

웹환경에서의 실시간 아바타 애니메이션 구현

조용범, 장인영, 이해성, 변혜란
연세대학교 컴퓨터과학과

A Real-Time Avatar Animation in World Wide Web Environment

Yongbum Cho, In-Young Jang, Hae-Sung Lee, Hyeran Byun
Dept. of Computer Science, Yonsei University

요 약

최근 들어 WWW(World Wide Web) 환경을 이용한 각종 Entertainment 산업이 활성화되고 있다. WWW상의 실시간 아바타 애니메이션의 구현은 WWW를 이용한 Entertainment 산업 및 가상현실을 위한 중요한 원천기술 역할을 하리라 기대된다. 특히, 모션캡처를 통해 얻어진 모션데이터로 아바타의 애니메이션을 구현함으로써 보다 자연스러운 동작을 기대할 수 있다. 한편 일반적인 모션데이터는 Euler Angle로 나타나는 데 반하여 VRML 환경하의 아바타에는 Euler Angle를 적용할 수 없다. 이는 VRML의 특성상 현재의 상태(State)를 저장할 수 없기 때문이다. VRML 환경하의 아바타에 적용 가능한 유일한 모션데이터 방식은 특정 축(Axis)과 이 축을 중심으로 한 회전각(Axis Angle)을 이용하는 것이다. 본 연구진은 이를 위하여 일반적인 모션데이터에 나타나는 Euler Angle를 특정 Axis와 Axis Angle로 변환해 주는 이론 및 모션데이터를 실시간으로 공급해 주기 위한 JAVA Streaming Server를 직접 개발하였다. 본 논문에서는 이 시스템의 전체적인 구조를 설명하고자 한다.

1. 서 론

인터넷의 급속한 보급으로 인해 WWW(World Wide Web) 환경은 주요한 개발환경이 되었다. 특히 WWW 환경에서는 정보사용자와 정보공급자간의 상호작용(Interactivity)이 가능하기 때문에, 보다 개인적인 취향과 기호에 따라 원하는 서비스를 제공하는 것이 현재의 추세이다. 현재 WWW에서의 VOD(Video On Demand), MOD(Music On Demand)등이 이에 해당한다. 이 서비스들은 사용자의 요구를 받아서 실시간으로 원하는 콘텐츠를 제공하는 것이다. 그런데, 이 서비스에서 문제점으로 지적되는 것은 콘텐츠의 질, 데이터의 전송대역폭, 그리고 사용자가 쉽게 사용할 수 있는지를 가능하는 사용자 인터페이스이다.

사용자의 요구에 따라 실시간으로 콘텐츠를 제공하는 서비스에서 콘텐츠의 질은 콘텐츠 공급자의 서버용량 및 서버의 데이터 전송대역폭, 사용자측의 수신대역폭과 관련이 있다. 사용자가 낮은 수신대역폭을 가진 경우에는 낮은 질의 콘텐츠를 제공받을 수밖에 없으며, 사용자가 높은 질의 콘텐츠를 얻기 위해서는 사용자는 필수적으로 높은 수신대역폭을 구축해야 하며, 공급자는 대용량 서버 및 대용량의 전송대역폭을 갖추어야 한다. 결국 사용자와 공급자가 모두 높은 지출을 감수하여야만 높은 질의 콘텐츠를 얻을 수 있는 것이다.

한편, 사용자 인터페이스는 사용자의 사용편의성 및 공급자의 공급 편리성과 관련이 있다. 사용자는 쉽게 구할 수 있으며 사용하기 편한 프로그램으로 원하는 서비스를 제공받길 원한다. 그리고, 공급자는 제공하는 콘텐츠의 포맷이 제작 및 공급하기 쉬우며 대중적인 것을 원한다. 그 이유는 공급자의 경우 콘텐츠 제작비용의 낭비를 막을 수 있으며, 사용자의 경우 특정한 포맷으로 제작된 콘텐츠와 플랫폼 의존적인 클라이언트 프로그램으로 서비스 받는 것은 사용자에게 불편을 주어, 단지 일부 전문적인 사용자를 겨냥한 것일 뿐 대다수의 사람에게는 만족을 주지 못하기 때문이다.

본 연구진은 이러한 문제점들을 인식하여 웹환경하의 실시간 아바타 애니메이션을 위하여 아래와 같은 제약조건들을 도출하였다.

첫째, 아바타 애니메이션은 WWW의 HTTP protocol을 Parsing 할 수 있는 웹브라우저 상에서 구동되어야 한다. 이것은 독립적인 클라이언트 프로그램이 아닌, 모든 사용자가 쉽게 사용할 수 있는 웹브라우저를 사용함으로써 대중성을 확보한다. 둘째, 실시간 스트리밍 기술을 사용하며, 공급자가 사용자에게 제공하는 아바타 애니메이션 데이터포맷은 모션데이터를 사용한다. 실시간 스트리밍 기술은 사용자가 공급자로부터 일부의 데이터만을 수신 받더라도 서비스를 이용할

수 있도록 하며, 서비스를 이용하는 동안에 나머지 데이터를 계속 전송 받는 방법이다. 아바타 애니메이션에 모션데이터를 사용하는 것은 최근 급속히 대중화되고 있는 모션캡처 장비를 이용하면 쉽게 데이터를 얻을 수 있기 때문이다. 또한, 모션데이터를 이용한 애니메이션은 적은 량의 데이터로도 훌륭한 움직임을 보일 수 있기 때문이다. 데이터 양의 크기에만 주목한다면 Inverse Kinematics가 모션데이터보다 더 적은 데이터 양으로 아바타의 움직임을 나타낼 수 있다. 그러나, 이 방법의 문제점은 계산량이 매우 크므로 클라이언트의 성능을 떨어뜨리고, 모션데이터 만큼의 자연스러운 애니메이션을 보장하기 어려운 단점이 있다. 이러한 이유로 본 연구에서는 모션데이터를 선택하였다.

그리고, 본 연구진은 모션데이터를 웹환경에서의 실시간 애니메이션 시스템에 적용하기 위해 필요한 Euler Angle의 특정 Axis와 Axis Angle로의 변환이론을 정립하여 시스템에 구현하였다.

2장에서는 VRML의 아바타 표준 모델인 H-Anim과 아바타 애니메이션을 위해 필요한 EAI(External Authoring Interface)에 대해서 설명하고, 3장에서는 모션캡처 데이터에 대해서 설명한다. 또한, 4장에서는 스트리밍 기술들, 5장에서는 본 시스템의 구조 및 실행에 대해서 설명한다. 그리고, 마지막으로 6장에서는 결론 및 향후 연구 방향에 대해 설명한다.

2. H-Anim과 EAI

본 연구에서는 VRML에서 확장된 표준 아바타인 H-Anim[1][2]을 이용하였다. H-Anim은 아바타를 모델링 하기 위한 신체 구조를 기술하고 있다. H-Anim은 골격과 부위에 따라 해부학적으로 구성되어 있는데, 이것은 그림 1을 보면 알 수 있다.

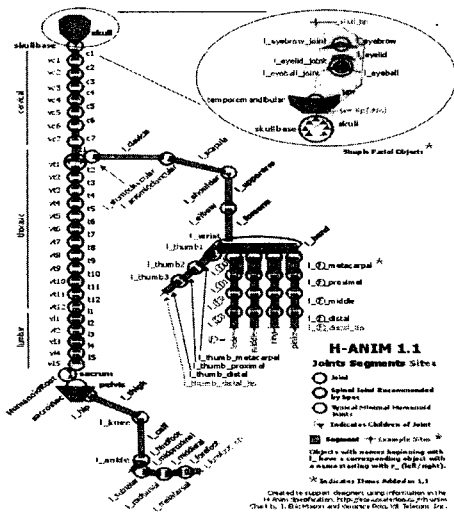


그림 1. H-Anim Specification

H-Anim은 크게 5가지의 프로토타입이 정의되어 있다. 이 5가지의 프로토타입은 JOINT Proto, SEGMENT Proto, SITE Proto, HUMANOID Proto, DISPENSER Proto 등이다. 각 프로

토타입에 대한 자세한 설명은 [HTTP://ece.waterloo.ca/~h-anim/](http://ece.waterloo.ca/~h-anim/)에서 볼 수 있다.

EAI(External Authoring Interface)는 VRML로 이루어진 가상 공간을 조작하기 위한 기술이다. 그리고, EAI는 외부 프로그래밍과 인터페이스를 정의한 것이므로 이론상 모든 프로그래밍 언어를 지원한다. 본 논문에서는 시스템을 특정 환경에 관계없이 구동하기 위하여 JAVA를 사용하였으며, 아바타의 각 관절 회전각을 변화시키기 위해서 EAI를 사용하였다.

3. 모션캡처 데이터

인체의 움직임을 표현하기 위한 여러 시도 중에서 사람의 움직임을 가장 잘 표현한 것이 모션캡처 방법이다. 모션캡처는 실제 사람의 움직임을 센서를 통해 감지하여 이를 데이터로 표현한 것이다. 여기서 모션캡처 데이터는 위치와 변환각도로 표현된다. 그리고, 모션캡처 데이터는 센서를 통해서 들어온 데이터를 처리하는 방법에 따라 여러 포맷으로 나뉜다. BioVision사의 BVH, MotionAnalysis사의 HTR, Acclaim사의 ASF/AMC 등이 있다. 본 연구에서는 BioVision사의 BVH를 사용하였다. BVH의 구조는 HIERARCHY와 MOTION 섹션으로 나뉘어져 있다. HIERARCHY 섹션은 골격을 정의하는 부분이다. 골격은 총 18개의 JOINT로 이루어져 있으며 Hips를 몸체의 Root로 하여 왼쪽 하체와 오른쪽 하체 그리고 상체방향으로 일련의 순서를 가지고 각각의 segment를 JOINT로 연결한 방식으로 구성되어 있다. MOTION 섹션은 각 JOINT에 적용되는 Euler Angle값으로 구성되어 있다.

4. 스트리밍

스트리밍은 최근에 인터넷 방송에서 많이 쓰이는 기술로서, 서버에서 클라이언트로 자료를 모두 보낸 후에야 클라이언트가 받은 자료를 실행하는 것이 아니라, 서버에서 일부분의 데이터만이라도 클라이언트로 전송되면 이를 바로 사용자에게 제공함으로써 실시간 play를 가능하도록 한다.

본 연구에서 구현된 스트리밍은 원시데이터를 서버가 클라이언트로 전하는 형태를 취한다. 서버에 저장되어 있는 데이터는 압축이 전혀 안된 ASCII형태의 모션캡처 데이터이고, 클라이언트는 이것을 네트워크를 통해 전송 받는다. 압축을 고려하지 않은 것은 원시데이터 형태에도 대역폭에 상관없이 일정한 질을 유지 할 수 있기 때문이다. 따라서, 현재는 이것에 대해서 고려를 하지 않았다. 또한 스트리밍의 중요한 특징인 손실에 대한 균전함이 보장되어야 하는데, 이 부분도 현재까지는 적용되지 않았다.

본 연구의 스트리밍은 Queue를 이용하여 버퍼를 만들고 여기에 서버가 보내온 데이터를 저장한다. 버퍼가 일정수준 차게되면 버퍼에서 데이터를 받아 이를 아바타 애니메이션에 적용한다. 그리고, 클라이언트는 버퍼를 항상 감시하여 서버에서 더 이상 보낼 데이터가 없을 때까지 버퍼를 채우도록 한다.

본 연구에서는 스트리밍을 JAVA로 구현하였다. JAVA는

웹환경에 적합하며, 플랫폼에 영향을 받지 않고, 또한 사용자가 쉽게 사용할 수 있기 때문이다.

5. 시스템 구조 및 실행

웹환경에서의 실시간 아바타 애니메이션을 위한 시스템은 크게 서버부분과 클라이언트 부분으로 이루어진다. 그림 2에서 서버부분은 모션 DB와 DB를 액세스하여 데이터를 보내는 서버로 이루어져 있다.

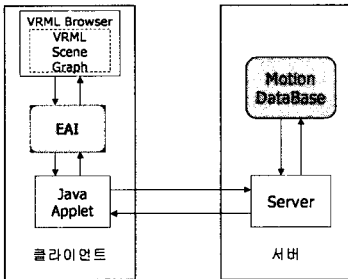


그림 2. 시스템 구조도

서버는 모션 DB와 통신하여 모션 데이터를 얻어내며, 이렇게 얻어낸 데이터를 클라이언트에 전송한다.

클라이언트에서는 VRML 브라우저를 JAVA 애플릿이 EAI를 통해서 컨트롤한다. JAVA 애플릿은 서버에서 전송된 모션 데이터를 EAI에 선언된 JOINT에 전달한다. 그리고, JOINT에 전달된 값들은 VRML 브라우저에 전달되어 H-Anim 아바타의 애니메이션을 만든다.

여기서 중요한 문제가 발생하는데, 그 문제는 VRML과 같은 Mark-up Language의 특성에 기인한다. 즉 브라우저는 단지 HTML, VRML과 같은 Mark-up 언어를 Parsing하여 이를 렌더링 할 뿐 VRML과 HTML에서 일어나는 작업들의 상태(State)를 기억하지는 못하기 때문이다. 한편 모션데이터가 제공하는 각 JOINT값들의 변화는 X축, Y축, Z축의 Euler Angle로 나타내어지는데, 이 값을 순서대로 적용하려면 각각의 축에 대해서 각각의 변환상태를 기억하여야만 한다. 그러나 VRML에서는 상태를 기억하지 못하기 때문에 각각의 축에 따라 적용되는 3번의 각 변환을 한번의 변환으로 해내야만 한다. 즉 Euler Angle을 특정 Axis를 기준으로 한 Angular Displacement로 변환하여야만 한다. 기존의 접근 방법들은 주로 quaternions[4]을 이용하였는데, 일반적으로 상당한 수준의 수학적식이 요구되며, 또한 singularity 처리가 항상 문제가 되었다. 그래서, 본 연구진은 전혀 새로운 접근 방법을 이용한 변환공식을 유도하였다. 이 변환공식은 지면의 제약으로 인해, 차후에 전문학술지에 발표할 예정이다.

그림 3은 모션데이터를 아바타에 적용한 결과화면이다. 그림 3에서 아래 부분의 애플릿은 JAVA AWT를 이용하여 프로그램 하였고, 웹브라우저는 Internet Explorer 5.0을 사용하였으며, VRML 플러그 인으로는 SGI의 Cosmo Player 2.0을 사용하였다.

애플릿의 버튼은 Play, Go, Enter로 이루어져 있으며, 리스트 박스는 모션데이터를 선택하도록 하였으며, Play버튼은 스트리밍을 시작하여 일정수준 버퍼가 데이터로 채워지면 동작을 하게끔 하였다. Go버튼은 버퍼에 적당량의 데이터가 채워져 있지 않더라도 H-Anim 아바타를 구동시키도록 하는 버튼이다. Enter버튼은 원하는 Frame을 입력받아 그 Frame의 동작을 생성시킨다. 그림 4는 3D Studio Max 3.x의 Character Studio에서 같은 동작을 Biped에 적용시켜서 실행시킨 결과이다.



그림 3. 실행화면

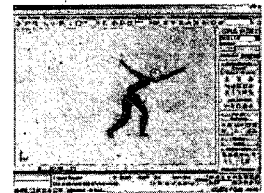


그림 4. 3DS Max 캡처

6. 결론 및 향후 연구 방향

웹환경에서 실시간 아바타 애니메이션은 모션캡처 데이터를 이용하여 작동되었다. 그러나, 문제점이 발견되었는데 EAI와 VRML 브라우저간의 통신 속도가 느리다는 것이다. JAVA 애플릿에서 받은 데이터를 처리하는 과정은 매우 빠르게 진행되지만 실제로 화면에 구현될 때는 Mark-up 언어의 느린 속도로 인해 부드러운 화면을 생성시키기 어려웠다. 이는 Frame Skipping이나 클라이언트 컴퓨터의 성능을 높임으로서 해결할 수 있다.

현재는 한 명의 아바타만을 움직이고 있으나, 향후의 연구에서는 배경의 삼입과 복수의 아바타를 움직이도록 할 것이며, 배경과 아바타, 그리고 아바타와 아바타 사이의 위치정보에 따라서 일어나는 문제들을 해결하여 멋진 애니메이션을 보일 수 있도록 발전시킬 것이다. 또한, 현재 지원하는 포맷이 BVH만 가능하므로, 다른 포맷도 수용할 수 있도록 개발할 것이다. 그리고, 마지막으로 아바타의 골격 크기 변화에 따른 동작의 부자연스러움을 해결하기 위해서 Motion Retargetting[5] 기술을 적용할 예정이다.

참고문헌

- [1] <http://ece.uwaterloo.ca/~h-anim/>, Humanoid Animation Working Group
- [2] <http://www.web3d.org>, Web 3D Consortium
- [3] J. Arfken, Academic Press, 1998, Mathematical Methods for Physicist
- [4] John C. Hart, George K. Francis, Louis H. Kaufmann, Visualizing Quaternion Rotation, ACM Trans. Graph. 1994
- [5] Michael Gleicher. Motion Editing with Space-Time Constraint. Proceedings of the 1997 symposium on Interactive 3D graphics, 1997