

# Web3D 기반 3D 캐릭터 애니메이션 모델링

김근영\*, 이해정\*, 조진애\*\*, 한성국\*, 이용주\*, 정성태\*, 정석태\*

\*원광대학교 전기·전자·정보 공학부

\*\*원광보건대학 패션 디자인과

{gykim1022, redrose, skhan, yjlee, stjung, stjoung}@wonkwang.ac.kr

\*\*jajo@wkhc.ac.kr

## 3D Character Animation Modelling based on Web 3D

Geun-Young Kim\*, Hyae-Jung Lee\*, Jin-Ei Cho\*\*, Sung-kook Han\*, Yong-Ju Lee\*, Sung-Tae Jung\*, Suck-Tae Joung\*

\*Dept of Electronic, Electronic & Information Wonkwang University

\*\*Dept of Fashion Coordination, Wonkwang Health Science College

### 요 약

현대 시각예술의 총아로 불리는 3D 캐릭터 애니메이션 제작기술은 영화, TV, 광고, 인터넷, 게임 등에서 많이 활용되고 있고, 제작기술 또한 나날이 발전하고 있다. 하지만 하나의 완성된 캐릭터 애니메이션이 제작되기까지 수없이 많은 비용과 시간을 필요로 한다. 본 시스템은 Humanoid 기반으로 3D 아바타를 생성하고, 캐릭터 애니메이션의 모델링에 있어서 자연스러운 애니메이션을 좀더 쉽고 유용하게 표현하기 위한 방법을 구현한다. 또한 캐릭터 데이터와 애니메이션 데이터를 어느 정도 분리하여 한번 생성된 애니메이션을 다른 캐릭터에 그대로 적용시켜 재사용할 수 있도록 하였다.

### 1. 서론

컴퓨터가 만들어 내는 3D 애니메이션은 비디오 게임이나 사이버 캐릭터와 같은 단순한 오락의 차원을 넘어 방송과 영화제작을 통한 새로운 문화창출의 도구로 이용되고 있을 뿐만 아니라 디지털 TV, 디지털 콘텐츠, 광고, 인터넷 등의 산업에도 미치는 효과는 대단히 크다. 또한 3D 애니메이션 기술은 인간이 상상할 수 있는 모든 세계를 표현하고 실제 세계에서 실현이 불가능한 것도 가능하게 해 주는 최첨단 표현기술로써 미래 산업에 결정적인 영향을 줄 것이다.

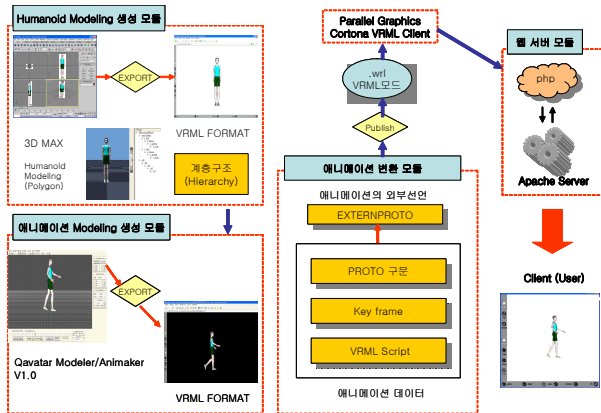
현재까지 많은 애니메이션 기법들이 제안되어 인간 골격 모델의 애니메이션을 생성하는데 이용되었

다.[1][2][3] 애니메이션 기법 중 모션 캡처는 기계적 장비를 이용하여 실제 동작 데이터를 얻어 내는 기술로 모션 캡처에 사용되는 장비가 고가이어서 특정한 움직임에 대한 데이터를 얻기에 경제적 부담이 따를 뿐만 아니라 데이터의 수정과 변형이 어렵다. 동작 제어 애니메이션은 물리적 법칙에 따라 물체의 사실적인 동작을 만들 수 있지만 시스템 구현과 직접적 제어가 어렵다.[4] 또 다른 기법으로 3차원 그래픽 도구나 CAD분야에서 많이 쓰이는 것과 같이 기하학적 계산을 근거로 하여 정밀한 애니메이션을 구성하는 방법도 있다. 하지만 필요에 따라서는 정밀한 구성보다는 짧은 시간에 쉽게 3차원 애니메이션 장면을 구성하는 방법이 필요하다.

본 논문에서는 VRML과 HTML을 접목시켜 캐릭터의 모델 데이터와는 독립적인 방법으로 3차원 애

본 연구는 산업자원부와 한국산업기술재단에서 시행한 지역전략산업 석·박사 연구인력 양성사업의 지원에 의해서 수행하였음.

니메이션을 보다 쉽고 편리하게 생성하여 모델과 애니메이션 데이터를 결합하여 손쉽게 애니메이션을 구현하는데 목적이 있다. 전체적인 시스템 구성도는 (그림 1)과 같다.



(그림 1) 전체적인 시스템 구성도

## 2. Humanoid Modeling 생성

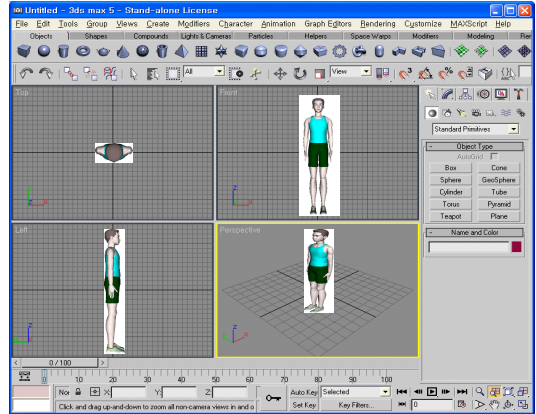


(그림 2) Humanoid Modeling 생성

(그림 2)의 Humanoid Modeling 생성에서는 3차원 인체를 Humanoid 구조 기반으로 분절구조와 계층구조를 반영하여 3D MAX 상에서 인체를 모델링하고 Humanoid 관절 기준에 맞춰 각 관절의 이름을 명시한다. 3D 인체 모델의 기본은 Editable Poly를 이용하여 얼굴, 몸통, 팔, 다리 등의 각각을 구성하고, 세부적인 모양을 만들어 간다. 우선 머리는 머리 형태와 비슷한 구를 이용해 기본 모형을 만든 다음 면을 분할하여 모양을 만든다. 오브젝트의 버텍스가 부족해 각이 많아지면 모디파이어 Mesh Smooth를 적용시킨 다음 Subdivision Amount/Iterations 1을 준다. 눈, 코, 입은 위치를 고려하여 필요한 버텍스를 추가하거나 이동한다. 눈 부분은 코 부분을 고려하여 버텍스를 추가하여 모양을 잡는데 Sub-object Edge 옵션 중 Create와 Divide를 사용한다. 코 부분은 Sub-object Polygon으로 선택해 Extrude 하고, 팔과 다리는 Box를 이용하여 Mesh Smooth 모디파이어를 주고 Iterations 값은 1을 입력, 모디파이어 스택에서도 Editable Poly로 이동해 Show End Result 버튼을 켜 다음 작업을 한다. 모든 작업은 얼굴 모델 제작과 눈, 코, 입, 귀 등을 제작하는 방식과 비슷하다. 이렇게 3D

MAX 상에서 만든 인체를 .wrl로 EXPORT하여 VRML 포맷으로 변환하였다.

인체는 H-Anim 1.1 스펙에 기준하여 모델링되며 다음 (그림 3)은 인체를 3D MAX 상에서 모델링한 모습이다.

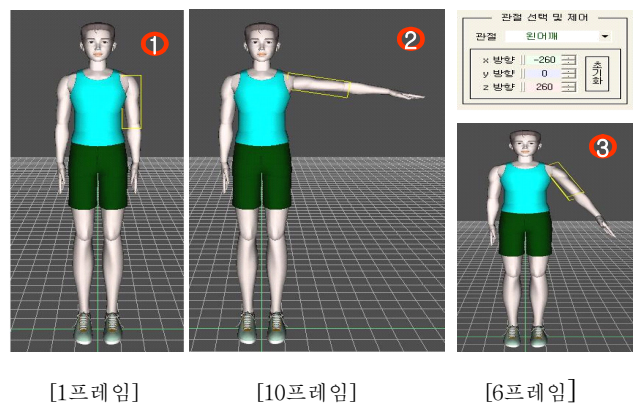


(그림 3) 아바타 모델 생성

3D MAX에서 만들어진 3D 인체를 VRML(.wrl) 파일로 익스포트(Export)하면 사용자의 동작에 반응하는 가상현실 환경에서 사용 가능한 VRML 포맷의 3D 캐릭터가 만들어지게 되는 것이다. 인체를 구성하기 위해서는 H-Anim 1.1 스펙에 기준된 관절 이름을 명시해야 하며 본 논문에서는 Humanoid 기반으로 모델링하기 위해 스펙에 명시된 각 관절의 이름을 사용하였다.

## 3. 애니메이션의 모델링

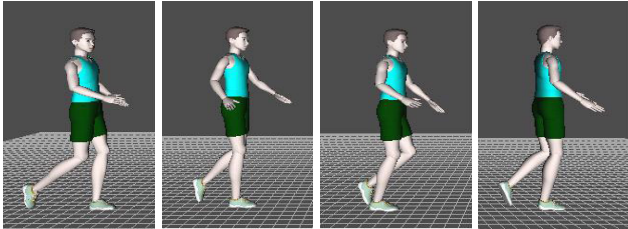
아바타의 애니메이션 모델링은 3차원 그래픽 저작 도구 Qavatar Modeler/Animaker V1.0 소프트웨어를 사용하였으며, 아바타 동작을 모델링하고 EXPORT하여 VRML 포맷으로 변환하였다. (그림 4)는 아바타의 동작 방법을 모델링하는 과정이다.



(그림 4) 아바타의 동작 방법

① 먼저 마우스 오른쪽 버튼으로 아바타의 각부분을 클릭하면 해당부분이 활성화되며 현재 프레임(1

프레임)된다. ② 1프레임이 저장되었으면 10프레임으로 이동한다. ③중간(2~9)프레임의 움직임은 보간법 기능을 이용하여 자동 생성되며, 세부적인 움직임이 필요할 경우 애니메이터 도구모음의 스피너버튼으로 세부조정이 가능하다. 위와 같은 방법으로 달리기 모션을 제작하면 (그림 5)와 같다.



(그림 5) 걷기 애니메이션

#### 4. 애니메이션 시스템

가. 애니메이션의 전체적인 구조

애니메이션을 구성하는 전체적인 VRML 문서의 구조는 다음과 같이 구성된다.

##### VRML file (Avatar Data)

```

H-ANIM Definition
DEF Humanoid Humanoid {
  humanoidBody [
    DEF hanim_HumanoidRoot Joint {
      }
    ]
  }
}

EXTERNPROTO for Animation Data
EXTERNPROTO Behavior_thankyou [
  eventIn SFTIME LaunchAnim
  exposedField SFTIME set_startTime
  exposedField SFTIME set_stopTime
  field MFNode HumansList
]
"animation/thanx.wrl"
    
```

##### VRML file (Animation Data)

```

Keyframe Data
DEF r_ankleRotInterp_BasicWalk OrientationInterpolator {
  key [ ... ]
  keyValue [ ... ]
}
DEF r_kneeRotInterp_BasicWalk OrientationInterpolator {
  key [ ... ]
  keyValue [ ... ]
}
...

ROUTE syntax (VRML Script)
Function Initialize() {
  ...
}
    
```

- H-ANIM Definition : Humanoid에서 정의된 인간의 관절 구조를 기반으로 하여 아바타의 모델을 3D MAX로 작성하여 VRML에서 변환한다.
- EXTERNPROTO for Animation Data : 외부에 작성해 놓은 애니메이션 PROTO 노드를 불러들여 반복 사용한다. PROTO 구문으로 작성된 파일의 위치를 알면 복잡한 구문을 작성할 필요 없이 간단한 코드로 작성이 가능하다.
- Keyframe Data : 아바타의 각 관절부위의 움직임의 좌표가 기술되는 부분이다.
- ROUTE syntax(VRML Script) : VRML 공간에

서 사용자의 행위에서 이벤트를 발생시켜 다른 오브젝트에 변화를 일으키는 동적 환경을 만들어 준다.

여기에서 중요 부분을 좀 더 자세히 살펴보면, 우선 PROTO 노드는 아래와 같이 정의된다.

```

EXTERNPROTO Human_work [
  eventIn SFTIME StartAnim
  eventIn SFBool loop_anim
  exposedField SFTIME set_startTime
  exposedField SFTIME set_stopTime
  field MFNode HumanList
  eventOut SFTIME RealStopTime
]
"work.wrl"
    
```

애니메이션을 위한 EXTERNPROTO 노드는 StartAnim field를 통하여 애니메이션을 구동시킬 수 있으며, HumanList field에 애니메이션이 적용될 아바타 데이터를 연결시키는 것이다. RealStopTime field는 다른 이벤트와의 동기화를 위해 제공된다. 실제 애니메이션 데이터는 별도의 VRML 파일로 저장되어 있으며, 애니메이션 데이터 파일에는 키프레임 애니메이션 데이터와 VRML 스크립트로 구성되어 있다. 본 시스템에서 제공되는 걷기 애니메이션 데이터 파일의 구조를 보면 다음과 같다.

```

PROTO Human_work [
  eventIn SFTIME StartAnim
  eventIn SFBool loop_anim
  exposedField SFTIME set_startTime 0
  exposedField SFTIME set_stopTime 0
  field MFNode HumanList []
  eventOut SFTIME RealStopTime
]
{
  DEF Work_Group Group {
    children [
      DEF Work_Time TimeSensor {
        .....
      }
      DEF L_shoulder OrientationInterpolator {
        key [ 0.000000, 0.050505, 0.101010, ..... ]
        keyValue [ 0.033100 0.998843 0.034880 1.624312,
        ..... ]
      }
      .....
    ]
  }
  DEF work_Script Script {
    }
  }
}
    
```

위 구조와 같이 애니메이션 데이터는 크게 세 부분으로 나눌 수 있다. 첫 번째는 애니메이션에 필요한 타임 센서(Time Sensor)로 애니메이션을 구동시키는 스위치 역할을 하며, 두 번째는 실제 애니메이션을 기술한 키 프레임 데이터이며, 세 번째는

ROUTE 구문을 만들기 위한 VRML 스크립트로써 애니메이션 데이터와 아바타 데이터를 연결시키는 ROUTE 구문을 자동으로 생성하는 역할을 한다.

#### 나. 애니메이션 변환 모듈

본 시스템에서 걷기 애니메이션은 VRML Viewer인 Cortona에서 아바타 모델이 불러들여지고, 웹 페이지에서 타임센서에 의해 이벤트가 발생하면 아바타 데이터 파일안의 Script로 값이 들어와 함수가 실행된다. 이 함수에서 애니메이션을 구동시키는 파일이 EXTERN PROTO 구문을 이용해 로딩되어 실제 사용자 화면에 애니메이션 동작이 보여지게 된다. (그림 6)은 애니메이션의 걷기 동작의 실행화면이다.



(그림6) 걷기 애니메이션 실행 화면

## 6. 결론

본 연구에서는 VRML에서 제공하는 확장 노드를 응용하여 애니메이션을 구현하였다. 본 시스템은 애니메이션 데이터와 아바타 데이터를 연결하는 ROUTE 문을 VRML 스크립트를 이용하여 동적으로 생성시켜 애니메이션 데이터를 약간의 수정으로 아바타 데이터에 쉽게 적용시킬 수 있었다. 또한 애니메이션 PROTO 노드는 EXTERNPROTO 노드를 사용하여 정의되었기 때문에 애니메이션 데이터를 별도의 파일로 분리, 저장할 수 있다. 즉, 애니메이션 데이터와 아바타 데이터의 물리적인 분리가 어느 정도 가능하게 되었다. VRML에서 지원하는 애니메이션은 질적인 애니메이션은 우수하지만 데이터를 다루는 방식에 있어 부족한 점이 있다. 따라서 VRML과 자바EAI를 이용하여 실시간에 인체를 렌더링할 수 있는 시스템과 신체의 각 관절부를 조작할 수 있는 사용자 인터페이스, 인터페이스에 적용되어 나타나는 애니메이션 방법이 모색되어야 할 것이다.

## 참고문헌

- [1] L.Gritz and J.Hahn, Genetic programming for articulated figure motion. *The journal of Visualization and Computer Animation*, vol.6, pp. 129-142, 1988.
- [2] J.Hodgins, W.Wooten, D.Brogan, and J.O'Brien, Animating human athletics. *Proceeding of SIGGRAPH '95*, pp. 8-14, 1995.
- [3] M. van de Panne and E.Fiume, Sensor-actuator networks. *Proceedings of SIGGRAPH '93*, pp. 335-342, Aug. 1993.
- [4] 김형균, "애니메이션의 효율적인 동작 제어에 관한 연구", 조선대학교 석사학위논문, 2003년
- [5] 이상영, "VRML2 (21일 완성)", 인포북, 1997.11
- [6] 고영덕, "VRML 2.0", 혜지원, 1998.7
- [7] VRML 97 International Standard ISO/IEC 14772-2:2004  
[http://www.web3d.org/x3d/specifications/vrml/ISO\\_IEC\\_14772-All/index.html](http://www.web3d.org/x3d/specifications/vrml/ISO_IEC_14772-All/index.html)
- [8] Anim 1.1 Specification for a Standard Humanoid  
<http://h-anim.org/Specifications/H-Anim1.1/#modeling>
- [9] Anim 1.1 Specification for a Standard Humanoid  
<http://h-anim.org/Specifications/H-Anim1.1/#hierarchy>
- [10] Z. Liu, S.J. Gorkler, M.F. Cohen, Hierarchical spacetime control, *Computer Graphics Proceeding, Annual Conference Series (SIGGRAPH 95 Proceedings)*, pp. 63-70, 1995
- [11] <http://www.web3korea.com/web3d/97node/Switch/index.html>  
<http://www.web3d.org/x3d/content/examples/Vrml2.0Sourcebook/index.html>
- [12] 이선호, "Web 3D Programming", 나노북스, 2003.1
- [13] 이민근, "3차원 캐릭터 애니메이션 저작 도구", 수원대학교 자연과학연구소 논문집 제2집, 1999
- [14] 송선희, "3차원 동작데이터 재활용 모델에 관한 연구", 호남대학교, 디지털디자인학 제4집, 2002