

VRML 을 이용한 FBA 3 차원 애니메이션 소프트웨어 설계와 구현*

⁰ 임정연, 성 원, * 박민식, 박종원
충남대학교 정보통신공학과, *한국전자통신연구원

A Design and Implementation of FBA 3-Dimensional Animation Software with VRML

⁰Jeong-Yon Lim, Won Seong, *Min-Sik Park, Jong-Won Park
Dept. of Information and Communications Engineering, Chungnam National University
*Electronics and Telecommunications Research Institute

요 약

최근 인터넷 상의 3D 게임이나 가상 세계 등의 증가로 3 차원 애니메이션에 대한 필요성이 고조되고 있다. 하지만 상당한 메모리와 전송 시간 및 데이터 전송, 처리에 대한 문제가 해결되어야 한다. 이러한 문제점을 해결하기 위해 새롭고 효율적인 3 차원 애니메이션 부호화 기술 개발이 필요하다. 현재, MPEG-4 는 BIFS 와 FBA 에 기반하여 애니메이션 애플리케이션들의 다양한 변형들을 구현할 수 있다. BIFS 는 인터플레이터, 스크립트, 센서 노드들 같은 몇 가지 애니메이션 노드들을 지원한다. 그리고 FBA 는 인간 모형에 흡사한 객체를 생성하고 움직임을 주는데 유용하게 쓰이는 애니메이션 방법이다. 이 방법들 중 본 논문은 FBA 애니메이션을 VRML 2.0 을 사용하여 구현하였고, 이 FBA 객체의 구성과 특징을 관찰하여 FBA 객체의 움직임을 생성하는데 필요한 데이터들을 압축해 데이터 양을 줄이는 알고리즘을 구현하였다.

1. 서론

최근 인터넷의 급성장과 멀티미디어 환경의 대중화 등으로 인해 3 차원 그래픽 정보의 전달방법에 대한 관심이 커지고 있다. 하지만 3 차원 데이터는 간단한 물체라도 상당한 데이터 양을 필요로 하며 실세계에 근접한 물체를 데이터 전송 및 처리하기 위해서는 상당한 대역폭과 컴퓨팅 파워가 필요하다. 그러므로 처리속도를 향상시키기 위해 3 차원 애니메이션 부호화 기술 개발이 필수적이다. 3 차원 애니메이션 부호화 기술을 개발함에 따라 일차적으로 과다한 데이터량으로 인한 메모리 손실을 막을 수 있고, 요즘 급속히 성장해 가는 인터넷을 통해 3 차원 가상 세계를 전송시 시간을 줄일 수 있다. 이렇게 높은 질의 애니메이션에 대한 요구가 계속 증가함에 따라 효율적인 "일반적 3D 모델 애니메이션" 코딩 기법은 MPEG-4 에서도 지원되어야 한다. 현재 MPEG-4 SNHC 소그룹에서의 3D 애니메이션 표준화 작업은 초보적인 시작 단계에 있다. SNHC 소그룹에서는 BIFS 와 FBA 라는 기존의 두 가지 애니메이션 방식에 기반을 두고 연구가 진행중이다.

본 연구는 MPEG-4 애니메이션 표준화 기반 중 하나인 FBA 애니메이션을 이용해 구현된 객체의 각 세그먼트에서 서로 독립적이고 좌우 대칭적인 특징을 관찰하여 아 모델의 데이터량을 줄이는 기법을 적용하였다. [1][2]

2. 3 차원 애니메이션 표준화 동향

MPEG-4 에서의 SNHC 의 3D 모델 코딩과 BIFS 의 애니메이션 코딩은 VRML2.0 에 밀접히 관계되어 있다. 또한, MPEG-4 FBA 는 인간 모델링을 위한 캐릭터 애니메이션을 지원하고 있다. 그러므로 애니메이션 애플리케이션들의 다양한 변형들은 MPEG-4 BIFS 와 FBA 에 기반하여 구현될 수 있다. 현재는 그래픽 애플리케이션들의 개발자나 사용자들은 계속해서 더 높은 질의 애니메이션을 요구하고 있는 반면, 하드웨어, 네트워크, 소프트웨어 기술들의 빠른 발전과 함께, 많은 기업들이 컴퓨터 그래픽 애플리케이션들에 사용될 좀 더 개선된 애니메이션 기술들을 연구하고 있는 실정이다. [3]

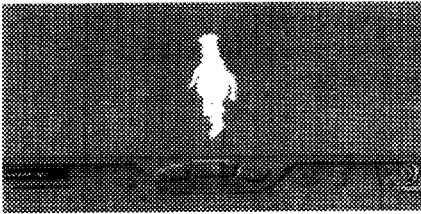
MPEG-4 는 BIFS 와 FBA 에 기반한 3D 애니메이션을 지원한다. BIFS 는 인터플레이터, 스크립트, 센서 같은 몇 가지 애니메이션 노드들을 지원한다. 인터플레이터 노드는 움직임을 표현하는데 유용하지만 많은 데이터를 생성하며 스트리밍을 위해 적합하게 설계되어 있지 않다. 또, MPEG-4 는 각 인터플레이터 노드 자체를 위한 압축 방식을 가지고 있지 않다. 이것은 또한 한 매쉬에서 부분 변화들을 표현하는데 상당히 많은 데이터를 필요로 한다. BIFS 에서는 인터플레이터 노드 데이터 압축과 부분변형을 지원하는 방식이 지원되어야 한다. [4]

VRML 인터플레이터 노드 대신에 사용자는 휴먼 객체에 기반하여 설계된 FBA 를 사용할 수 있다. 이는 휴먼 캐릭터 애니메이션을 위해 개발되었기 때문에 비슷한 다른 객체를 모델링하고 애니메이션하는데 효율적으로 이용될 수 있다. 하지만 인간 모형이 아닌 문어나 유충같은 객체를 모델링하거나, 부분 변형, 데이터의 상당한 양을 고려해야 한다. [5][6]

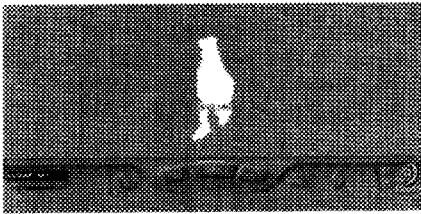
* 본 연구는 1999 년도 한국전자통신연구원의 연구 지원에 의해 수행되었음.

3. VRML을 이용한 FBA 구현

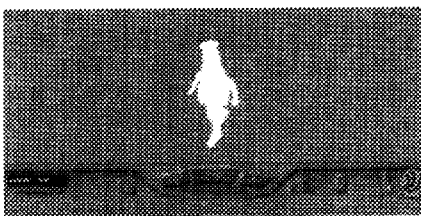
인간모형(humanoid)의 몸체는 팔, 손, 발과 같은 부분(segment)들로 구성되어 있고 이러한 부분들은 각각 손목, 무릎 등의 연결(joint)들에 의해서 서로가 연결되어지게 된다. 그래서, 인간모형을 애니메이션시키기 위해서는 먼저 연결(joint)들을 접근하고 접근한 연결들의 연결 각도(angle)들을 바꾸게 된다. 몸체의 각 부분들은 전형적으로 다각체들의 메쉬 구조로 되어 있고 구성된 메쉬의 정점들의 위치를 변화시킴으로써 움직임을 줄 수 있다. 다음은 인간 모형 모델의 한 예이다. [7]



(a)



(b)



(c)

[그림 1] 인간 모형의 예

3.1. 사용 노드들

가. 연결 (Joint) 노드

몸체에서의 각 연결(joint)은 하나의 연결 노드(joint node)에 의해서 표현된다. 이 연결노드는 각 몸체 세그먼트의 그것의 직계 부모에 대한 관계를 정의하는데 사용되어진다. 하나의 연결은 다른 연결 노드의 자식(child)이 되거나 humanoidRoot Joint의 경우에는 humanoidBody field 안에서의 한 자식만이 될 수 있다. 연결 노드는 또한 다른 연결 정보를 저장하기 위하여 사용될 수 있다. 특히, 하나의 이름이 제공되는데 이것으로 애플리케이션들은 수행시간에 각 연결 노드를 식별할 수 있다. 연결 노드는 인간모형에 움직임을 주기 위한 힌트들을 포함하고 있다. 이러한 힌트들은 최대와 최소의 연결 한계를 포함하고

부가적으로 연결 한계들의 방향 등도 추가할 수 있다. 그러나, 이런 한계들은 인간 모형의 장면 그래프(scene graph) 안에서의 어떤 메커니즘에 의해서 강제로 사용되어지지는 않는다. 이것들은 단지 정보를 주는 목적만으로 제공되어지는 것이다. 이러한 정보들의 사용이나 연결 한계들의 이용 등은 애플리케이션에 전적으로 달려있다.

나. 세그먼트(segment) 노드

인간 모형의 몸체(body)는 팔, 손, 발과 같은 세그먼트들로 구성되어 있다. 각 몸체의 세그먼트는 세그먼트 노드에 저장된다. 세그먼트 노드는 전형적으로 하나의 그룹 노드로서 구현되어질 수도 있다.

다. 인간 모형(humanoid) 노드

인간모형 노드는 연결들, 세그먼트들, 뷰 들에 대한 참조 사항과 작가와 저작권 같은 사람이 읽을 수 있는 데이터를 저장하기 위해서 사용되어진다.

4. 인간모형(humanoid) 데이터 압축

4.1. 인간 모형(humanoid) 모델의 구성

인간모형 데이터는 전체 뼈대를 이루는 데이터 화일과 팔, 다리, 머리, 발, 손 등 각자를 나타내는 세그먼트 데이터로 구성되어 있다. 전체의 뼈대를 이루는 데이터 화일엔 기본적인 양식을 주축으로 하여 이 모델에서 쓰여질 세그먼트와 이 각 세그먼트들이 어느 관절에 연결되는지 그리고 실질적인 동작을 위해 각 파라미터 값을 기술한다. 또 각 관절에서 세그먼트에 이르기까지 한 동작에 대한 전체 뼈대 화일에서 다뤄진 기본 파라미터 값의 변화를 기술하는 인터플레이터 노드가 쓰여진다. 각 세그먼트는 독립적 형태를 갖는다.

4.2. 특징

세그먼트(segment)들에는 skull, l_davicle, l_scapula, l_upperarm, l_forearm, l_hand, l_wrist, r_davicle, r_scapula, r_upperarm, r_forearm 등 이 있는데, 표준 모델의 경우 (정상 모델) 사람의 형태는 좌우 대칭이므로 여기서의 세그먼트 데이터들도 좌우 대칭임을 알 수 있다. 이 좌우 대칭 데이터를 열어보면, 기본적인 파라미터 외에 각 세그먼트의 기본적인 틀을 기술하는 정점, 이 기본 모델에 실제 모양을 덧붙여주는 법선 데이터 그리고 이 각 포인트들을 하나의 삼각형 메쉬로 설명해 주는 정점과 법선 데이터 각각의 연결값(connectivity) 정보로 이루어져 있다. 다음은 인간모형의 예에서 l_forearm, r_forearm의 포인트 데이터이다.

/* Lforearm */		
Point [
0	0.0879881	-9.65221e-10,
3.86088e-09	0.000595087	0.112386,
-0.112386	0.000595087	3.86088e-09,
0.112386	0.000595087	3.86088e-09,
-3.86088e-09	0.000595087	0.112386,
-0.0525249	-0.51803	0.0125878,
-0.00122206	-0.497028	-0.0970431,
-0.00117551	-0.489963	0.1002,
0.0900102	-0.480226	0.00151001
]		

[그림 2] 왼쪽 forearm의 좌표값

/* r_forearm */		
Point [
-7.69216e-09	0.0879881	-9.65221e-10.
-1.3738e-08	0.000595087	0.112386.
0.112386	0.000595097	1.3686e-08.
-0.112386	0.000595077	-1.3686e-08.
1.3634e-08	0.000595087	0.112386.
0.0525249	-0.51803	-0.0125878.
0.00122211	-0.497028	0.0970431.
0.00117554	-0.489963	0.1002.
-0.0900102	-0.480226	0.00151
]		

[그림 3] 오른쪽 forearm 의 좌표값

[그림 2, 3]과 같이, 대칭인 두 세그먼트 파일에서 포인트 값에 대한 인덱스 값이 거의 비슷함을 알 수 있다. 여기서 나열하지 않은 연결 데이터와 법선 데이터 그리고 그 인덱스 값도 좌우에 값이 이와 비슷한 형태를 가지고 있다. 정상 모델의 경우 각 세그먼트의 위치 정보나 트랜스폼 노드에서의 필드 값(크기, 회전, 이동 등)도 거의 같은 값을 갖는다.

4.3. 모델 압축

인간모형 모델은 팔, 다리 같은 대칭 값을 갖는 세그먼트로 이루어져 있다. 이런 특징으로 어느 한쪽 세그먼트와 비교해 다른 한쪽의 세그먼트 데이터의 크기를 줄여나가는 방법을 이용하였다. 대칭인 각 데이터 셋을 보면, 대개 x 축의 값에서 약간의 차이를 보였으며, 이 차이를 보이는 값들 중에 대부분은 같은 절대값을 갖는다는 것을 알 수 있다. 또 y, z 축상의 값은 위 모델에서는 다른 값을 가지는 점은 없었다. 즉, 정상 모델에서의 데이터는 몸의 중앙을 중심으로 대칭해서 봤을 때, 각 깊이(z)나 세로 축(y)에 대한 값은 서로 같지만, 가로 축(x)에 대한 값은 부호가 다를 수 있으며, 각 세그먼트의 모양에 따라서도 x 축의 값이 서로 차이가 생길 수 있다. 이러한 특징을 이용해 먼저 좌우 좌표 값의 크기를 비교해 다른 경우를 고려하고 같은 인덱스 상에서 대칭되는 각 데이터들의 값을 비교해 그 차이가 뚜렷한 값과 부호가 다른 값들을 골라낸다. 대칭되는 값들에 좌표 수를 비교하고, 각 값들에 차이 값을 산출한다. 0 이 아닌 값들을 골라내어 아래의 구조에 인덱스와 차이값을 넣는다. 그리고 각 산출된 값에 따라 부호 값만이 다른 경우를 골라내 오른쪽 세그먼트를 기준으로 왼쪽 세그먼트에 데이터들을 줄여나간다.

5. 결과 데이터

두 데이터 r_forearm, l_forearm 값을 이 알고리즘에 실행시켜 본 결과는 다음과 같다.

```

/* forearm result */
RL = 0
X
6 0.002444
7 0.002351
X_positive
3 8
X_negative
2 5
    
```

[그림 4] 좌우 좌표값 비교 결과

RL 값은 오른쪽과 왼쪽에서 비교한 포인트의 크기가 같다는 것을 의미하고, 그 다음은 x 축에서 6, 7 번째의 데이터 값이 오른쪽에 비해 왼쪽의 값이 이 만큼의 차이 값 가짐을 말한다. 그리고 그 다음은 두 데이터를 비교했을 때 오른쪽 세그먼트 값들에 비해 3, 8 번째 인덱스 값을 가지는 포인트들이 x 축의 값에서만 (+) 부호 값을 가지며, 반대로 2, 5 번째 데이터에서 오른쪽 세그먼트의 2,5 번째 데이터가 (+) 값을 가지는 반면, 왼쪽 세그먼트에서는 (-) 부호 값을 가짐을 의미한다. 나머지 y, z 축 상에 좌우의 세그먼트 값은 같음을 알 수 있다.

6. 결론 및 향후과제

3 차원 모델의 애니메이션이 제대로 이루어지기 위해서는 모델을 저장하고 전송하는데 필요한 공간과 시간의 문제를 해결하는 것이 무엇보다도 중요하다. 이러한 문제점들을 해결하기 위해서 3 차원 애니메이션 부호화 기술이 개발이 필요하다. 3 차원 애니메이션 부호화 기술을 개발함에 따라 과다한 데이터량으로 인한 메모리 손실을 막을 수 있고, 전송 시간을 줄임으로써 최근에 크게 관심이 고조되고 있는 사이버 세계의 모델링을 현실과 근접하게 만들 수 있게 된다.

본 연구는 여러 애니메이션 영역 중에서 인간 모형의 애니메이션을 다룬 FBA 방식을 선택하여 VRML 플 이용하여 부호화 소프트웨어를 구현하였다. 그리고, 인간 모형의 좌우 대칭성을 이용하여 구성 데이터를 압축하는 알고리즘을 개발, 구현하였다.

본 연구의 향후 연구 사항은 크게 두 가지로 요약할 수 있다. 첫째는, 일반적 애니메이션 부호화 방식이 되기 위해서 반드시 필요한 "자유 형태 변형(Free-Form Deformation)"을 어떻게 효율적으로 수행시킬 것인가에 대한 것이다. 두 번째는 좀 더 나은 효율의 데이터 압축률을 이루는 것이다.

7. 참고문헌

- [1] G. Taubin and J. Rossignac, "Geometric Compression Through Topological Surgery", ACM Transactions on Graphics, Vol 17, No. 2, April 1998.
- [2] G.Taubin, W. P. Horm, F. Lazarus and J. Possignac, "Research Report : Geometry Coding and VRML", IBM RC-20925(#92674)
- [3] D.K. McNeil and H.C. Card, "Adaptive Compression of Animated Sequences", IEEE WESCANEX '95 Proceedings
- [4] ISO/IEC JTC1/SC29/WG11, MPEG99/M4519, "Proposal for 3D Dynamic Mesh Coding Requirements", D. K. Kim, et al, March,1999.
- [5] ISO/IEC JTC1/SC29/WG11 N2459, "MPEG-4 Overview", 1998.
- [6] 안치득, "영상 부호화 국제표준화 동향", 한국 통신학회지, 제 14 권 제 9 호, pp.22-26, 1997년 9 월
- [7] A. E. Johnson and M. Hebert, "Control of Polygonal Mesh Resolution for 3-D Computer Vision",April,1997.