

아바타 애니메이션을 위한 행위 기술 언어의 정의 및 활용

김택수⁰, 최윤철
연세대학교 컴퓨터과학과

Definition and Application of The Behavior Description Language for Avatar Animation

Taek-Soo Kim⁰, Yoon-Chul Choy
Dept. of Computer Science, Yonsei Univ.

요 약

가상환경에서 애니메이션을 구현하는 방법으로는 여러 가지가 있다. 그러나 그 방법들을 살펴보면 애니메이션 데이터와 아바타 데이터가 서로 분리되지 못하고, 애니메이션 데이터는 아바타 데이터에 종속되어 있다는 단점이 있다. 이에 본 연구에서는 애니메이션 데이터를 아바타 데이터에 독립적으로 기술할 수 있도록 새로운 행위 기술 언어를 정의하고 이를 활용할 수 있도록 하는 시스템을 개발, 제시하도록 한다.

1. 서론

가상현실은 복잡한 3 차원 데이터를 신속하게 처리할 수 있는 대용량의 컴퓨터를 요구하는 분야이며, 현실감을 높이기 위해 몸에 장착하는 고가의 장비를 구비해야 경험할 수 있는 학문이다. 그러나 현재 구축되어 있는 인터넷 환경에서 고가의 장비 없이도 참여자가 가상현실을 즐길 수 있도록 하는 제한된 범위에서의 가상현실, 즉 비몰입형 가상현실 또는 데스크탑 가상현실이라고 불리는 시스템도 최근 각광받고 있다. 이러한 예로 대표적인 것이 VRML(Virtual Reality Modeling Language)을 이용한 가상현실인데, VRML은 복잡한 3 차원 그래픽을 컴퓨터 프로그

래밍 언어와 같은 형식으로 작성할 수 있게 하여 데이터의 크기를 줄임으로써 네트워크의 전송 속도를 향상시켰으며, 3 차원 그래픽을 표현할 수 있는 그래픽 카드를 가지고 있다면 누구나 가상 세계를 경험해 볼 수 있다.

가상현실에서 가장 중요한 연구 분야 중의 하나가 바로 아바타의 애니메이션이다. VRML에서는 애니메이션을 구현하기 위해 자체적으로 몇 가지 노드를 지원하고는 있지만, 약간의 문제점을 가지고 있어 사용하기가 불편하고 애니메이션의 구현이 어렵다. 이에, 본 연구에서는 VRML에서의 애니메이션의 한계를 극복하기 위해 새로운 행위 기술 언어를 제안하고, 가상환경

시스템을 구축할 때 이 행위 기술 언어를 활용할 수 있도록 지원하는 어플리케이션을 개발하여 데스크탑 가상환경 시스템에서 손쉽게 활용할 수 있도록 하고자 한다.

2. 관련연구

2.1 VRML에서의 애니메이션

VRML에서 제공하는 기본 기능을 이용해 애니메이션을 구현하였을 경우 다음과 같은 문제가 발생한다. 첫째, 아바타 데이터와 애니메이션 데이터의 분리가 불가능하다. 즉, 데이터를 서로 ROUTE 구문으로 연결시켜야 하기 때문에 애니메이션 데이터를 따로 저장할 수 없다. 둘째, 애니메이션 데이터는 아바타 데이터에 종속적이다. 따라서 아바타 데이터에 수정이 가해졌을 경우 애니메이션 데이터를 같이 수정해야 하며, 수정하지 않으면 애니메이션이 실행되지 않는다.

아바타 애니메이션을 필요로 하는 많은 연구에서는 위와 같은 VRML의 단점을 극복하기 위해 다른 방법을 시도하고 있는데, 크게 확장 노드를 사용하는 방법과 시스템 모듈에서 애니메이션을 담당하는 방법의 두 가지로 나눌 수 있다.

2.2 확장 노드를 사용하는 애니메이션

EPFL(Swiss Federal Institute of Technology)의 LIG(Computer Graphics Lab.)에서는 애니메이션을 위한 PROTO 노드를 자체적으로 정의하여 사용한다[1]. 이 PROTO 노드는 애니메이션 데이터와 아바타 데이터를 연결하는 ROUTE 문을 VRML 스크립트를 이용하여 동적으로 생성시킨다. 따라서, 애니메이션 데이터를 많이 수정하지 않고도 여러 아바타 데이터에 쉽게 적용시킬 수 있다는 장점이 있다. 또한, 애니메이션 PROTO 노드는 EXTERNPROTO 노드를 사용하여 정의되었기 때문에 애니메이션 데이터를 별도의 파일로 분리, 저장할 수 있다.

그러나, 아바타 데이터에는 EXTERNPROTO 노드 선언 부분과 실제 노드 설정 부분을 추가 해주어야 하며, 애니메이션 데이터 부분에도 아바타에서 사용하는 조인트 노드의 목록을 작성 해주어야 한다. 따라서 이 방법 역시 아바타 데이터와 애니메이션 데이터의 분리가 불가능하며 서로에게 종속적이다.

2.3 별도의 모듈에서 제어하는 애니메이션

Hewlett-Packard Lab.에서 개발한 LivingSpace는 VRML 환경에서 다중 사용자를 지원하는 가상환경 시스템이며, Living Worlds 표준에 맞게 제작되었다. LivingSpace는 자바(Java)와 EAI(External Authoring Interface)를 이용하여 개발, 웹 브라우저와 VRML 브라우저 플러그인만 있으면 사용할 수 있도록 제작되었다 [2].

LivingSpace에서는 역운동학(Inverse Kinematics)을 이용하여 아바타 애니메이션을 구현했으며, 애니메이션은 VRML에서 다루지 않고 별도의 애니메이션 모듈에서 구현하고 있다.

이렇게 외부 프로그램에서 애니메이션을 담당할 경우, 애니메이션을 수정, 추가하기 위해서는 프로그램 자체를 수정해야 하기 때문에 일반 사용자가 애니메이션을 확장할 수 없으며, 역운동학 애니메이션을 사용하므로 프로그램에서 지원하는 동작 외에 새로운 동작을 만들어 내는 것은 쉽지 않고 키프레임 애니메이션에 비해 애니메이션의 질이 떨어진다는 단점이 있다.

3. 행위 기술 언어의 정의

앞에서 살펴본 여러 가지 문제점들을 해결하기 위하여 본 논문에서는 애니메이션 데이터를 정의할 때 사용하는 새로운 행위 기술 언어를 제안하고자 한다.

본 논문에서 제안하는 행위 기술 언어는 XML(eXtensible Markup Language)을 기반으로

하고 있는데, 이러한 구조를 제안하는 이유는 다음과 같다.

첫째, XML은 데이터를 다루기 쉽다. XML은 여러 프로그래밍 언어에서 지원하고 있어 개발이 용이하고 DOM(Document Object Model)을 제공하여 문서 구조 분석 및 데이터 처리가 쉽게 이루어진다.

둘째, XML은 저장 및 검색이 용이하다. 이미 XML 전용의 상용 데이터베이스가 출시되어 있으며 XML을 저장하고 검색하는 것은 연구 단계를 거쳐 실용화 단계에 접어들지 오래이다.

셋째, 확장성이 뛰어나다. XML은 확장성을 위해 설계된 구조를 가지고 있어 확장성이 뛰어나므로 XML 기반으로 설계된 언어는 쉽게 수정하거나 확장할 수 있다.

본 연구에서는 XML을 기반으로 한 새로운 행위 기술 언어를 구상하였으며, 그 DTD(Document Type Definition)를 정의하였다. 이 행위 기술 언어를 사용하면 키프레임 애니메이션과 역운동학(Inverse Kinematics) 애니메이션을 기술, 사용할 수 있으며 미리 정의된 애니메이션 데이터를 재정의할 필요 없이 쉽게 재사용할 수 있다.

다음은 본 논문에서 정의한 행위 기술 언어의 DTD의 일부이다.

```

<!ELEMENT xanim (low+ | mid+ | ik+)>
<!ELEMENT low (describe?, joint+)>
<!ELEMENT joint (position+ | rotation+)>
...
<!ATTLIST joint
name CDATA #REQUIRED
>
...
<!ELEMENT rotation (#PCDATA)>
<!ATTLIST rotation
key CDATA #REQUIRED
    
```

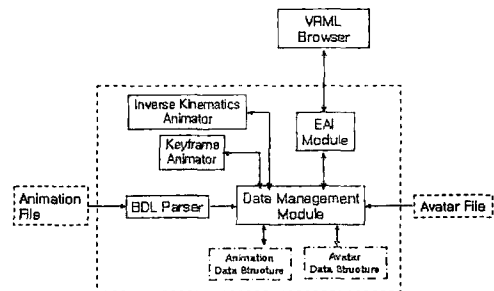
```

x CDATA #REQUIRED
y CDATA #REQUIRED
z CDATA #REQUIRED
>
...
    
```

[표 1] 행위 기술 언어의 DTD (일부)

4. 기반 시스템의 설계 및 구현

본 논문에서 제안한 행위 기술 언어는 그 자체로는 사용이 불가능하며 이를 뒷받침하는 시스템이 필요하다. 따라서 본 논문에서는 행위 기술 언어를 활용할 수 있도록 도와주는 시스템을 설계, 구현하여 제시하고자 한다. 본 시스템의 구조를 도식화하면 다음과 같다.



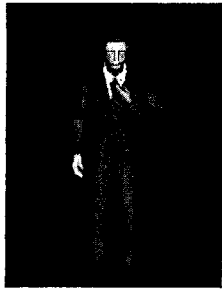
[그림 1] 시스템 다이어그램

[그림 1]에서 보는 바와 같이, 우선 행위 기술 언어를 해석할 수 있는 해석기가 필요하고, 애니메이션 데이터 및 아바타 데이터를 관리하는 데이터 관리 모듈이 필요하며, 애니메이션 모듈 및 EAI 모듈 등이 필요하다.

위 시스템 모듈 중에서 키프레임 애니메이션 모듈을 위해 사원수(Quaternion)[3]를 사용하였으며, 역운동학 애니메이션 모듈을 위해서는 UPENN의 Center for Human Modeling and Simulation에서 개발한 IKAN (Inverse Kinematics using Analytical methods) 라이브러리를 사용하여 개발하였다[4].

5. 애니메이션의 실행 결과

우선 아바타의 구조와 상관없이 애니메이션의 실행이 가능하다는 것을 확인하기 위해 하나의 데이터를 작성하고 그 데이터를 구조가 다른 두 휴머노이드에 적용시켜 보았다. [그림 2]에서 보이는 휴머노이드는 21 개의 조인트 노드를, [그림 3]에서 보이는 휴머노이드는 77 개의 조인트 노드를 가지고 있다. 행위 기술 언어를 사용하면 데이터의 수정 없이도 비슷한 수준의 애니메이션을 보여줄 수 있음을 알 수 있다.

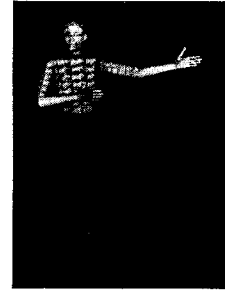


[그림 2] 일반적인 휴머노이드의 애니메이션



[그림 3] 정교한 휴머노이드의 애니메이션

또한, 기존에 정의한 애니메이션을 불러와서 재사용하는 것이 가능함을 알 수 있다. [그림 4]에서 볼 수 있는 애니메이션은 허리, 왼팔, 오른팔의 데이터를 따로 저장한 다음 각각의 데이터를 따로 불러와서 함께 실행시킨 모습이다.



[그림 4] 기존의 데이터를 재사용한 애니메이션

6. 결론

본 논문에서는 아바타 데이터와 독립적인 아바타 행위 기술 언어를 제안하였고 이 행위 기술 언어를 활용하기 위한 시스템을 설계, 구현하였다. 본 논문에서 제안하는 행위 기술 언어를 사용, 애니메이션을 정의하면 아바타의 구조에 관계없이 애니메이션의 적용이 가능하며, 기존에 작성한 데이터를 재활용할 수 있고, 역운동학을 이용한 애니메이션이 가능하다. 또한 애니메이션 데이터가 아바타 데이터와는 독립적으로 존재하고 또 종속적이지 않으므로 애니메이션 데이터베이스의 구축 등 다양한 활용이 가능하다.

7. 참고문헌

- [1] Christian Babski, "Animation Prototype", <http://ligwww.epfl.ch/~babski/StandardBody/Animation.html>
- [2] Mike Wray, Vincent Belrose, "Avatars in LivingSpace", Proceedings VRML 99 of the fourth symposium on The virtual reality modeling language, 1999, p.13-19
- [3] Nick Bobick, "Rotating Objects Using Quaternions", July 3, 1998, http://www.gamasutra.com/features/19980703/quaternions_01.htm
- [4] UPENN, Center for Human Modeling and Simulation, "IKAN Programmer's Manual"