

풍선 속 공기로 인한 표면의 회전과 변형을 효율적으로 처리하기 위한 PBD 기반 프레임워크

이주용^o, 김종현^{*}

^o인하대학교 정보통신공학과,

^{*}인하대학교 소프트웨어융합대학 디자인테크놀로지학과

e-mail: jonhyunkim@inha.ac.kr

PBD-based Framework for Efficiently Handling Rotation and Deformation of Surfaces due to Air in Balloon

Ju-yong Lee^o, Jong-Hyun Kim^{*}

^oDept. of Information and Communication Engineering, Inha University,

^{*}College of Software and Convergence (Dept. of Design Technology), Inha University

● 요약 ●

본 논문에서는 위치기반 동역학(Position based dynamics, PBD)을 기반으로 하는 프레임워크를 활용하여 풍선 내 공기로 인한 표면의 회전과 변형을 효율적으로 표현할 수 있는 새로운 방법을 제안한다. 기존의 경우 볼륨 형태인 다면체 메쉬(Tetrahedral mesh)를 활용하여 표면 내부를 모델링 하거나 입자 기반의 유체 시뮬레이션을 통하여 공기역학을 계산해야 하지만, 각각의 동역학뿐만 아니라 상호작용까지 고려해야 되기 때문에 계산량이 커서 다양한 분야에서 활용하기 어렵다. 본 논문에서는 이 문제를 효율적으로 풀어내기 위해 공기의 움직임을 파악하기 위한 유체 시뮬레이션을 계산하지 않고도, 풍선 내 바람에 의한 표면 수축 및 확장을 PBD기반으로 풀어내고, 공기가 빠질 때 나타나는 표면의 회전을 효율적으로 계산할 수 있다. 본 논문에서 제안하는 방법은 정점(Vertex)의 개수가 많은 모델에서도 실시간 처리되는 결과를 보여 줄 수 있기 때문에 게임뿐만 아니라 실시간을 요구하는 물리기반 가상환경 구축에 활용될 수 있다.

키워드: 위치기반 동역학(Position based dynamics), 옷감 시뮬레이션(cloth Simulation), 풍선(Balloon), 공기역학(aerodynamics), 공기저항(Air resistance)

I. Introduction

본 논문에서는 PBD[1]를 기반으로 하는 프레임워크를 활용하여 공기에 의한 풍선의 상호작용에서 나타나는 물리적 효과를 효율적으로 계산하는 방법을 제안한다. 기존 방식의 경우 다면체 메쉬나 입자를 활용하여 내부를 이산화하고, 유체 시뮬레이션을 통해 풍선 내 공기의 흐름을 계산할 수 있지만[2], 계산량이 크고 변형체 표면의 경계 조건에 따라 결과가 달라질 수 있기 때문에 이를 수치해석으로 풀어나가는 것을 게임이나 실시간 애플리케이션에서 활용하기에는 비효율적인 접근법이다. 본 논문에서는 이 문제를 PBD 기반으로 계산하기 때문에 매우 효율적으로 계산할 수 있고, 실시간으로 동작하기 때문에 다양한 산업에서 활용할 수 있다.

II. Preliminaries

PBD는 옷감 시뮬레이션과 같은 변형체를 위치기반으로 풀어나간 동역학이다. PBD의 핵심 아이디어는 오일러 방법(Euler's method)으로 외력에 의한 다음 위치를 추정한 후에 후크의 법칙을 기반으로 한 제약(Constraint) 조건을 이용하여 내력(Internal force)을 유지하는 최적화 시스템을 만드는 것이다. PBD 해법(Solver)은 오일러 방법에 의한 위치 추정과 제약조건에 의한 위치 계산으로 나눌 수 있다. 오일러 방법에 의한 위치 추정은 아래 수식을 이용하여 계산한다 (수식 1 참조).

$$\begin{aligned}v_i &= v_i + \Delta t w_i f_{ext}(x_i) \\p_i &= x_i + \Delta t v_i\end{aligned}\quad (1)$$

여기서 v 는 속도, w 는 역질량(Inverse mass), f_{ext} 는 외력, x 는 이전 위치, p 는 추정된 위치, i 는 각 i 번째 정점의 정보를 의미한다. 추정된 위치인 p 를 보정하여 제약조건을 만족시키기 위해 가우스자

이텔 방법(Gauss-Seidel method)을 사용하며, 각 제약에 대한 방정식은 비선형이기 때문에 반복적으로 오차를 줄여나가는 방식으로 수치적 해를 찾는다. 그리고 마지막으로 아래 수식을 이용하여 다음 시간에서의 위치와 속도를 계산한다 (수식 2 참조).

$$\begin{aligned} v_i &= (p_i - x_i) / \Delta t \\ x_i &= p_i \end{aligned} \quad (2)$$

PBD에서 가장 핵심은 점들을 제약조건에 투영시켜 최적의 해를 찾는 것이다. 이러한 제약은 외부 힘이 아닌 후크의 법칙과 같은 내부 힘에 의한 것이기 때문에 운동량이 보존되어야 한다. 이를 위해 아래 수식에서 C 의 기울기(Gradient)인 ∇C 를 강체(Rigidbody)로 가정하고, 제약조건에 대한 최적 해를 찾는다 (수식 3 참조).

$$C(p + \Delta p) \approx C(p) + \nabla_p C(p) \cdot \Delta p = 0 \quad (3)$$

위 수식에서 ∇C 를 강체로 가정하기 위해 Δp 의 방향을 ∇C 로 할당하고, 수식을 전개하면 각 정점이 이동해야 할 위치 변화인 Δp 를 계산할 수 있다. 수식 3을 개별적인 정점에 대해 정리하면 수식 4와 같다.

$$\begin{aligned} \Delta p_i &= -sw_i \nabla_{p_i} C(p_1, \dots, p_n) \\ s &= \frac{C(p_1, \dots, p_n)}{\sum_j w_j |\nabla_{p_j} C(p_1, \dots, p_n)|^2} \end{aligned} \quad (4)$$

III. The Proposed Scheme

1. Constraint

본 논문에서 이용한 제약은 길이와 부피에 대한 것으로 나눌 수 있다. 길이에 대한 제약조건은 질량스프링 시스템과 같이 초기 길이로 유지하려는 성질을 활용하며 수식 5를 통해 계산한다.

$$C(p_1, p_2) = |p_1 - p_2| - d \quad (5)$$

위 수식을 수식 4에 대입하면 수식 6과 같은 결과가 나오는데 이를 거리 제약조건에 대입하면 수식 3을 만족시킬 수 있다 (Fig. 1 참조).

$$\begin{aligned} \Delta p_1 &= -\frac{w_1}{w_1 + w_2} (|p_1 - p_2| - d) \frac{p_1 - p_2}{|p_1 - p_2|} \\ \Delta p_2 &= +\frac{w_2}{w_1 + w_2} (|p_1 - p_2| - d) \frac{p_1 - p_2}{|p_1 - p_2|} \end{aligned} \quad (6)$$



Fig. 1. Distance constraint.

본 논문에서는 길이 제약조건을 이용하여 정점들 사이를 이어주는 구조적 스프링(Structural spring)과 교차 되는 에지(Cross-over

edge) 사이를 이어주는 굽힘 스프링(Bend spring)을 만들어 주었다. 본 논문에서는 삼각형 메쉬를 사용하기 때문에 구조적 스프링은 삼각형의 정점 인덱스인 [0,1,2] 순서대로 이어준다 (Fig. 2a 참조). 그리고 굽힘 스프링은 선택한 정점을 제외한 두 정점의 이웃 정점 중에 접치는 정점 두 개를 이어주면 에지를 공유하는 두 삼각형의 완전히 반대편에 있는 두 정점을 이어줄 수 있다 (Fig. 2b 참조).

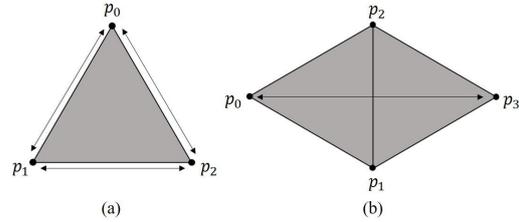


Fig. 2. Generation of structural and bend spring.

다음으로 부피에 대한 제약은 길이와 마찬가지로 현재 부피에 초기 부피를 빼서 계산한다 (수식 7 참조). 부피에 대한 기울기도 수식 8을 이용하여 계산한다.

$$C(p_1, \dots, p_n) = \left(\sum_{i=1}^{n_{\Delta s}} (p_{t'_i} \times p_{t''_i}) \cdot p_{t'_i} \right) - V_0 \quad (7)$$

$$\nabla_{p_i} C = \sum_{j: t'_j=i} (p_{t'_j} \times p_{t''_j}) + \sum_{j: t''_j=i} (p_{t'_j} \times p_{t''_j}) + \sum_{j: t'_j=i} (p_{t'_j} \times p_{t''_j}) \quad (8)$$

위 수식을 수식 4에 대입하여 Δp_i 를 구할 수 있다. 이를 적용하면 초기 부피에 따라 달라지는 결과를 얻을 수 있다. Fig. 3은 토끼모델의 부피에 따른 제약 때문에 표면이 수축하고 확장되는 결과를 잘 보여주 고 있다.



Fig. 3. Volume constraint simulation.

2. Balloon effect

본 논문에서 풍선 효과를 구현하기 위해 부피에 대한 제약에 대입하는 초기 부피인 V_0 를 활용한다. 공기를 넣을 때는 V_0 를 높이고 빠질 때는 낮추면 자연스럽게 물체의 부피가 늘어나고 줄어드는 표면 변형이 나타난다. 공기가 들어가고 나갈 때 풍선의 부피가 늘어나고 줄어드는 이유는 풍선 내부의 압력 변화 때문이다. 풍선 내부의 압력이 대기압보다 높으면 부피가 늘어나고 낮으면 줄어든다. 따라서 풍선 내부의 압력인 PV 을 이상기체 법칙(Ideal gas law)인 수식 9를 계산하고 대기압과 같아지면 공기가 빠지는 것이 멈추도록 하였다.

$$PV = nRT \quad (9)$$

여기서 n 은 기체의 몰 수이고, R 과 T 는 각각 기체 상수와 온도를 나타낸다. 본 논문에서 대기압은 101,325Pa이고 온도는 293.15K, 그리고 기체는 헬륨으로 가정하였다. 또한 기체의 몰 수는 풍선 내부의 추가된 부피로 근사하였다.

3. External force

본 논문에서 외부 힘은 총 4가지이며 다음과 같다 : 중력, 부력, 공기저항, 반작용력. 먼저 중력인 g 는 기본적인 값인 $9.8m/s^2$ 으로 설정했고, 부력인 F_{buoy} 은 수식 10에 의해 계산된다. 따라서 부력의 크기가 중력보다 커지면 풍선은 떠 오르게 된다. 여기서 밀도는 다음과 같이 설정했다 : 풍선 내부인 ρ_{ext} 는 헬륨의 밀도, 외부인 ρ_{inter} 는 일반 대기의 밀도로 가정.

$$F_{buoy} = -gV(\rho_{ext} - \rho_{inter}) \quad (10)$$

여기서 V 는 풍선의 부피를 나타낸다. 다음으로 공기저항인 F_{drag} 은 수식 11을 이용하여 계산하며, 정점이 움직이는 방향에 따라 기중치를 주기 위해서 속도 벡터에 법선 벡터를 내적 하여 공기저항의 크기를 제한한다. 여기서 저항 계수인 C_d 는 0.5로 설정했다.

$$F_{drag} = -\frac{1}{2}\rho v^2 AC_d(v \cdot n) / \|v\| \quad (11)$$

마지막으로 풍선에서 바람이 빠지는 효과는 반작용력을 기반으로 설계했다. 반작용력을 계산하기 전 베르누이 방정식에 대한 이해가 필요하다. 베르누이 방정식은 유체가 흐를 때 속력, 압력, 높이의 관계에 대한 방정식이다 (수식 12 참조).

$$P + \rho gh + \frac{1}{2}\rho v^2 = const \quad (12)$$

위 수식에서 좌변의 두 번째 항은 위치에너지, 세 번째 항은 운동에너지에 대한 식이며, 초기에는 풍선에서 빠져나가는 유체의 운동에너지가 존재하지 않기 때문에 초기화를 할 때 우변의 상수를 계산할 수 있다. 이 상수는 항상 일정해서 좌변의 속력인 v 에 대해서 식을 정리하면 풍선에서 빠져나가는 유체의 속력을 구할 수 있다. 여기에 공기가 빠져나가는 정점의 법선 벡터를 곱해서 풍선에서 빠져나가는 유체의 작용력으로 가정하고 반대 방향으로 반작용력을 적용하여 풍선에서 공기가 빠져나가는 형태를 설계했다.

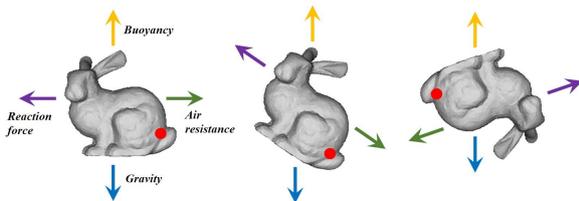


Fig. 4. The external force applied when air is let out of a balloon.

Fig 4는 빨간 점의 법선 벡터 방향으로 바람이 빠진다고 가정했을 때 풍선에 적용되는 전체 외력을 나타낸 그림이다.

4. Torque

일반적으로 풍선에서 바람이 빠질 때는 직선운동만 하는 것이 아니라 회전도 같이 표현된다. 하지만 풍선은 변형체이기 때문에 토크(Torque)를 적용하려면 공간을 클러스터 단위로 분할해서 각 영역마다 회전을 계산하거나, 정점마다 회전을 추출해야 하지만 이 과정들은 계산 과정이 커지기 때문에 비효율적이다. 본 논문에서는 풍선 효과를 효율적으로 구현하고자 하기 때문에 회전은 강체로 가정하여 하나의 회전 행렬로 통합했다. 실제로 풍선은 공기가 차면 법선 벡터 방향으로 부풀어 오르기 때문에 일반적인 변형체와 같은 복잡한 회전을 갖지 않아서 비주얼 시뮬레이션 관점에서는 좋은 결과를 만들어 낼 수 있다.

일반적으로 토크는 힘의 작용점과 힘 벡터를 외적 하여 계산할 수 있으며 (수식 13 참조), 강체이므로 작용점의 위치를 객체의 로컬 좌표 기준으로 한다. 이때 로컬 좌표의 중심은 질량 중심이 된다.

$$\tau = p \times F \quad (13)$$

수식 1을 보면 힘과 역질량을 활용하여 가속도를 구하는데, 회전 운동 또한 회전력 토크와 역 관성 텐서로 수식 14에서 각속도를 구할 수 있다. 그리고 여기서 계산한 각속도를 이용하여 유도 과정을 거치면 수식 15에서 회전 방향(Orientation)을 계산할 수 있다. 이를 회전 행렬로 변환 후 로컬 좌표계에서 현재 위치를 곱하면 회전 운동을 표현할 수 있다. 수식 15에서 ω 는 각속도를 사원수(Quaternion)로 변환한 것이다.

$$\dot{\theta} = \dot{\theta} + \Delta t I^{-1} \tau \quad (14)$$

$$\theta = \theta + \frac{\Delta t}{2} \omega \theta \quad (15)$$

여기서 역 관성 텐서는 수식 16과 17에 의해 계산되며, 역 관성 텐서는 로컬 좌표에서의 값이다. 하지만, 다른 값들은 월드 좌표에서의 값이므로 이를 월드 좌표로 변경해준 뒤 사용해야 한다. 그리고 강체는 역 관성 텐서를 전처리 과정에서 한 번 계산하지만, 본 논문처럼 변형체의 경우 매번 업데이트해야 한다.

$$I_{xy} = \sum_{i=1}^n m_i (p_i \cdot x)(p_i \cdot y) \quad (16)$$

$$I = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix} \quad (17)$$

IV. Results

본 논문에서는 공기에 따른 풍선의 물리적 변형과 회전을 PBD에서의 제약을 활용하여 효율적으로 표현하는 방법을 제안하였다. 본 논문에서 제안한 방법을 사용하면 풍선 내부와 외부의 유체를 직접 모델링하지 않고도 풍선 효과를 보여줄 수 있다. Fig 5에서는 토끼모델 내부에 바람이 가득 찼을 때와 빠졌을 때의 팽창과 수축을 보여주는 결과이다.



Fig. 5. The physical deformation and rotation of the balloon in response to air using our method.

Fig. 6에서는 토크로 인해 회전하는 물체를 보여주기 위해 4개의 정점을 선정해서 각기 다른 색으로 이동 하는 궤적을 시각화했으며, 풍선에서 공기가 많이 빠질 때 회전력이 증가됨을 잘 표현했다.

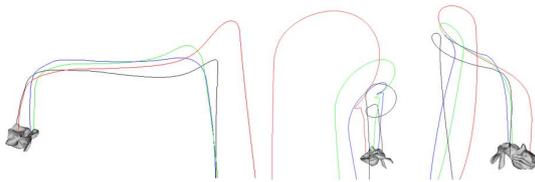


Fig. 6. Visualization of the rotation represented when the wind is taken out of a balloon.

Fig. 7은 풍선 내부에 공기를 넣다가 갑자기 빼는 장면을 제작한 것이며, 공기에 얼마나 차 있냐에 따라서 풍선의 움직임(변형 및 회전)이 자연스럽게 변하는 것을 보여준다. 공기가 차있을 때는 부력에 의해 상승하다가 바람이 빠질 때 그 움직임의 역으로 변하는 것을 궤적으로 잘 보여주고 있다.

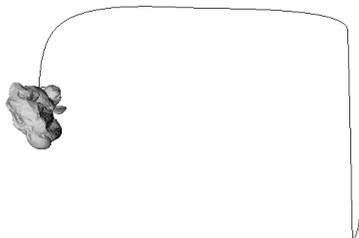


Fig. 7. Putting air inside a balloon and suddenly letting it out.

V. Conclusions

본 논문에서는 풍선 효과를 PBD 프레임워크를 통해 효율적으로 표현하였다. 풍선의 부피는 제약을 통해, 압력은 이상기체 방정식을 통해, 주변 공기의 흐름은 외부 힘을 통해, 공기가 빠지며 날아가는 것은 베르누이 방정식을 통해, 회전 운동은 토크를 통해 구현하였다. 이는 내부와 외부의 유체 흐름을 입자나 볼륨 메쉬로 명시적으로 모델링하지 않아도 실제 풍선의 물리적 움직임과 유사한 형태를 시뮬레이션했다. 향후 풍선이 맞닿아 있는 경계 부근에서 나타나는 풍선의 물리적 효과에 대해서 추가로 연구할 계획이다.

REFERENCES

- [1] Müller, Matthias, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. "Position based dynamics." *Journal of Visual Communication and Image Representation* 18, no. 2 (2007): 109-118.
- [2] Jameson, Antony, and Massimiliano Fatica. "Using computational fluid dynamics for aerodynamics." (2006): 1-4.