 면은 y=0인 평면에 있다 그래서 $obj_y=0$으로 하여 매개변수 t를 구하고 t를 이용하여 obj_x, obj_z값이 결정된다.

3.2. 물체의 이동과 속도 결정

물체에 가해지는 힘은 터치 포인터의 움직이는 속도에 따라 달라진다. 즉, 손을 빠르게 움직일 경우 힘의 크기가 커서 물체에 부딪칠 경우 물체는 빠르게 움직이고 느리게 움직이면 힘의 크기가 작아서 물체는 느리게 움직일 것이다. 이러한 힘의 크기를 결정하기 위해 이전 프레임에 그려진 위치와 현재프레임에 그려진 위치를 이용하여 (그림 3)과 같이 위치변화량인 속도를 결정한다.



(그림 3) 손 이동시 속도

3.3. 테이블 탑 디스플레이에서 물리 충돌 표현

물체 충돌 표현에서 보여 지는 물리현상은 크게 두 가

지가 있는데 이동과 충돌이다.

이동은 테이블위에서 퍽이 움직이는 현상이다. 퍽은 속도벡터의 방향과 크기에 의해서 이동하게 되는데 이때 테이블 위를 이동하면서 수식(2)와 같이 테이블과 퍽의 마찰력에 의해 속도가 줄어들게 된다.

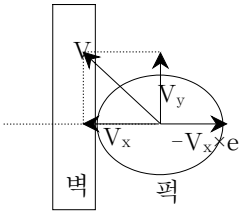
$V_2 = V_1 \times \mu$ (V_1: 현재속도, V_2: 나중속도, μ: 마찰계수)

수식(2)

충돌은 먼저 물체와 물체의 위치를 확인하여 충돌여부를 파악해야한다. 충돌하였을 경우 각 물체의 상태에 따라 속도 벡터를 변화시킨다. 충돌은 크게 물체와 테이블 벽 사이, 물체와 물체 사이의 충돌이 있다.

1) 물체와 벽의 충돌

물체와 테이블 벽의 충돌 시 테이블은 정지 상태이고 물체는 이동 방향을 바꾸게 된다. 이 경우에는 물체의 충돌 후 속도벡터만 구하면 된다. 충돌 후 속도벡터를 구하기 위해서 테이블 벽의 충돌 위치와 물체의 위치를 연결하는 축을 구한다. (그림 4)와 같이 이 축을 기준으로 하여 물체의 속도를 분할한다.

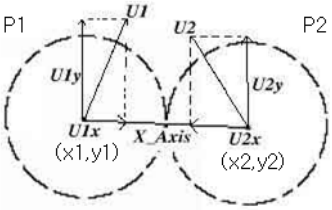


(그림 4) 물체와 테이블 충돌

분할된 속도 중에 축과 평행한 속도벡터가 테이블과의 충돌로 인해 방향이 반대방향이 된다. 충돌 후 반대 방향이 된 속도 벡터는 충돌 계수에 의하여 속도가 줄어들게 된다. 분할 후 계산한 속도를 다시 더하면 물체의 충돌 후 속도가 된다.

2) 물체와 물체의 충돌

물체와 물체의 충돌 시 서로에게 힘을 가하게 되어 물체의 이동 방향이 바뀌게 된다. 물체끼리의 충돌여부 판단은 (그림 5)와 같이 물체의 중심위치간의 거리를 계산하여 두 물체의 반지름의 합보다 작을 경우 두 물체가 충돌했다고 판단한다. 충돌했으면 충돌후의 속도벡터를 계산하여 물체에 대입하면 물체는 충돌후의 위치로 이동하게 될 것이다.



(그림 5) 물체와 물체가 부딪친 경우

충돌 후의 속도벡터를 계산하기 위해서는 먼저 물체의 중심을 연결하는 축의 단위 벡터를 수식(3)과 같이 구한다.

$$X_{Axis} = (x_2 - x_1,\ y_2 - y_1) / (\text{두 점 사이의 거리})$$

수식(3)

충돌하기 전의 속도벡터를 X_{Axis}를 기준으로 분할을 한다. 물체의 중심을 이은 선을 x축이라고 하면 물체의 속도벡터 U1이 x축의 방향으로 어느 정도의 크기를 가지고 있는지 알기위해 내적을 수식(4)와 같이 구한다[4].

$$U1_x = X_{Axis} \times (X_{Axis} \cdot U1)$$

$$U1_y = U1 - U1_x$$

$$U2_x = -X_{Axis} \times (-X_{Axis} \cdot U2)$$

$$U2_y = U2 - U2_x$$

수식 (4)

크기를 구했으므로 x축 단위 벡터를 곱하면 $U1_x$가 구해지게 된다. $U1_y$ 는 U1과 $U1_x$의 차이를 계산하여 구할 수 있다.

분할한 속도 벡터와 물체의 질량을 이용하여 충돌후의 속도 벡터를 수식(5) 와 같이 계산한다. 분할된 속도벡터 $U1_x$와 $U2_x$가 서로 충돌하였기 때문에 에너지 보존의 법칙과 운동량의 법칙에 의해서 수식(5) 와 같이 계산되어지고 $U1_y$와 $U2_y$는 외부로부터의 힘의 변화를 받지 않았기 때문에 충돌 후에도 속도는 같다. 결론적으로 물체의 충돌 후의 속도 벡터는 $V1_x$+$V1_y$가 된다.

$$V1_x = \frac{((M1 - e \times M2) \times U1_x) + ((1+e) \times M2 \times U2_x)}{M1 + M2}$$

$$V1_y = U1_y$$

$$V2_x = \frac{((M2 - e \times M1) \times U2_x) + ((1+e) \times M1 \times U1_x)}{M1 + M2}$$

$$V2_y = U2_y$$

(e : 충돌계수, $M1, M2$: 질량)

$$V1 = V1_x + V1_y$$

$$V2 = V2_x + V2_y$$

수식 (5)

3.4. 네트워크를 통한 정보 교환

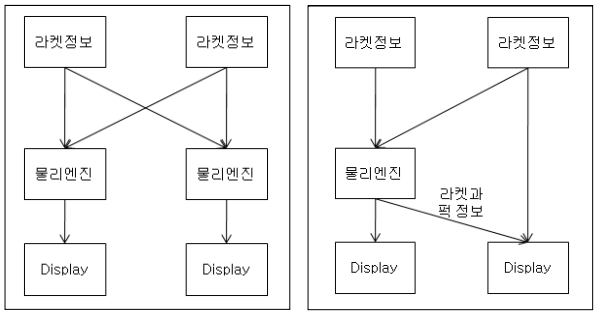
네트워크 환경을 이용하는 것은 공간적인 제약에서 좀 더 자유로워 질 수 있으며, 여러 명이 함께 게임을 할 수 있다는 장점이 있다. 네트워크를 통해 상호작용을 하기 위해서는 각 단말기에서 데이터의 주고, 받음을 (그림 7)과 같이 처리해야 한다. 테이블 탑 디스플레이에서 테이터를 교환하기 위해서는 두 가지의 데이터 전송, 처리방식을 고려할 수 있다.

첫 번째, (그림 6)의 왼쪽 그림처럼 각 단말기는 터치 정보만을 주고받아 자신의 물리엔진에 적용시켜 상호작용을 진행시킨다. 이 방법을 적용시킬 경우 전송하는 데이터의 양이 적은 장점은 있지만 네트워크의 전송 속도에 의해 두 단말기 사이에 시간차가 생기게 되고 같은 시간에 터치 정보가 달라서 물리엔진에 의한 물체위치 정보의 계산 결과는 서로 달라진다.

두 번째, (그림 6)의 오른쪽 그림처럼 한쪽 단말기에서는 터치 정보만을 전송하고 반대쪽 단말기에서 계산한 물체위치 정보와 터치정보만을 받아와서 상호작용을 하는

방식이다. 이 방법은 전송하는 데이터의 양은 첫 번째 방법보다는 많지만 쉽게 두 단말기의 정보를 같게 할 수 있다.

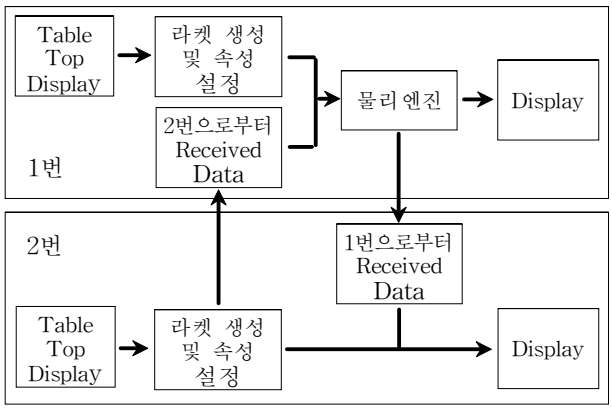
본 논문에서는 서로 같은 화면을 보일 수 있는 두 번째 방식을 사용하여 시스템을 구현하였다.



(그림 6) 정보교환 방식

4. 상호작용 물리엔진을 이용한 응용 구현

본 논문에서 구현된 시스템은 두 대의 테이블 탑을 사용하여(그림 7)과 같은 구성으로 이루어져 있다. 1번 시스템은 물리엔진이 구동하고, 2번 시스템은 물리엔진이 구동하지 않는다. 두 시스템이 각자 물리엔진을 구동 시킬 경우 두 시스템의 데이터가 서로 달라지는 현상을 막기 위해서 한쪽의 시스템만 물리엔진을 구동시켰다. 테이블 탑 디스플레이를 통해 사용자로부터 입력을 받으면 각각의 시스템은 입력 정보를 이용하여 라켓을 생성 및 설정한다. 2번 시스템은 물리엔진이 구동하지 않기 때문에 라켓 데이터를 1번 시스템에게 전송한다. 1번 시스템은 사용자의 입력의해 만들어진 라켓과, 네트워크를 통해 전송받은 라켓, 그리고 펙의 데이터를 이용해 물리엔진을 구동한다. 물리엔진을 통한 계산 결과는 2번 시스템에게 전송되고, 화면에 표현된다. 2번 시스템은 1번 시스템에게 받은 데이터를 화면에 표현한다.

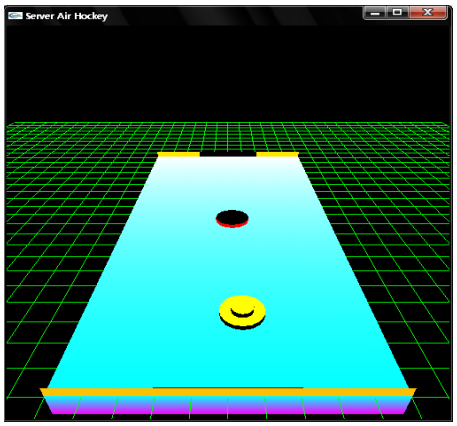


(그림 7) 에어하키 시스템 구성도

(그림 8)은 구현된 에어하키게임의 화면이다.

5. 결론

본 논문에서는 테이블 탑 평면상에서 물체의 선택을 위해 2차원 관점을 3차원으로 변환하였고, 물체의 이동과 물체와 물체의 충돌 시 나타나는 물리현상을 벡터연산을 통해 구현하였고, 이를 여러 대의 테이블 탑에서 이용할 수 있도록 네트워크를 설정하였다. 본 논문에서는 테이블 탑에서 이와 같은 기능들의 구현을 에어하키 게임을 통해 보였다. 이 상호작용 물리엔진의 기대 효과는 실제와 같은 재미를 느낄 수 있으므로 흥미를 증가시킬 수 있었고, 테이블 탑 디스플레이의 인터페이스를 자연스럽게 받아들일 수 있었다는 것이다. 그러나 테이블 탑 디스플레이에서 물체를 움직일 때 손가락을 화면에 터치하여 움직이는데, 화면과 손가락 사이의 마찰 때문에 부드럽게 움직이지 못하는 단점이 있었다.



(그림 8) 구현된 에어하키 화면

향후 연구 과제로는 이러한 점을 해결하기 위해 다른 도구를 이용한 인식 방법을 연구하고, 멀티 터치를 활용한 시스템 개발이 요구될 것이다.

참고문헌

[1] 김송국, 이칠우, “멀티터치를 위한 테이블-탑 디스플레이 기술 동향,” 한국 콘텐츠 학회 논문지 제7권 제2호, pp. 84-91, 2007.

[2] Richard S. Wright, “OpenGL SuperBible,” 제3판, 정보문화사, 2005.

[3] Dave Astle, Kevin Hawkins, Todor J. Fay, “Beginning OpenGL Game Programming,” Course Technology Ptr 2004.

[4] http://nehe.gamedev.net/, NeHe Production

[5] 김선우, “윈도우 네트워크 프로그래밍 : TCP/IP 소켓 프로그래밍,” 한빛미디어, 2004.

[6] 김송국, 이칠우, “협력적인 상호작용을 위한 테이블 탑 디스플레이 기술 동향,” 한국콘텐츠학회 종합학술대회 논문집 한국콘텐츠학회 2006 추계 종합학술대회 논문집 제4권 제2호, 2006. 11, pp. 616-621.

[7] 이기석, 이동춘, 김항기, 박상욱, 박창준, “게임 물리기술 동향,” 전자통신동향분석 22권 4호 2007,8월.

[8] 장한별, 김송국, 김장운, 이칠우, “테이블 탑 디스플레이 기반 실감형 전자 요리 시스템”, 한국콘텐츠학회 종합학술대회 논문집 한국콘텐츠학회 2007 춘계 종합학술대회 논문집 제5권 제1호, 2007. 6, pp. 34-38.