

가상 환경에서의 대화형 체감 시스템

배희정*, 손욱호, 장병태

*한국전자통신연구원 VR센터 증강현실연구팀
hjbae@dreamwiz.com

Interactive Haptic System in Virtual Environment

Heejung Bae*, Wookhom son, Byungtae Jang
*VR Center, Electronics & Telecommunications Research Institute

요약

가상 현실이나 게임 제작 분야에서는 체감에 의한 실제감을 실시간으로 처리하는 것이 요구되고 있다. 체감 효과를 표현하기 위한 일반적인 형태는 PhanTom에서 사용 가능한 햅틱 라이브러리를 사용하지만, 처리 속도의 문제 때문에 대화형 시스템(interactive system)에는 그 사용의 한계를 가진다. 본 논문에서는 햅틱 렌더링의 근본적인 문제인 실시간 렌더링 문제를 극복하기 위한 비선형 보간에 의한 근사 방법을 제안한다. 제안하는 방법은 하이브리드 접근 방식으로, 캐릭터에 작용하는 동역학적 움직임을 햅틱 효과 생성을 위하여 기하학적으로 재구성함으로써 체감효과 생성에 필요한 시간을 최소화 할 수 있다. 본 논문의 결과로는 캐릭터의 빈번한 제어에서 발생하는 체감 효과를 빠르게 생성할 수 있음을 보이며, 안정적이면서 그럴 듯한 효과를 생성할 수 있음을 보인다.

1. 서론

컴퓨터 그래픽스와 가상현실 기술의 발달에 따라 컴퓨터 화면에서 실세계의 물체들을 실제와 흡사하게 표현할 수 있게 되었다. 최근에는 기존의 시각적 특수 효과를 벗어나 사용자에게 청각 효과나 체감 효과를 통한 실제감을 부여하는 방법에 대한 관심이 고조되고 있다[1].

화면에 표현되는 캐릭터의 경우에는 3D 모델링이나 스캐닝 기구들을 사용하여 간단히 컴퓨터 화면상에 재현이 가능하다. 또, 이들의 동작은 가상 환경에서 발생하는 사용자와의 상호 작용에도 실제와 흡사하게 실시간으로 재현할 수 있다[2]. 반면에, 부딪히거나 움직이는 등에서 부여될 수 있는 체감 경우에는 이들과는 다르게 새로운 몰입감을 부여할 수 있다. 그림 1에 제시된 가상현실에서의 격투 장면에서와 같이, 사용자는 캐릭터로부터 하나의 완전한 가상 세계를 느낄 수 있다. 즉, 여러 가상 환경에서

발생되는 캐릭터의 움직임은 사용자의 제어에 대응되도록 표현되고, 폭발물과 같은 가상 환경의 힘에 의해 캐릭터가 충돌되는 상황을 실제와 흡사하게 체감하게 될 수 있다. 캐릭터를 제어하는 입장에서 실시간으로 이루어지는 캐릭터와의 교감이 가장 중요한 요소가 된다.



그림 1. 격투의 한 장면
(폭파에 의한 체감 효과)

본 논문에서는 체감을 포함하는 가상 시스템을 제안한다. 가상 세계에서의 캐릭터 체감을 컴퓨터 상에서 재현하기 위해서는, 우선 체감의 효과를 생성하여야 한다. 실시간으로 체감을 사용자에게 전달하기 위해서는 캐릭터의 움직임을 실시간으로 제어하고, 캐릭터 내의 각 부분에 대한 충돌과 충격량 등의 물리적 성질에 대응되도록 생성한다. 캐릭터의 체감은 기하학적 성질과 물리적 성질들을 이용하여 동역학적 방법에 기반하여 생성한다. 즉, 햅틱 기구를 통하여 가상의 힘을 생성한 후, 이 제어에 의하여 캐릭터가 움직이는 힘을 계산하고, 이때 발생하는 체감의 경우에는 캐릭터의 동역학적 움직임에 의하여 그 효과를 생성한다. 이 과정에서 생길 수 있는 물체들 간의 충돌은 impulse dynamics 방법으로 해결한다.

본 논문은 일반적인 PC 환경에서도 대화형으로 캐릭터의 체감을 재현하는 데에 목적을 둔다. 이를 위해서 동역학에 기초하여 단순화된 충돌 모델을 사용한다. 또, 운동학 방법들에서 필연적으로 풀어야 하는 대형 연립방정식을 캐릭터의 위상 정보를 이용하여 선형 시간에 풀 수 있는 방법을 제시한다. 본 논문이 제시한 방법들은 단순히 격투에만 한정된 것이 아니라 대화형의 실시간 게임 환경에서 응용할 수 있다.

본 논문의 전체 구성은 다음과 같다. 2장에서 가상 체감 시스템의 전체 구성에 대해서 설명하고, 3장에서는 캐릭터의 체감 효과를 모델하는 방법에 대해 기술한다. 4장은 본 논문에서 사용한 가상 체감 모델에 의한 효과 생성 방법과 impulse dynamics solver에 대해서 설명한다. 5장에서는 가상 체감의 실제 효과 재현에 대한 예제들을 보이고, 마지막으로 6장에서는 결론과 향후 과제를 제시한다.

2. 전체 구성

본 논문의 가상 체감 시스템은 펜티엄 급의 PC에서 DirectX 라이브러리를 이용하여 C++ 언어로 구현하였다. 그림 2는 가상 모빌 시스템의 전체 흐름도이다.

체감 인터페이스는 물리적 장치를 이용하여 사람이 직접적인 접촉이 없이도 촉각 정보를 전달하는 장치이다. 사용자에게 모델링 되어 있는 가상의 물리적 특성을 실제와 똑같이 느끼게 하는 것을 목적으로 둔다.

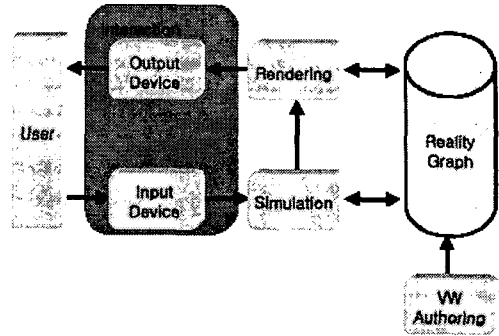


그림 2. 가상 체감 시스템의 개념도

시뮬레이션 모듈은 물리 기반 동역학적 모듈과 이벤트 처리기(event processor) 모듈로 나뉘어 진다. 시뮬레이션 모듈의 구성은 그림 3에서와 같다.

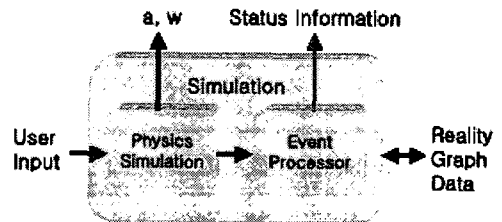


그림 3. 사용자 제어에 따른 시뮬레이션 모듈

물리 기반 동역학 모듈은 가상 환경에 대한 운동학(kinematic), 동역학(dynamics)을 해결하고 캐릭터와 가상 환경에서의 객체들과의 상호 작용을 감지한다. 물리 시뮬레이션의 입력값은 입력 장치를 통한 사용자 입력 정보이고, 출력값은 운동학과 동역학에 의해 풀린 캐릭터의 가속도 값과 각속도 값을 렌더링 및 노이즈 효과와 관련된 캐릭터의 이벤트 정보(event information)을 출력한다. 이벤트 정보는 사용자 입력에 의해 발생하는 사용자 입력 정보(user input information), 가상객체와 캐릭터와의 상호작용에 의해 발생하는 상호작용 정보(interaction information), 그리고 일정한 시간에 의해 발생하는 시간 정보(time information)가 있다.

출력장치의 촉각 부분에서는 3자유도를 가진 운동판(motion platform)에서 렌더링 된 결과가 운동판의 기구학을 거쳐서 최종적으로 다리 길이를 출력하여 운동판을 움직여주게 된다. 형태로 이벡트 신호를 받아 출력하게 된다.

3. 햅틱 렌더링 모델(Haptic Rendering Model)

햅틱 렌더링은 햅틱 인터페이스에서 가상의 물체와 물리적 장치사이에서 사실적인 힘(realistic force)를 느끼게 하는 데 필요한 일련의 계산적인 모든 과정을 가리킨다.

즉, 사용자는 물리적 장치를 움직이겠지만 가상의 환경에서는 물리적 장치에 해당하는 햅틱 handle이 움직이면서 물체에 부딪히게 된다. 이때 물체의 물리적 성질과 함께 어느 방향에서 어떻게 부딪히느냐에 따라서 변하는 햅틱 handle에서의 힘을 결정하여야 한다.

3.1 충돌 처리

충돌처리는 매우 빠르게 처리 되어야 물체와의 충돌이 지연되거나 예상하지 못한 bouncing들을 느끼게 된다. 햅틱 렌더링을 위한 충돌처리는 동역학에 기반하여 단순화된 모델을 사용한다.

충돌 대상 객체에 대해서 각각 계층적 bounding spher, OBbtrees를 씌운다. 일반적으로는 전처리 단계에 공간을 반대로 분할해 나가면서 계층구조를 만들어 나가지만, 장면 그래프가 각 객체를 이루는 노드간에 계층구조를 가지고 있기 때문에 이것을 이용해서 계층

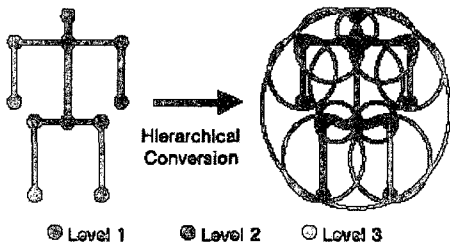


그림 5 캐릭터의 계층구조 생성

구조를 가지는 bounding sphere, OBbtrees를 그림 *에서와 같이 구성하였다.

그림 *에서와 같이 모든 노드들을 leaf로 놓고, left 노드에 있는 것들을 묶어 계층 구조를 만들어 나가는 방법이다. OBB도 같은 방법으로 계층구조를 생성한다. 그리고, 각 노드를 update할 때는 leaf 노드에서부터 root,까지 올라가는 bottom-up방법을 이용한다.

다음으로, 객체들간의 충돌이 일어나는지를 테스트 한다. 빠른 계산을 위해서 여기서는 bounding volume에 대한 교차 검사를 한다. 즉, Obbs-Obbs, Obbs-Sphere, Sphere-Sphere 간의 검사를 통해 충돌

이 일어나면 다음 단계에서 충돌 검사를 하고 그렇지 않으면 충돌이 일어나지 않았음을 알리고 검사를 종료한다.

충돌이 발생한 경우, 그 반응을 일으켜야 할 방향에 대한 결과값을 출력하고, 그렇지 않을 경우 앞 과정으로 돌아가 다른 객체들 간의 충돌 검사를 한다.

그림 *은 간단한 충돌 반응처리 과정을 보여준다. 즉, 부딪힌 두 bounding volume에 대해서 중심점 좌표를 구하고, 그 둘의 벡터차를 구해 각 volume에 대해 방향 벡터를 주면, 제시된 bounding volume이 이용하도록 하였다. 실제로는 벡터값을 참조해 충돌 반응의 방향을 결정하여 처리한다.

3.2 마찰을 고려하지 않은 모델(Friction-less Model)

충돌의 감지이후, 햅틱 렌더링 모듈에서 처리되어야 할 것은 이들에 대한 반작용으로 주어질 힘을 결정하는 것이다. 가상의 물체의 물리적 성질, 스프링, 질량, 댐피 등의 성분으로 모델링이 가능하며, 일반적으로 이들의 성질을 이용하여 처리한다. 물체의 거칠기 등 표면의 성질을 묘사하기 위해서는 마찰력 등의

전체 시스템의 내부 처리는 뉴턴의 동역학 법칙에 의하여 이루어지며, 작용하는 촉각 힘은 아래와 같이 구할 수 있다.

$$f = (p_c - p_t) \cdot k - v \cdot b$$

여기서, 접촉점에서 발생하는 힘 f 는 접촉점 p_c 와 충돌이 발생하는 위치인 p_t 지점을 나타낸다. 그리고, k 는 스프링 경직도를 나타내고, b 는 물체의 상대속도를 나타낸다. 스프링의 경직도는 물체의 물리적 속성에 따른 힘의 크기를 조절할 수 있으며, 상대속도는 힘의 발산을 방지할 수 있다.

이렇게 적용된 접촉 힘은 가상 물체의 딱딱한 정도나, 충격력 등을 느끼게 하는데 사용된다. 예를 들어 매우 딱딱한 물체를 느끼게 하려면 사람이 눌러도 거의 안 들어 갈만큼 큰 힘을 내어 주어야 하므로, 시스템이 불안정해지는 현상을 쉽게 관찰할 수 있다. 이러한 문제를 해결하기 위해서는 에너지를 소비할 수 있는 방법으로 가상의 모델에 임의의 damper를 다는 방법을 적용한다.

$$f = k_{wall} \Delta x + B \cdot \dot{x} \quad \text{for } \dot{x} < 0$$

$$f = k_{wall} \Delta x \quad \text{for } \dot{x} \geq 0$$

물체가 0의 위치에 있을 때 위의 식과 같은 방법으로 모델링하면 딱딱한 물체를 비교적 안정되게 구현할 수 있게 된다[*]. 그러나, 이 경우 damper로 인해 사람이 인위적인 진성을 느낄 수 밖에 없는 단점을 가지게 된다.

3.3 마찰을 고려한 모델

실제의 물체는 고유 특성상 마찰력을 가진다. 이러한 물체의 마찰을 고려하지 않을 경우, 물체의 탄력적(elastic)이기 때문에, 외부의 다양한 제어에 대하여 근본적으로 불안정성을 포함한다. 특히, 적분에 의한 스프링의 근사 방법에서는, 늘어지는 속성으로 인하여 위치의 변화율이 매우 커지므로, 실제와 흡사하게 근사 시키는 것은 더욱 어렵다.

$$f_v = f_c * (1 + ((\|p_{c,t} \Delta t - p_t + \Delta t\|) * d$$

$$f_t = p_{c,t} - p_c$$

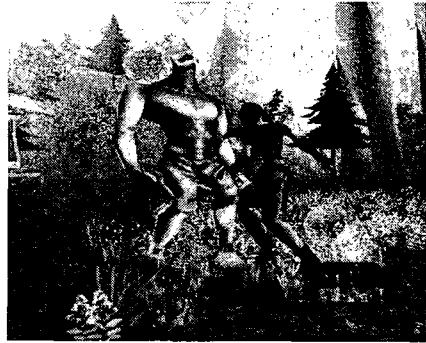
$$f_v = -p_t + f_v$$

$$p_n = p_c + (f_t + f_v)$$

이를 해결하기 위하여, 스프링의 경직도(stiffness) k 를 증가시키거나 적분되는 시간의 간격 Δt 를 조정하여 처리할 수 있으나, 비현실적인 탄력 현상이 발생하거나 안정된 상태를 위한 처리 시간이 증가하는 단점을 가진다.

4. 실험 결과

본 실험에서는 햅틱 효과를 실시간으로 렌더링할 수 있는 방법을 제시하였다. 그림 5는 격투의 게임에서 체감효과를 생성할 수 있는 장면들을 제시하였다. 캐릭터간의 충돌과 가상환경에 실재하는 객체들 사이에 발생하는 충돌 상황에 대하여 체감효과를 1000Hz 속도로 생성하고 있음을 보이고 있다.



이 처리 결과는 기존의 워크스테이션용인 PhanTom에서 가능하던 대화형 햅틱효과를 일반적인 PC 환경에서도 생성할 수 있는 데에 큰 의미가 있다.

5. 결론 및 향후 연구과제

본 논문에서는 실시간 체감 시스템을 다루었다. 또한, 체감 효과에 필요한 햅틱 렌더링을 실시간으로 생성할 수 있는 방법을 제시하였다. 단순화된 동역학 방법의 의한 하이브리드 접근 방식으로 해결함으로써, 안정적이면서 신속하게 효과 생성이 가능하도록 하였다.

제안하는 방법은 일반적인 PC 환경에서도 1000Hz의 속도로 체감효과 생성속도를 나타냈으며, 고품질의 게임 환경에서도 그 적용이 가능하도록 한다.

향후에는 제안된 방법의 처리 시간을 정량적으로 분석하여, 기존 방법들과 비교하고, 이를 바탕으로 전반적인 개선을 수행할 필요가 있다.

[참고문헌]

- [1] M.Desbrun, P.Schroder, and A. Barr, "Interactive animation of structured deformable objects," *Graphics Interface '99*, 1999.
- [2] D.E.Breen, et al., "Predicting the drape of woven cloth using interacting particles," *SIGGRAPH '94*, pp.365-372, 1994.
- [3] D.Baraff and A.Witkin. "Large steps in cloth simulation," *SIGGRAPH '98*, Vol.32, pp.43-52, 1998.
- [4] S.P.Timoshenko and J.N.Goodier, *Theory of Elasticity*, McGraw-Hill, 1970.
- [5] Y.M. Kang, et al, "Real-Time Animation Technique for flexible fabricparts," *WSCG processing*, pp 322-329, 2000.
- [6] X. Provat "Deformation Constraints in mass-spring model to describe rigid cloth behavior".*Proceedings of graphics interface*, pp. 147-154, 1995.
- [7] H.J. Bae, et al., "Procedural Approach to generate Real Time Motions of Cloth," *Proceeding of KISS*, Vold.28(2), pp616-618, 2001.