

작업레벨의 아바타 모션 모델링 및 생성을 위한 모션 편집기 구현

김가영*, 김재경*, 임순범**, 최윤철*
* 연세대 컴퓨터과학과
** 숙명여자대학교 멀티미디어학과

Implementation of Motion Editor for Task-level Avatar Motion Modeling and Creation

Ka-Young Kim*, Jae-Kyung Kim*, Soon-Bum Lim**, Yoon-Chul Choy*
* Dept. of Computer Science, Yonsei University
** Dept. of Multimedia, Sookmyung Women's University

요 약

최근 3D 그래픽을 활용한 다양한 서비스들이 많이 개발되어 제공되고 있다. 특히 가상환경 내에서 아바타를 이용하여 콘텐츠 정보를 제공하는 서비스는 사용자의 흥미를 유발하기에 효과적이며, 정보 전달력 면에서도 뛰어나다는 장점으로 인해 최근 가장 각광받는 분야이다. 이러한 3D 아바타의 모션을 생성하기 위한 다양한 편집기 시스템이 제안되어 사용되고 있지만, 현재 시스템에서 생성하는 아바타 모션은 표준 형식이 아닌 어플리케이션에 비 독립적 방식으로 저장되기 때문에 모션의 재사용성 측면이나 활용성에서 비효율적이다. 그러므로 이와 같은 문제점을 해결하기 위해 사용자가 직접 아바타를 조작하여 작업레벨 모션을 정의할 수 있으며, 생성된 모션을 표준형식인 XML 파일로 변환하여 저장해주는 시스템을 제안하고자 한다.

1. 서론

3D 그래픽 기술의 발달과 그래픽 가속기 성능의 향상으로 인해 최근 가상환경 내에서 아바타를 이용하여 콘텐츠를 전달하는 서비스들이 많이 개발되어 제공되고 있다. 아바타를 이용하여 콘텐츠를 전달하는 서비스의 경우 시각적인 흥미 유발이 쉽다는 측면과 동시에 정보 전달력 면에서도 매우 효율적이다.

이러한 3D 아바타를 이용한 서비스를 보다 효과적으로 제공하기 위해서 아바타의 모션을 생성하고 제어하기 위한 효율적인 시스템의 제안이 필수적이다. 현재 아바타 모션 생성 및 제어에 대한 다양한 연구가 진행되고 있으며, 프로그램에 대한 사전지식이 없는 사용자들도 쉽게 아바타 모션을 생성하고 제어할 수 있는 편집기 시스템이 개발되어 사용되고 있다.

그러나 대부분의 시스템이 인터넷 방송과 같은 특정분야에 제한된 모션만을 제공하거나 사용자가 직접 필요한 동작을 생성하여 사용할 수 없다는 문제점을 가지고 있다. 또한 기존 편집기 시스템을 통해서 생성된 아바타 모션은 표준 방식으로 저장되지 못하고 시스템에서 제안한 방식에 따라 저장되므로, 모션의 관리 및 재사용성 측면에서 어려움이 많았다.

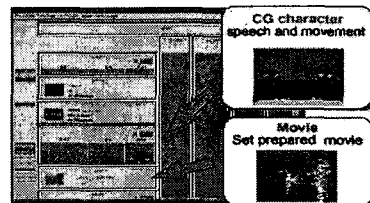
이러한 문제점을 해결하고 사용자가 원하는 모션을 직접 아바타를 조작, 제어함으로써 생성할 수 있는 아바타 모션

편집기 시스템을 제안하고자 한다. 편집기 시스템을 통해서 생성된 모션은 표준 XML 파일로 자동 저장되며, 사용자는 간단한 파라미터 조작만으로 새로운 모션의 재정의가 가능하다. 그리고 생성된 하나의 모션들을 통합하여 새로운 작업레벨 모션의 정의가 실시간으로 가능하다.

2. 관련연구

2.1 TVML(TV Program Markup Language)

일본 NHK 에서 웹 기반 TV 프로그램을 데스크 탑 환경에서 [1] 제작하기 위해 자체적으로 개발한 언어인 TVML을 조작할 수 있는 편집기를 개발하여 사용하고 있다. TVML 편집기 [2]를 통해서 사용자는 방송에 등장하는 아바타의 동작을 생성할 수 있으며 사용자에게 의해서 정의된 행동들은 정해진 시나리오에 따라 화면에 보이게 된다.



[그림1] TVML 편집기를 이용한 아바타 동작 정의

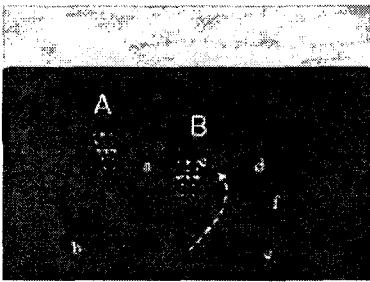
본 논문은 2003년도 산업자원부의 신기술실용화개발사업 지원에 의하여 연구되었음

하지만 TVML 및 편집기의 경우 인터넷 방송이라는 특수한 상황에서 사용될 수 있는 제한된 동작만을 지원하고 있다. 또한 XML과 같은 표준적인 형식이 아닌 자체 정의의 언어를 사용하고 있기 때문에 확장성 및 재사용성이 떨어져 범용적인 활용 면에서 제한을 가지고 있다. 다음 [그림1]은 TVML 편집기를 이용하여 아바타의 동작을 정의하는 예이다.

2.2 Dynamic 모션 편집기

Dynamic 모션 편집기[3]라는 툴을 사용해서 사용자는 자신이 원하는 동작을 쉽게 정의할 수 있다. 아래 그림과 같은 사용자 인터페이스를 제공하여 사용자는 자신이 움직이길 원하는 아바타의 부분을 선택하여 마우스 드래그를 통해서 움직임을 줄 수 있다. 사용자의 마우스 드래그 이벤트가 발생할 때마다 시스템은 자동적으로 모션 데이터 정보틀 Knowledge-Base에 저장하게 된다. 이러한 과정을 통해서 하나의 모션이 정의된다.

Knowledge-Base내의 모션데이터는 사용자가 정의한 특정 알고리즘에 따라 아바타에 적용이 가능하다. 사용자는 Visual Script를 통해서 아바타가 움직이기 위한 경로만을 지정하고, 나머지 아바타 모션을 제어하는 부분은 Knowledge-Base와 아바타 사이에 필요한 동작을 찾아서 매핑시켜 주는 알고리즘에 따라 자동 수행된다.



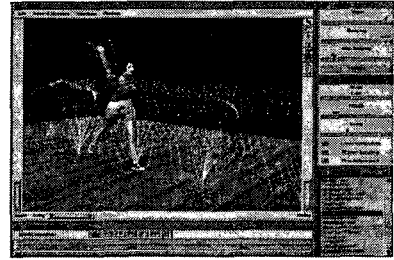
[그림2] Visual Script 실행모습

위의 그림[2]와 같이 Visual Script를 사용하여 사용자는 원하는 아바타를 선택하여 원하는 목적지까지 경로를 지정해 줄 수 있다. 이 때 목적지 까지 도달하는 과정 중에 아바타가 수행하게 되는 모션들은 알고리즘에 의해 미리 정의된 모션들을 보여주며, Script 코드가 실행과정을 통해서 자동 생성된다.

이와 같은 시스템은 저장된 Script가 시스템에 독립적이지 못하다는 단점을 지닌다. 그래서 새로운 시스템에 Script를 활용하기가 불편하며, XML 과 같은 표준형식의 Script가 아니므로 재사용성 측면이나 확장성 측면에서 비효율적이다.

2.3 모션클립을 이용한 편집기

모션 캡처는 캐릭터 애니메이션에서 가장 효율적인 기술 중 하나이다. 사람의 움직임과 가장 유사한 동작을 추출하기 위한 광학적인 또는 자기학적인 모션 캡처 시스템[4]을 사용하여 사용자가 원하는 동작을 생성한다. 이렇게 생성된 모션들은 클립 형태로 모션 라이브러리[5]에 저장되어 필요한 동작을 재사용할 수 있도록 한다.



[그림3] 모션 캡처를 이용한 아바타 편집기

각 모션 클립들은 서로 다른 캐릭터에 적용 가능하도록 호환성이 유지되며, 두 개 이상의 클립을 묶어서 하나의 동작을 완성시킬 수 있다. 이러한 기능을 구현하기 위해서 각 모션 클립을 다루는 편집기들이 많이 구현되고 있다.

모션 캡처를 사용한 캐릭터 애니메이션의 구현은 사람의 동작을 직접 캡처하여 사용하기 때문에 가장 사실적이고 유연한 동작을 얻을 수 있다는 장점을 가지고 있지만 모션 캡처를 수행하기 위해 필요한 장비자체가 고가의 장비이므로 일반 사용자들이 쉽게 사용하기엔 어려움이 많다. 또한 구현된 모션클립을 라이브러리로 저장하여 사용한다 해도 사용자가 필요로 하는 동작들을 모두 충족시키기엔 불가능하다는 단점을 가진다[6].

3. 아바타 모델 정의 및 모션 기술언어

3.1 아바타 모델의 정의

앞 절에서 언급한 H-Anim 인체모형은 아바타 모형을 구현하는데 기본적인 제안 조건이며 이 규정을 반드시 따라야 한다는 요구사항은 아니다. 이러한 내용은 H-Anim의 제안서에도 명시되어있다. 그러므로 자신이 구현할 환경과 목적에 적합한 아바타 모델을 구현하는 과정에서 H-Anim에서 제안한 사항들 중 필요한 내용을 취합 선택하는 것이 바람직하다.

본 연구에서는 사이버교육이나 데이터방송 등과 같이 특정 도메인 환경에서 요구되는 작업 수준의 모션을 표현 및 생성하기 위하여 아바타 작업레벨 모션 표현 모델링을 하였다. 이를 위해 다음 정의1부터 정의4를 제안하였다.

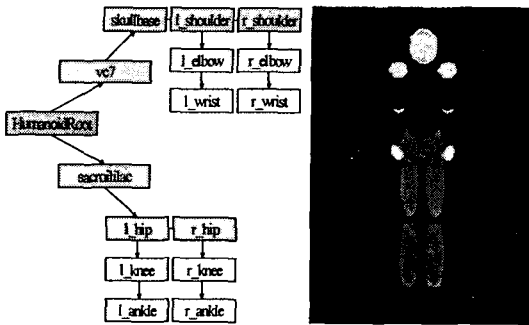
[정의 1] 아바타의 작업레벨 모션(TB)은 아바타의 신체구조(AH), 각 구조에 대한 자유도(DOF)와 이에 따른 모션(M_i) 혹은 하나 이상의 다른 작업레벨 모션들(TB_i)의 조합으로 이루어진다.

$$TB = \{AH, DOF, \{M_i \sum TB_i\}\}$$

[정의 2] 아바타의 신체구조(AH)는 아바타 신체 표준인 H-Anim의 사양을 준수하여 조인트(J_i)와 세그먼트(S_i)의 16개 쌍으로 이루어진다.

$$AH = \sum (J_i, S_i)$$

다음 그림[4]의 왼쪽은 16개의 조인트로 이루어진 아바타의 모델을 도식화한 것이며, 오른쪽은 OpenGL을 사용하여 구현한 3D 아바타 모습이다.



[그림 4] 16개의 조인트의 구조와 OpenGL로 구현한 3D 아바타 모델

조인트는 최상위 노드인 HumanoidRoot로 시작하여 다음과 같이 계층적인 구조로 이루어진다.

```
J = HumanoidRoot
HumanoidRoot = (sacroiliac, vc7)
sacroiliac = (l_hip(l_knee(l_ankle)), r_hip(r_knee(r_ankle)))
vc7 = (skullbase, l_shoulder(l_elbow(l_wrist)), r_shoulder(r_elbow(r_wrist)))
```

```
세그먼트 역시 조인트와 같은 구조로 이루어지며 sacrum을 부모노드로 하여 같은 구조의 조인트와 쌍을 이룬다.
S = sacrum
sacrum = (pelvis, c7)
pelvis = (l_thigh(l_calf(l_hindfoot)), r_thigh(r_calf(r_hindfoot)))
c7 = (skull, l_upperarm(l_forearm(l_hand)), r_upperarm(r_forearm(r_hand)))
```

- [정의 3] 자유도(DOF)는 1부터 3까지 각 관절이 가질 수 있는 도수에 따라 분류하였다.
- 3 DOF = {HumanoidRoot, sacroiliac, l_hip, r_hip, l_shoulder, l_wrist, r_shoulder, r_wrist, skullbase}
 - 2 DOF = {Ankle}
 - 1 DOF = {l_knee, r_knee, l_elbow, r_elbow}

즉, 어깨(shoulder)나 팔목(wrist) 등은 x, y, z축 모든 방향으로 회전이 가능하며 팔꿈치(knee) 등은 한 평면에서만 회전이 가능하다. 총 자유도는 33으로 자연스러운 모션보다는 어떤 의미 있는 행동을 하기에 충분한 수치이다.

[정의 4] 모션(M)은 임의의 시간 t에서 각 관절이 가지는 DOF내의 회전운동(Rotation)으로 이루어지며 아바타 신체 구조 전체는 회전 외에 이동도 가능하다.

M = {Rotation(J, DOF), Translation(AH)} at each time t.

3.2 모션 기술언어의 정의

앞에서 살펴본 문제점들을 해결하기 위해 본 논문에서 제안하는 방식은 XML(eXtensible Markup Language)을 기반으로 하는 아바타 모션 기술언어를 정의하는 것이다. 이러한 구조를 제안하는 이유는 다음과 같다[7].

첫째, XML 데이터는 다루기가 쉽다는 점이다. XML은 여러 프로그램 언어에서 지원하고 있어 개발이 용이하고 DOM을

제공하여 문서 구조 분석 및 데이터 처리가 쉽다.

둘째, XML은 저장 및 검색이 용이하다. XML 전용의 데이터베이스가 상용화되어있으므로, XML 기반의 모션 스크립트를 작성한 경우 쉽게 재사용 할 수 있다.

셋째, XML은 확장성을 위해 설계된 구조를 가지고 있어서 확장성이 뛰어나다. 그러므로 XML 기반으로 작성된 스크립트는 쉽게 수정하거나 확장할 수 있다. 현재 아바타 모션을 기술하기 위한 데이터의 표준 부재로 인해 모든 시스템에서 제한 없이 사용할 수 있는 데이터의 개발이 절실하다. 이러한 상황에서 XML의 확장성은 많은 융통성을 가질 수 있다.

또한 키프레임 애니메이션을 지원하도록 하였다. 그래서 사용자가 프레임별로 필요한 동작을 생성하여 하나의 완벽한 작업 수준의 모션을 만들 수 있다. 각 프레임에 해당하는 동작만을 수정하여 기존 모션을 손쉽게 변화시켜 재사용할 수 있다는 장점도 가진다.

본 논문에서 제안하는 아바타 모션 기술언어의 DTD는 다음과 같다.

```
<!ELEMENT motion (composite?, primitive, parameterization?)>
<!ATTLIST motion
  name                PCDATA          #REQUIRED
  >
<!ELEMENT primitive (StartTime, joint)>
<!ELEMENT StartTime (#CDATA) >
<!ELEMENT joint (HumanoidRoot, l_hip, l_knee, l_ankle, r_hip, r_knee, r_ankle, skullbase, l_shoulder, l_elbow, l_wrist, r_shoulder, r_elbow, r_wrist)
<!ATTLIST joint
  x                    CDATA          #REQUIRED
  y                    CDATA          #REQUIRED
  z                    CDATA          #REQUIRED
  time                 CDATA          #REQUIRED
  > ... 생략
```

4. 작업레벨 모션 생성을 위한 시스템

4.1 모션 데이터

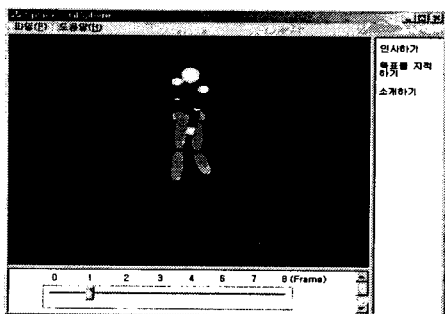
아바타의 모션은 키 프레임 애니메이션에 의해 이루어진다. 따라서 키 프레임별로 아바타에 적용되는 조인트 노드의 회전축, 회전각, 이동좌표 그리고 시간 등의 데이터가 주요한 아바타 모션의 자료가 된다.

따라서 본 시스템에서는 모션 데이터를 저장하기 위한 구조체를 생성하고 프레임별로 사용자가 조작한 아바타의 회전각은 배열의 형태로 저장을 한다.

4.2 시스템 작동 방법 및 모션 생성과정

아바타 모션편집기를 통해서 사용자는 원하는 모션을 생성할 수 있는데, 하나의 작업레벨 모션을 생성하기 위해서는 하나 이상의 프리미티브 모션을 생성하는 과정을 수행해야 한다.

프리미티브 모션은 화면 하단의 프레임 컨트롤을 사용하여 하나의 프레임을 선택하고, 그 프레임에 수행하게 될 아바타의 모션을 사용자가 아바타의 각 부분을 직접 선택하여 마우스 드래그를 통해서 움직임을 준다. 이때 아바타의 조작용을 위한 뷰(View)는 회전(Rotation)이 가능하도록 구현하여 사용자가 원하는 방향을 선택하고 그 상태에서 아바타의 각 부분을 선택하여 조작할 수 있도록 하였다.



[그림 5] 구현된 아바타 모션 편집기

모든 프레임별로 프리미티브 모션 조작성이 끝나면 작업레벨 모션의 이름을 정하고 저장을 한다. 그러면 화면 왼쪽의 리스트 컨트롤 부분에 저장한 작업레벨 모션의 이름이 보이며, 기존 작업레벨 모션의 이름들이 모두 리스트형태로 제시된다.

사용자가 작업레벨모션 리스트에서 원하는 모션을 선택하면 뷰(View)에서 모션을 볼 수 있고, 작업레벨모션들을 선택하여 새로운 작업레벨모션을 생성할 수 있다. 이때 사용자 필요에 따라 모션의 파라미터 값을 변경하여 새로운 작업레벨 모션을 만들 수 있는 인터페이스도 제공한다.

위에서 설명한 과정을 통해서 일반 사용자들도 쉽게 작업레벨모션을 생성하며, 간단한 파라미터 조작만으로 XML 코드에 관한 이해 없이도, 새로운 모션을 만들거나 기존 모션을 수정하여 사용할 수 있다.

```
<?xml version="1.0">
<BehaviorList>
<TaskBehavior name="introduce">
<Composite name="walking"/>
<Motion>
<r_elbow>
<time from="1.0s" to="0.0s">
<rotation x="3.0" y="1.2" z="0">
</r_elbow>
<r_shoulder>
<time from="0.0s" to="1.0s">
<rotation x="0.0" y="0" z="4.0">
</r_elbow>
.....중략.....
</TaskBehavior>
</BehaviorList>
```

[그림 6] 생성된 XML 기반의 모션 기술언어

5. 결론

기존 아바타 모션을 생성하기 위한 편집기 시스템들은 특정 분야에 필요한 동작들만을 정의하여 필요에 따라 취합 선택할 수 있도록 하거나 텍스트 기반의 UI를 이용하여 새로운 모션을 정의 할 수 있도록 하였다. 그러나 이와 같은 시스템은 사용자의 필요에 따라 새로운 모션을 정의 할 수 없고, 표준 방식의 행위 기술언어가 아닌 시스템의 비독립적인 형태의 기술 언어로 저장하도록 되어있어 생성된 모션 자체의 재사용성 측면에서도 어려움이 많다.

이런 문제점들을 개선하기 위해 그래픽 UI를 사용하여 프로그램 언어에 대한 사전지식이 없는 사용자들도 쉽게 아바타를 조작하여 원하는 모션을 생성할 수 있고, 생성된 모션을 표준 XML 형식으로 저장하여 XML 언어의 확장성과 편의성 그리고 저장의 용이함 등의 장점을 활용할 수 있는 편집기 시스템을 제안하였다.

모션 편집기 시스템을 사용하여 사용자는 키프레임 별로 필요한 premetive 모션을 직접 아바타를 조작하여 생성하고, 각 premetive 모션을 하나의 task-level 모션으로 정의 가능하다. 이런 방식을 통해서 생성된 하나의 task-level 모션은 편집기 내에서 간단한 파라미터의 조작만으로 새로운 모션으로 재정의되거나 재사용할 수 있다.

또한 본 논문에서는 특정 도메인내에서 필요한 최소한의 모션만을 정의하여 사용하고 있으나 사용자의 필요에 따라 새로운 도메인에 필요한 동작들도 어플리케이션에 독립적으로 기존 모션 데이터를 이용하여 재정의, 재사용할 수 있다.

[참고문헌]

- [1] Yukari Shirota, et al, "A TV Program Generation System Using Digest Video Scenes and a Scripting Markup Language", HICSS, 2001.
- [2] Masaki Hayashi, et al, "TVML Automatic TV Program Generation from Text-based Script", ABU Technical Review No.190, 2001.
- [3] Kihyun Kim, Sangwook Kim, "Information Extraction and Synchronization Control Method for Multi-Human Animation", WSCG 2000
- [4] Jehhee Lee, Sung Yong Shin, "A Hierarchical Approach to Interactive Motion Editing for Human-like Figures", Proceedings of SIGGRAPH99, 1999.
- [5] Jehhee Lee, et al, "Interactive Control of Avatars Animated with Human Motion Data", Proceedings of SIGGRAPH 2002.
- [6] Chrisian Babski, Daniel Thalmann, "Real-Time Animation And Motion Capture In Web Human Director(WHD)", Web3D 2000, 2000.
- [7] F.Boumpfrey,etc,"Professional XML Applications", Wrox99