

물리 기반 모션 엔진의 구현

정일권*, 박기주, 이인호

한국전자통신연구원, VR 연구부, 디지털액터연구팀
전화 : 042-860-1230

Implementation of a Physically Based Motion Engine

Il-Kwon Jeong*, Ki-Ju Park, Inho Lee

Digital Actor Research Team, VR R&D Department
Electronics and Telecommunications Research Institute
E-mail : jik@etri.re.kr*

Abstract

Recent performance improvement in computer and graphics hardware makes it possible to simulate a physical phenomenon in real time. VR department at ETRI has implemented a fast and robust physically based motion engine (PBM) for their general-purpose 3D online game engine, 'Dream 3D'. This paper shows the underlying algorithms of the PBM and introduces the structure and implementation results of it briefly.

I. 서론

키프레임이나 모션캡처에 기반한 애니메이션은 디자이너의 수작업 혹은 연기자의 동작에 기반한 애니메이션을 생성시켜준다. 하지만, 키프레임에 의한 애니메이션은 물리적으로 타당하고 사실적인 애니메이션을 표현하는데 한계가 있으며, 모션캡처는 위험하거나 연기가 불가능한 동작은 캡처하기 어렵다는 단점이 있다.

최근 컴퓨터와 그래픽 카드의 성능이 날로 고성능화 되어 감에 따라 복잡한 수치 계산이나 애니메이션을 PC에서 표현하는 것은 더 이상 어려운 일이 아니게 되었다. 최근에 몇몇 상용 물리 기반 모션 엔진이 등장한

것도 같은 맥락에서 이해할 수 있다.

본 논문은 뉴턴의 운동 법칙을 기반으로 하는 물리 기반 모션 엔진(Physically Based Motion Engine; PBM)에 대해 소개하고자 한다. ETRI 가상현실연구부에서는 Dream3D 온라인 3D 게임 엔진의 일부로 PBM을 개발하였으며, 이를 통해 물리 법칙에 따른 매우 사실적인 물체 애니메이션 및 매우 현실적인 사용자와의 상호 작용이 가능하다. 2 장에서는 구현된 PBM의 특징을 간략히 소개하고, 3 장에서 구현에 사용된 알고리즘에 대해 설명한다. 4 장에서는 구현 결과를 제시하고 5 장에서 결론을 내린다.

II. 구현된 PBM의 특징

PBM은 3차원의 가상 공간상에 있는 물체들의 움직임을 물리 법칙에 따라 실시간으로 계산해서 시뮬레이션 해주는 소프트웨어라 할 수 있다. PBM은 크게 강체 엔진과 비정형 물체 엔진으로 나눌 수 있으며, 강체 엔진은 충돌 감지와 동역학을 이용한 충돌 처리 부분으로 이루어져 있으며, 비정형 물체 엔진은 널리 쓰이는 질량-스프링 모델을 사용하여 구현하였다 [1].

개발된 PBM은 다음과 같은 특징을 가진다.

1) 강체에 대한 고급 충돌 처리 및 제한 조건 처리 - 강체 객체들에 대한 직육면체 또는 구 형태의 바운딩 박스를 지원하며 강체들간의 실시간 충돌 감지, 충돌 후 처리를 지원한다.

2) 스프링 객체, spherical 및 hinge 조인트 지원 - 다양한 물리 환경 시뮬레이션을 위하여 스프링 객체, spherical 및 hinge 조인트를 지원한다. 이들 객체를 이용하여 연결된 강체들은 보다 다양하고 사실적인 움직임을 표현할 수 있으며 특히 인간과 같은 복잡한 다관절체 표현에 유용하다.

3) 실시간 수치 적분 및 강체 시뮬레이션 - 일반적인 Euler 방법 외에도 Runge-Kutta 4 차 수치 적분 방법을 이용하여 빠르고 정확한 시뮬레이션을 가능케 하였다.

4) Height 맵을 이용한 바닥과의 충돌 처리 - 게임 환경에서 보다 용이하게 적용될 수 있도록 바닥(게임 지형) 충돌 처리를 지원한다.

5) 비정형 객체 시뮬레이션 모듈 - 천, 깃발과 같은 비정형 객체의 사실적인 움직임 표현을 위하여 비정형 물체 실시간 시뮬레이션을 지원한다.

III. PBM 에 사용된 알고리즘

구현된 PBM 은 강체와 비정형 객체 부분으로 나눌 수 있다.

1. 강체 엔진

강체 시뮬레이션은 크게 충돌 감지와 충돌 처리 부분으로 나눌 수 있다. 엔진 전체 알고리즘은 [4]를 기반으로 작성되었다. 구와 구의 충돌 감지는 중심과 반지름을 이용하여 쉽게 처리할 수 있으며, 구와 직육면체와의 충돌 감지도 직육면체의 각 모서리 및 꼭지점과 구의 중심을 비교하여 처리할 수 있다. 직육면체끼리의 충돌 감지는 [3]에서 제시된 방법을 사용하였다. 이를 쓰면 총 15 번의 테스트로 충돌 감지가 가능하다. 움직이는 각도에 제약이 있을 때의 문제의 해결을 위해 [2]를 구현하여 해결하였으며 그 외의 구속 동역학은 [4]를 기본으로 하였다.

먼저 구속되지 않은 강체의 시뮬레이션에 대해 간단히 기술한다. 질점의 경우 그 움직임은 다음과 같이 기술된다.

$$\frac{d}{dt}Y(t) = \frac{d}{dt} \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ F(t)/m \end{pmatrix} \quad (1)$$

하지만, 강체는 그림 1 에서처럼 선속도 외에도 회전 성분인 각속도를 가진다.

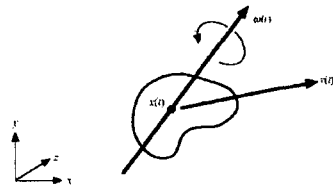


그림 1. 강체의 선속도와 각속도

위와 같은 강체의 회전은 벡터 $\omega(t)$ 로 나타낼 수 있다. $\omega(t)$ 의 방향은 물체가 회전하는 축의 방향을, $\omega(t)$ 의 크기, 즉 $|\omega(t)|$ 는 각속력을 나타낸다. 선형 속도의 경우, 위치의 시간에 대한 미분이 속도가 된다. 물체의 회전(자세)을 나타내는데 3x3 회전 행렬 $R(t)$ 를 사용한다면, 각속도 $\omega(t)$ 와 $R(t)$ 의 관계는 다음과 같다.

$$\dot{R}(t) = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} R(t) = \omega(t)^* R(t) \quad (2)$$

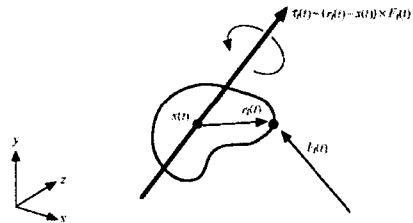


그림 2. $r_i(t)$ 에 가해지는 힘 $F_i(t)$ 에 의한 회전력 $\tau_i(t)$

그림 2 에서, $r_i(t)$ 에 가해지는 외부 회전력 $\tau_i(t)$ 는 다음과 같이 정의된다.

$$\tau_i(t) = (r_i(t) - x(t)) \times F_i(t) \quad (3)$$

물체에 가해지는 총 외부 힘 $F(t)$ 는 $F_i(t)$ 의 합이며 총 회전력도 유사하게 정의된다.

$$F(t) = \sum F_i(t) \quad (4)$$

$$\tau(t) = \sum \tau_i(t) = \sum (r_i(t) - x(t)) \times F_i(t) \quad (5)$$

강체의 선형운동량 $P(t) = Mv(t)$, $\dot{P}(t) = F(t)$ 인 관계가 있다. 유사하게 각운동량 $L(t)$ 는 다음과 같이 정의된다. $L(t) = I(t)\omega(t)$. 여기서 $I(t)$ 는 3×3 행렬로 관성 텐서라고 불리는데 물체의 질량 분포를 나타낸다. $L(t)$ 와 총회전력 $\tau(t)$ 는,

$$\dot{L}(t) = \tau(t) \quad (6)$$

인 관계가 있다. 따라서, 최종적인 강체의 상태 방정식은 다음과 같다.

$$\frac{d}{dt} Y(t) = \frac{d}{dt} \begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ \omega(t) \times R(t) \\ F(t) \\ \tau(t) \end{pmatrix} \quad (7)$$

$I(t) = R(t)I_{body}R(t)^T$ 인 관계를 사용하면 $I(t)$ 를 쉽게 계산할 수 있다. I_{body} 는 물체 지역 좌표계에서 계산된 관성 텐서다.

두 강체가 충돌하면 충돌 반응을 처리해야 한다. 충돌 지점을 p , 충돌 평면의 법선 벡터를 k 라고 하고, 충돌 후 변수에 $\bar{\quad}$ 표시를 하고, 가해진 충격량을

R 이라 하면 다음의 네 식이 성립한다.

$$\begin{aligned} \bar{m}_1 v_1 &= m_1 v_1 + R \\ \bar{m}_2 v_2 &= m_2 v_2 - R \\ \bar{I}_1 \omega_1 &= I_1 \omega_1 + \rho_1 \times R \\ \bar{I}_2 \omega_2 &= I_2 \omega_2 - \rho_2 \times R \end{aligned} \quad (8)$$

여기서 ρ_1 과 ρ_2 는 각 강체의 질량 중심에서 충돌 지점의 차를 나타내는 벡터다. 벡터의 성분까지 생각할 때 식 (8)은 총 12 개의 식이고 미지수는 5 개의 벡터 즉, 15 개가 된다. 이때, 탄성계수가 0 이고 충돌 표면에 마찰력이 없어 충격량이 충돌 표면의 수직 방향으로만 작용한다는 가정을 이용하여 충격량 계산이 가능하며, 이를 이용하여 실제 탄성계수에 대한 충격량도 계산할 수 있다. 마찰 계수가 존재하고 미끄러지는 상태일 경우엔, 충격량의 수평 성분과 수직 성분의 크기의 비인 마찰 계수를 새로운 조건식으로 하여 풀 수 있다.

구속 동역학은 시스템의 구성에 힘 외에 강체의 움직임에 제약을 포함한다는 문제이다. 자세한 내용은 참고 문헌 [2]와 [4]를 참조하기 바란다.

2. 비정형 물체 엔진

구현된 비정형 물체 엔진은 널리 사용되는 질량-스프링 모델을 사용하여 구현하였다. 젤리와 같은 3 차원 물체의 충돌 반응은 아직 구현되지 않았으며, 깃발이나 천과 같은 2 차원 형태를 지원한다. 질량-스프링 모델은 다음과 같은 스프링에 작용하는 힘에 대한 기본식으로부터 출발한다.

$$F(t) = -k(x(t) - L) \quad (9)$$

여기서 L 은 정상 상태의 스프링 길이이며, k 는 스프링 상수이다. 격자 모양으로 질점을 배치하고 각 질점을 스프링으로 연결함으로써 옷감을 모사할 수 있다. 뒤틀림을 억제하는 shear 스프링과 접힘을

방지하는 band 스프링도 추가 가능하다.

IV. 구현 결과

구현된 PBM은 IBM Pentium III급의 PC, Windows, DirectX 8 이상의 환경에서 원활히 동작한다. PBM은 크게 다음과 같은 4개의 대표 클래스로 이루어져 있다.

□ CGeSimulationWorld는 동역학과 충돌 시뮬레이션을 위한 world 클래스인데, 사용자는 이 클래스 객체를 만들고 각종 강체 및 제한조건 들을 설정/등록한 후 Animation()을 통하여 물리 환경을 시뮬레이션하게 된다. 충돌 감지와 dynamics를 연결시키기 위한 클래스인 CGeCDWorld와 CGeDynamicsWorld 객체와 CGeSimObject 객체들을 멤버로 가진다.

□ CGeCDWorld는 Collision Detection 계산을 하게 되는 World 클래스로서 시스템에 존재하는 충돌 감지를 위한 모든 객체들을 유지하고 관리한다.

□ CGeDynamicWorld는 시스템에 존재하는 모든 강체와 스프링을 관리하고 강체의 현재 위치를 계산하기 위한 클래스이다.

□ CGeSimObject 는 물체의 충돌 감지를 위한 자료와 dynamics 를 위한 자료를 유지하기 위한 클래스이다.

그림 3 은 기본적인 충돌 처리 기능을 보여주는 스크린 샷으로서 구와 박스가 충돌할 때의 반응이다.



그림 3. 구와 박스 객체의 충돌



그림 4. 사람 형태의 관절체와 계단과의 충돌

그림 4 는 구형 조인트와 hinge 조인트를 이용하여 구성된 사람형 관절체가 계단에 자유 낙하 되었을 때의 충돌 반응이다. 그림 5는 Dream 3D 게임 엔진에 포함된

PBM 이 동작하는 모습이다. 스프링으로 연결된 바닥판들로 구성된 구름다리과 비정형 객체인 깃발의 나부김을 볼 수 있다. 제시된 예들은 렌더링을 제외할 때 2GHz CPU 에서 초당 100~500 프레임으로 계산되었다.

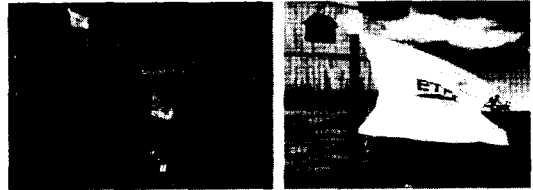


그림 5. 드림 3D 에서의 PBM: 구름다리와 깃발

V. 결론

본 논문에서는 충돌 감지와 구속 동역학을 이용한 충돌 처리를 이용하여 구현한 물리 기반 모션 엔진을 소개하였다. 스프링과 다양한 조인트, 안정적인 수치 적분, 하이트 맵, 비정형 객체 등을 지원함으로써 스포츠 게임이나 게임 내 물리 현상 구현에 효과적으로 사용 가능하도록 하였다.

References

- [1] D.M. Bourg, *Physics for Game Developers*, O'reilly, 2001
- [2] D. Baraff, "Fast contact force computation for nonpenetrating rigid bodies", *ACM Computer Graphics (proc. of SIGGRAPH '94)*, 23-34, 1994
- [3] S. Gottschalk, M.C. Lin, and D. Manocha, "OBBTree: A hierachical structure for rapid interference detection", *ACM Computer Graphics (proc. of SIGGRAPH '96)*, 171-180, 1996
- [4] *Physically based modeling: principles and practice*, Course notes from SIGGRAPH '97, 1997