

이차 행동 기반의 다이나믹 지글 본 애니메이션*

박성준*, 안득용^o, 오성석**^{***}

호서대학교 게임공학과

{sjpark*, oseong**}@hoseo.edu, gotodagate^o@gmail.com

Secondary Action based Dynamic Jiggle-Bone Animation

Sung-Jun Park*, Deug-Yong An^o, Seong-Suk Oh**

Dept of Game Engineering, Hoseo University

요 약

현대 게임 개발에 있어서 장신구와 같은 세밀한 오브젝트에 대한 이차 애니메이션 기술에 대해 많이 연구하고 적용하고 있는 추세이다. 3D 그래픽 도구에서는 이러한 오브젝트에 애니메이션 생성을 위해 지글본 디포머라는 기술을 응용하고 있지만 실시간적으로 변형하기 어렵고 그래픽 개발자들에게 많은 시간을 들여야 한다는 부담이 있다. 반면 물리 엔진을 사용하여 구현할 수도 있지만 게임에서 한 장면 안에 여러 개의 오브젝트를 물리엔진을 이용하여 처리한다면 계산량이 많아져 효율적이지 못하다. 본 논문에서는 실시간 적으로 변형하기 쉬우며 물리엔진과 같은 비슷한 효과를 내기 위한 다이나믹 지글본 애니메이션 알고리즘을 제안한다. 제안한 알고리즘에 대한 성능 평가를 위해 본 의 개수에 따른 성능과 한 장면 안에 여러 개의 지글본 애니메이션을 사용했을 경우에 대해 실험하였고 비교적 효율적인 결과를 도출하였다.

ABSTRACT

The secondary animation technology for the detailed objects including accessories is being studied and applied to the modern game development. The jiggle-bone deformer is used for 3D graphic tools as a technology to create the animation of these objects, but it is disadvantageous in that the real-time modification is difficult and the graphic developers need much time. The secondary animation can also be realized using a physical game engine, but the cost of animation process increases when many objects in a scene of a game are rendered, and it has a low efficiency. This paper proposes a dynamic jiggle-bone animation algorithm, which can be modified in real time and has the similar effect to the physical game engine. To evaluate the performance of the proposed algorithm, tests were conducted with varied number of bones and for the case of one scene with the animation of many jiggle-bones, and the results were adjudged relatively efficient.

Keywords : Jiggle-bone, Secondary Animation, Physics

접수일자 : 2009년 12월 10일

심사완료 : 2010년 01월 06일

※ 이 논문은 2009년도 호서대학교의 재원으로 학술연구비 지원을 받아 수행된 연구임(2009-0077)

※※ 교신저자 : 오성석(oseong@hoseo.edu)

1. 서론

근래의 게임에서는 장신구와 같은 세밀한 오브젝트를 애니메이션 화하여 시각적으로 현실감을 높여주는 기법을 많이 사용하려고 하는 추세이다. 대표적인 예로서 망토, 귀걸이, 바스트, 깃털 등이 있다. 이와 같은 애니메이션 기법들은 게임에 있어서 스케일과 중요도가 적지만 없으면 게임의 연출 효과가 떨어지기 때문에 게임 개발 시 충분히 고려해야 하는 부분이다. 보통은 이러한 세밀한 오브젝트의 동작을 위해 그래픽 개발자들이 손수 작업하는 경향이 많다. 만약 n 개의 장신구에 대해 각각의 애니메이션 작업을 처리한다면 그래픽 개발자에 대해 많은 부담과 함께 게임 개발에 있어서 시간, 비용 등이 소모하게 된다. 이러한 문제를 해결하기 위해 대부분의 게임 개발에 있어서 물리 엔진을 사용하여 처리하고 있다.

물리 엔진은 자연 현상의 효과를 근접하게 표현하기 위해 힘, 중력, 관성, 마찰력, 탄성력 등의 물리적 요소를 다양하게 반영하고 있다[1]. 물리엔진을 사용하면 고 퀄리티의 애니메이션 처리가 가능하나 그만큼 계산량이 많아 중요한 알고리즘 처리 이외는 게임 개발에 있어서 부담이 되는 부분 중의 하나이다. 특히 가볍게 애니메이션 처리되어도 충분한 장신구와 같은 오브젝트를 한 장면 안에 물리엔진을 사용하여 여러 개를 렌더링 한다면 게임의 실시간 플레이어에 있어서 급격한 프레임 저하를 가져올 수 있다.

본 논문에서는 그래픽 개발자의 부담을 줄이고 물리엔진과 같은 효과를 내면서 비용을 절감할 수 있는 이차 행동 기반의 다이내믹 지글본 애니메이션 알고리즘을 제안한다. 지글본 애니메이션은 본 애니메이션을 동적으로 만들어 주는 애니메이션 기법이다[2]. 본 논문에서 제안한 다이내믹 지글본 애니메이션은 기본적으로 외부의 힘, 중력, 탄성력의 3가지 기본 물리적 요소만을 고려하였다. 이 3가지 요소만을 고려한 것은 장신구와 같이 스케일이 적거나 게임 플레이상에서 중요도가 적은 애니

메이션을 처리할 때 가장 기본적으로 고려해야 할 요소이다. 본 논문에서 주장하는 지글본 애니메이션은 물리적 요소를 충분히 고려한 정확한 물리 시뮬레이션 구현 방법이 아니기 때문에 이외에 추가로 물리적인 요소가 고려된다면 물리 엔진을 사용하는 것이 효과적이다. 힘과 중력은 귀걸이, 망토, 천 계열 등에 적용될 수 있으며 탄성력은 투구 위의 깃털과 같은 오브젝트에 적용가능하다. 이러한 세 가지 특성은 일반적으로 스케일이 적은 장신구를 애니메이션 처리할 때 공통적으로 사용되는 요소이므로 범용적으로 사용할 수 있는 장점이 있다. 이외에 고려할 수 있는 물리적인 요소들은 게임의 규모나 특성에 따라 고려되어 질 수 있다.

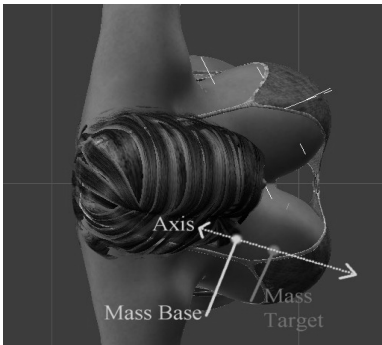
본 논문은 2장에서 게임에서 쓰이는 지글 본 애니메이션에 대한 관련 연구를 소개하고, 3장에서는 실제 개발한 알고리즘에 대해 알아본다. 4장 실험 및 분석에서는 성능 테스트를 통해 제안한 알고리즘에 대한 효율성을 검증하였다. 끝으로 5장에서는 결론 및 향후과제로 마무리 한다.

2. 관련 연구

이차 행동을 애니메이션 화 하는 방법은 다양하게 구현할 수 있다. 일반적으로 그래픽 저작 도구인 3ds MAX에서 지원해주는 지글 디포머란 기술이 있고 물리 엔진을 사용하여 구현하는 방법이 있다.

지글 디포머(jiggle deformer)는 3ds Max나 Maya등의 3D 그래픽 툴에 내장되어 있거나 플러그인으로 추가하여 쉽게 애니메이션의 사실감을 높여주는데 사용되는 도구이다[3]. 지글 디포머는 자동적으로 2차 행동(Secondary Action)을 만들어 낸다. 직접 사용자 조작을 통해 변형하지 않는 것이 특징이다. 지글 디포머는 물체의 표면이나 곡선에 대한 점을 이용해 천천히 또는 빠른 속도로 위, 아래로의 흔들림 변형, 또는 곤충의 더듬이를 진동하는 모습 등을 효과적으로 만들 수 있는 방법이

다. 지글 디포머를 이용하면 편리하게 2차 애니메이션을 생성할 수 있지만 그만큼 그래픽 개발자에게 많은 부담을 주게 된다[4]. 지글 디포머 방법은 물리적인 요소를 고려하지 않고 키 프레임 형태로 본의 애니메이션을 연출하기 때문에 실시간으로 파라미터의 값을 조정하여 변형하기가 쉽지 않다. 본 논문에서 주장하고 있는 다이내믹 지글 본 애니메이션은 실시간으로 가장 기본적인 물리 값을 이용하여 상황에 따라 애니메이션의 연출을 변형할 수 있는 장점이 있다.

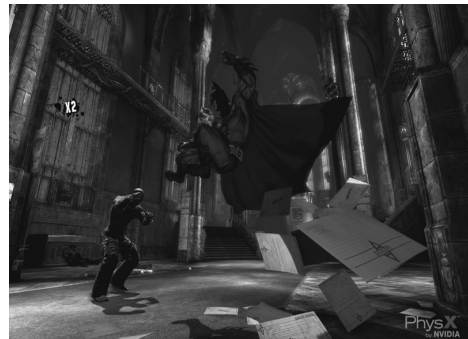


[그림 1] jiggle deformer를 이용한 가슴

NVIDIA사의 PhysX 엔진은 게임에서 사용하는 대표적인 범용 물리엔진이다. PhysX 엔진은 현실감과 물리적 환경효과를 실시간으로 게임에서 즐길 수 있도록 구현해준다. 폭발, 훔날리는 잔해, 출렁이는 물 등의 효과를 PhysX를 통해서 게임에 구현되고 있다[5,6,7]. PhysX는 많은 수의 병렬 프로세서를 이용하여 하드웨어 가속을 한다. 하지만 NVIDIA사의 GPU를 사용해야만 기술의 최적화를 이끌어 낼 수 있다는 단점이 있다. 피직스 엔진을 사용하여 2차 애니메이션을 생성할 수 있지만 정교한 애니메이션 구현인 만큼 계산량이 많아서 필수적인 부분만 구현하는 경향도 있다.

Verlet Intergration은 물리엔진에서 많이 응용하는 알고리즘으로 뉴턴의 운동 법칙을 응용한 기술이다. 이 기술은 주로 과학적 시뮬레이션을 처리하는데 많이 응용되고 있는데 분자 동역학 실험이나 입자들의 궤도를 계산하는 분야에 사용되고 있

다. verlet Intergration의 장점은 간단하고, 빠르고, 매우 안정적으로 처리하고 있어 게임 엔진의 물리분야에 적극 도입하여 사용되고 있다. Verlet Intergration의 물리 시스템은 역 운동학을 사용하여 옷의 움직임, 식물들, 부드럽고 단단한 인체 애니메이션을 시뮬레이션 할 때 주로 사용한다[1,8]. Verlet Integration역시 nVidia사의 PhysX 엔진과 마찬가지로 정교한 애니메이션을 실시간으로 처리하여 2차 애니메이션(Secondary Animation)을 생성할 수 있지만 장신구와 같은 간단한 애니메이션 처리에 있어서 많은 비용이 든다는 단점이 있다.



[그림 2] PhysX 를 이용한 배트맨 : 아캄 수용소

본 논문에서는 지글 디포머의 장점과 실시간으로 처리할 수 있는 물리엔진의 장점을 이용하여 다이내믹 지글 본 알고리즘을 제안한다.

3. 다이내믹 지글 본 애니메이션

본 논문에서는 물리엔진을 사용하지 않고 본 애니메이션과 같은 수준의 연산 값으로 2차 애니메이션(Secondary Animation)을 구현하였다. 다이내믹 지글본 애니메이션을 구현하기 위한 요소로서 힘, 중력, 탄성력 등 3가지의 물리적 특성을 고려하였다. 일반적으로 동적인 움직임을 구현하기 위해서는 위의 3가지 이외에 거리, 질량과 같은 요소를 고려하여 정확한 물리 법칙에 근거하여 구현해야 하며, 이러한 기능들은 범용 물리 엔진에서 잘

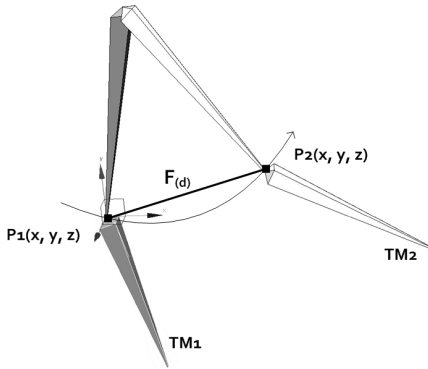
구현되어 있다. 그러나 적은 비용으로 시각적 효율성을 높인다는 측면에서는 기본적인 요소가 필요하다. 망토의 움직임과 같은 경우 힘과 중력의 요소를 생각해야 하며, 투구 위의 깃털과 같은 움직임은 탄성력을 고려해야 한다.

3.1 거리에 따른 외부 힘 계산

[그림 3]에서 보여 지는 본은 망토를 구성하고 있는 본(부모(위)-자식(아래))이다. 본의 자식 본이 망가나 혹은 격자 형태로 구성된다면 좀 더 부드러운 망토를 애니메이션 할 수 있다. [식 1]에서 A_{rot} 은 거리에 따른 힘($F(d)$), 중력($F(g)$), 그리고 마지막으로 되돌아가까지는 탄성력($F(e)$)의 모든 힘을 합하여 얻어지는 최종적인 본의 회전 값을 의미한다.

첫 번째 지글본을 이용한 애니메이션에서 힘의 생성은 부모 본의 움직인 거리에 비례하여 얻게 된다. 일반적으로 생각할 수 있는 방법은 하위 계층으로 이어져 있는 자식 본에 각각의 적당한 힘을 부여하는 방식이다.

그러나 이와 같은 방법은 하위 노드에 있는 자식 본의 개수가 많아질수록 실시간 애니메이션 처리가 어렵다. 또한 유연성이 결여 되어 매번 상황이 바뀔 때 마다 힘의 값을 재설정 해주어야 하는 번거로움이 있다. 본 논문에서는 부모 노드의 움직인 거리가 힘과 비례한다고 정의하고 구현하였다.



[그림 3] 지글 본에서의 힘 요소

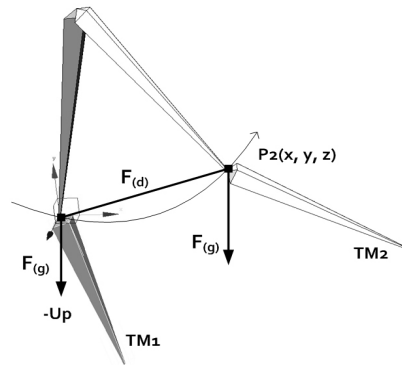
$$A_{Rot} ::= F(d) \dots\dots\dots[\text{식 1}]$$

$$F(d) = \overline{P_1 P_2} \dots\dots\dots[\text{식 2}]$$

위와 같은 방법을 적용할 경우 고려해야 할 사항은 움직인 거리가 클 때, 즉 $F(d)$ 의 값이 클 경우 $P2$ 에서 자식 본의 회전각도가 커져 상대적으로 어색한 움직임을 나타 낼 수 있다. 또한, 자식 본이 여러 개 일 경우에도 하위 본으로 내려 갈수록 회전 값이 누적되면서 애니메이션 되므로 급격한 움직임이 발생할 수 있어 적당한 보간 처리 기법이 필요하다. 본 논문에서는 점차적으로 감소할 수 있도록 하여 이러한 문제점을 해결하였다

3.2 중력

본의 회전 값을 얻기 위해서는 힘의 요소뿐만 아니라 중력에 대한 요소 값을 고려하여 최종 각도를 얻어낼 수 있다. 중력 힘은 [그림 4]에서 보는 바와 같이 $P1$ 과 $P2$ 에서의 상향 벡터의 반대 방향에 중력 가속도 상수(G_k)를 곱한 값이다. [식 5]는 외부 힘과 중력을 최종 고려한 계산식이다.



[그림 4] 지글 본에서의 중력 요소

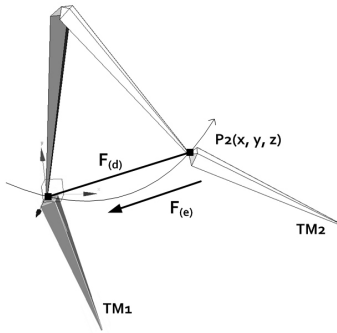
$$A_{Rot} ::= F(d) + F(g) \dots\dots\dots[\text{식 3}]$$

$$F(g) = (-\overline{Up}) * G_k \dots\dots\dots[\text{식 4}]$$

$$A_{Rot} = \overline{P_1 P_2} + (-\overline{Up}) * G_k \dots\dots[\text{식 5}]$$

3.3 탄성력

탄성력은 다른 말로 복원력이라고 하며 외부의 힘에 변형된 물체가 원래의 모양으로 복귀 하는 것을 말한다[9]. 게임에서는 범용 물리엔진에 이미 구현해 놓았으며 대표적인 예는 온라인 게임 마비노기 영웅전에서 사용했던 투구 위의 깃털을 들 수 있다.



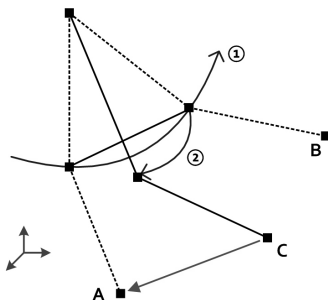
[그림 5] 지글 본에서의 탄성력 요소

$$A_{Rot} ::= F_{(d)} + F_{(g)} + F_{(e)} \dots\dots\dots [식 6]$$

$$F_{(e)} = N(-\overrightarrow{F_{(d)}}) * E_k \dots\dots\dots [식 7]$$

$$A_{Rot} = \overrightarrow{P_1 P_2} + (-\overrightarrow{Up}) * G_k + N(-\overrightarrow{F_{(d)}}) * E_k \dots [식 8]$$

탄성력의 주요 알고리즘은 탄성 방향에 탄성 상수(E_k) 값을 곱한 값으로 정의할 수 있다. 탄성 상수 값은 상황에 따라 파라미터로 임의적으로 설정 가능 하다. [식 8]은 외부 힘, 중력, 탄성력을 고려하여 애니메이션 처리하기 위해 최종적으로 본의 회전 각도를 구해내는 계산식이다.



[그림 6] 흔들리는 본의 위치가 다를 경우

[식 7]에서 보면 탄성 방향은 외부 힘의 상반 벡터로 정의되고 있는 것을 알 수 있다. 이때 중요한 고려사항은 되돌아가는 기준 본의 위치이다. 투구 위에 부착된 깃털의 예를 들어 보면, 캐릭터가 가만히 서있을 경우 깃털은 A위치에서 B의 위치로 움직이게 되며(①, 좌우로 흔들림 경우) 복원되는 위치는 A 지점이다. 이때 캐릭터가 뛰어가는 경우라면 본의 위치는 B에서 C로 움직이게 될 것이고 (②, 앞뒤로 흔들림) 복원 되는 위치는 이전 본인 B의 위치가 아니라 원래 본의 위치인 A 지점이 되어야 한다.

또한 탄성력은 본의 초기위치와 현재 위치의 거리와 비례하여 설정하게 되지만 정비례하게 값을 주면 너무 많이 움직여서 현실감이 많이 떨어지는 애니메이션을 연출하게 된다. 예를 들어 캐릭터가 가만히 서있으면서 숨을 쉴 때 탄성력은 고려하지 않고 애니메이션을 연출하지 않는다. 반면 움직임이 클 경우는 탄성력을 고려하여 초기 위치로 되돌아 갈 수 있도록 해야 한다.

$$\text{if} (Parent\ Bone \equiv A_{position}) \dots\dots\dots [식 9]$$

$$No\ Animation$$

$$\text{else} (Parent\ Bone \equiv C_{position}) \dots\dots\dots [식 10]$$

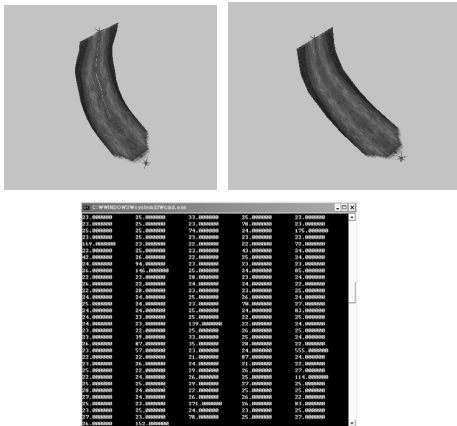
$$F_{(d)} + F_{(e)}$$

지글 본 애니메이션에서 탄성력을 고려할 경우 탄성력($F_{(e)}$)과 외부 힘($F_{(d)}$)을 비교하여 애니메이션 연출 여부를 결정해야 한다. [식 9]는 외부 힘의 값이 지극히 적을 경우를 말하는 것이고 이때는 애니메이션을 연출하지 않는다. [식 10]은 움직임이 클 경우를 말하는 것이고 이때는 탄성력을 고려하여 애니메이션을 연출 할 수 있도록 해야 한다. 한 가지 예를 들자면 게임에서 칼날을 휘두를 경우를 들 수 있다.

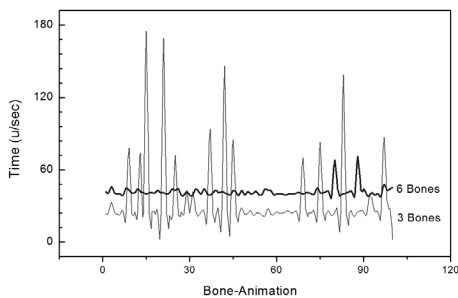
4. 실험 및 분석

본 논문에서 제안한 다이나믹 지글 본 알고리즘의 성능 분석을 위해 3가지 실험을 하였다. 첫 번째로 본의 개수에 따른 처리 시간이며, 두 번째 실험은 지글 본 오브젝트 개수에 따른 처리 시간을 실험하였다. 마지막 실험에서는 범용적으로 사용되는 물리엔진인 nVidia사의 Physics 엔진에서 제공하는 깃발 애니메이션과의 성능 비교 실험을 하였다. 실험한 PC 사양은 CPU(Intel Duo Core 2.66 GHz), RAM(2MB), 그래픽 카드는 NVIDIA GeForce 8600 GT를 사용하였다.

첫 번째 실험에서는 본의 개수가 3개일 경우와 6개일 경우에 대해 조사하였다. [그림 7]은 각각 3개, 6개의 본 애니메이션을 처리한 결과 및 이에 따른 결과 값을 출력하고 있다.



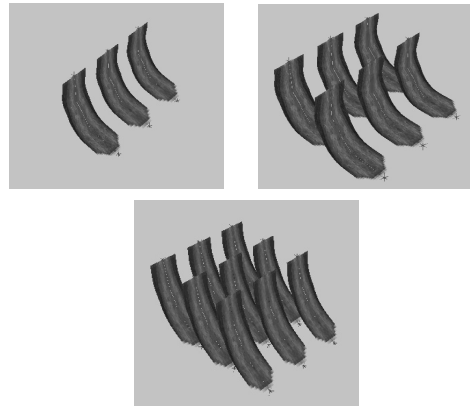
[그림 7] 본이 3개, 6개 일 경우



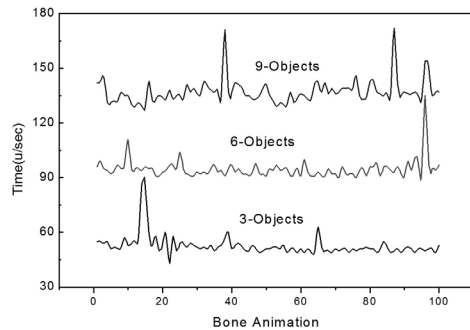
[그림 8] 본의 개수에 따른 처리 시간

[그림 8]에서 보는 바와 같이 본의 개수가 3개일 경우 평균적으로 34.64(u/sec)의 처리 시간을 보였고, 본의 개수가 6개일 경우 평균 41.8(u/sec)의 처리 시간을 보였다. 이는 본의 개수가 2배가 될 경우 계산 량이 기하급수 적으로 급격히 증가하지 않고 일정한 처리 시간을 보인다는 것을 알 수 있다. 중요한 점은 시간 단위가 micro second 단위로 측정하여 매우 빠르다는 것을 알 수 있다.

두 번째 실험은 게임에서 여러 개의 지글 본 애니메이션이 있다는 전제하에 오브젝트의 개수를 증가하여 실험하였다. 오브젝트의 개수는 3개, 6개, 9개 등으로 하였고 이에 대한 각각의 처리 시간을 실험하였다. [그림 9]은 이에 대한 결과화면을 보이고 있다.



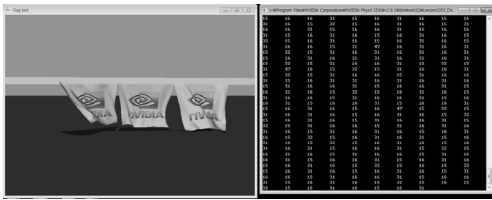
[그림 9] 오브젝트의 개수가 3개, 6개, 9개일 경우



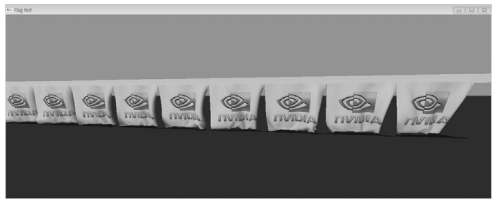
[그림 10] 오브젝트 개수에 따른 처리 시간

두 번째 실험에서는 [그림 10]에서 보는 바와 같이 오브젝트의 개수가 늘어날수록 애니메이션 처리 시간이 배수 증가함을 알 수 있었다. 3개의 오브젝트의 지글본 애니메이션 평균 처리 시간은 52.66(u/sec)이고, 6개일 경우는 94.69(u/sec)이며 9개일 경우는 137.15(u/sec)이다. 첫 번째 실험에서와 마찬가지로 오브젝트의 개수가 늘어날수록 계산량이 급격히 증가하지는 않았다.

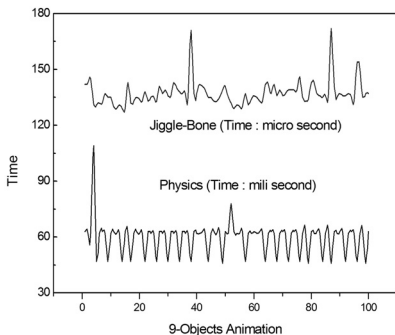
마지막 실험에서는 본 논문에서 제안한 지글본 애니메이션과 nVidia에서 제공한 피직스 물리 엔진을 사용하여 서로 비교 실험하였다. [그림 11]은 3개의 깃발 애니메이션을 처리하여 그 결과 값을 출력한 것이다.



[그림 11] Physics 엔진을 사용한 애니메이션과 그에 따른 처리 시간



[그림 12] 9개의 Flag 애니메이션



[그림 13] Jiggle-Bone과 Physics엔진을 사용한 성능 평가

[그림 12]는 9개의 깃발 오브젝트를 Physics 엔진을 사용하여 애니메이션 한 것이다. [그림 13]은 9개의 오브젝트에 대하여 각각 지글본 애니메이션과 Physics 엔진을 사용한 결과에 대하여 비교 평가한 그림이다. 지글본 애니메이션 기법은 평균 137.15(u/sec)이고 Physics 엔진은 58.97(m/sec)이다. 측정된 결과에 대해서 지글본 애니메이션이 Physics 엔진을 사용한 기법보다 월등히 빠른 것을 알 수 있다.

본 논문에서는 세 가지 실험을 통해 성능 평가를 하였지만 실제 게임에서 적용했을 경우에 다양한 애니메이션 처리 및 이펙트 효과가 있어 처리 시간이 달라질 수 있다. 이러한 이유로 지글본 애니메이션에 대한 부분만을 고려하여 실험하였다.

5. 결론 및 향후과제

본 논문에서는 적은 비용을 가지고 세밀한 3D 오브젝트에 대한 이차 애니메이션 생성 알고리즘에 대해 논의하였다. 일반적으로 게임에 있어서 점차 장신구의 애니메이션 요구가 늘어나고 있지만 대부분 물리 엔진을 사용하여 개발하고 있다. 물리 엔진을 기반의 장신구의 개수가 점차 많아질 경우 계산량이 많아져서 실시간적으로 처리하기가 어려워진다. 그래서 본 논문에서는 물리적 요소 중 외부의 힘, 중력, 탄성력만을 고려하였으며 일반 물리적 특성 보다는 애니메이션 연출에 맞추어서 알고리즘을 최적화하였다.

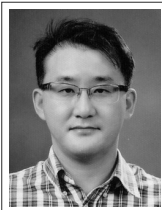
실험 방법으로는 본의 개수에 따른 처리 시간과 오브젝트 개수에 따른 처리 시간을 비교 분석하였고 범용적인 물리엔진을 사용하여 성능실험을 비교하였다. 실험 결과로서 본의 개수가 늘어날수록 계산량이 많아지지만 급격하게 증가하지는 않은 결과를 보였고, 오브젝트의 개수가 많아지더라도 일정한 처리 시간을 보여 실시간적으로 안정된 프레임 수를 유지하는 것을 알 수 있었다.

향후 계획으로는 외부의 힘, 중력, 탄성력 등의

물리적인 요소를 기반으로 한 애니메이션 보다는 좀 더 직관적인 알고리즘을 적용하여 보다 적은 비용으로 2차 애니메이션을 생성할 수 있도록 개선할 것이고 실험 및 분석 방법도 물리 엔진과 비교하여 검증할 계획이다.

참고문헌

- [1] Michael Dickheiser, "GAME PROGRAMING Gems 4", 와우북스, 2005.
- [2] jigglebone,
[http://developer.valvesoftware.com/wiki/\\$jigglebone](http://developer.valvesoftware.com/wiki/$jigglebone)
- [3] Eric Keller, Anthony Honn, "Mastering Maya 2009", SYBEX, 2009
- [4] A Nealen Et. Al, Physically Based Deformable Models in Computer Graphics. EUROGRAPHICS 2005, STAR - State of The Art Repor, pp 071-094.2005
- [5] Anderson Maciel , Tansel Halic , Zhonghua Lu , Luciana P. Nedel , Suvrano De , "Using the PhysX engine for physics-based virtual surgery with force feedback", The International Journal of Medical Robotics and Computer Assisted Surgery, Volume 5 Issue 3, pp 341 - 353
- [6] http://kr.nvidia.com/object/physx_faq_kr.html
- [7] 최민규, "게임에서의 물리기반 시뮬레이션 기술의 현황과 전망", 전자공학회지 제34권 제10호, pp70-79, 2007
- [8] Matthias Müllera, Bruno Heidelbergera, Marcus Hennixa and John Ratcliffb, "Position based dynamics", Journal of Visual Communication and Image Representation Volume 18, Issue 2, pp 109-118, 2007
- [9] Hurbert Nguyen, "GPU Gems 3", Addison Wesley, 2007



박성준 (Sung-Jun Park)

1997년 2월 호서대학교 컴퓨터 공학과 (공학사)
1999년 2월 건국대학교 컴퓨터공학과 (공학석사)
2005년 2월 건국대학교 컴퓨터 공학과 (공학박사)
2006년 3월-현재 호서대학교 게임공학과 조교수
한국콘텐츠학회 정회원
한국게임학회 정회원

관심분야 : 게임, 가상현실, HCI, Bioinformatics



안득용 (Deug-Yong An)

2008년 2월 호서대학교 게임공학(학사)
2008년 3월-현재 호서대학교 게임공학과(석사)

관심분야 : 게임프로그래밍



오성석 (Seong-Suk Oh)

1992년 9월 홍익대학교 미술대학(학사)
1998년 2월 The College of Savannah Art and Design (석사)
2009년 3월-현재 호서대학교 게임공학과 교수
한국콘텐츠학회 정회원
한국게임학회 이사

관심분야 : 게임디자인, 게임심리학, 게임그래픽