
X3D 기반에서의 3차원 Humanoid 모델링과 애니메이션 구현

이성태* · 오근탁** · 김이선** · 이윤배***

Modeling and Animation Implementation of 3D Humanoid
base on the X3D

Sung-Tae Lee* · Guan-Tack Oh** · Yi-Sun Kim** · Yun-Bae Lee***

요 약

3D Humaniod를 통해 인간은 상상할 수 있는 모든 세계를 넘나들고, 실제 세계에서 실현이 불가능한 것도 가능하게 해주는 대리 역할을 한다. 더욱이 인터넷상에서의 3D 게임이나 가상현실등의 증가로 인해 현실세계에서 존재하는 물체들에 근접하게 모델링하기 위한 3D 캐릭터 애니메이션으로 관심이 쏠리고 있다.

본 논문에서는 3차원 객체를 높은 품질로 표현하기 위해서 새로운 framework을 적용한다. 다양하고 자연스러운 형태의 표현을 위해 Web3D 자료 형태인 VRML과 XML을 이용하여 3차원 Humanoid의 모델링과 애니메이션 기법을 제안하고 Simulation을 통해 타당성을 검증한다.

ABSTRACT

We can travel every imaginary world and do every impossible thing via 3D character. People interest in 3D character animation for modeling like real world due to the increase of 3D game and imaginary Virtual Reality on the internet.

In this paper, new framework for present of 3D character of high quality is applied. I represent 3-dimensional Humanoid modeling and animation technology and show the adequacy via simulation for various and natural representation certificate using VRML as a Web3D information type and XML.

키워드

X3D, Humaniod, VRML, XML

1. 서 론

3차원 캐릭터 애니메이션 가운데 특히 인체의 단관절 캐릭터 모델의 애니메이션은 그 복잡한 구조와 운동의 다양함으로 인하여 캐릭터 모델의 동작제어 및 동작발생기술이 캐릭터의 사실적인 애니메이션을 위하여 필수적일 뿐만 아니라 애니메이션의 요소기술로서도 매우 중요한 역할을 하고 있다. 현재 인체와 같은 단관절 캐

릭터의 애니메이션에서 캐릭터의 동작을 제어하고 발생시키는 방법에는 여러가지가 있다. 캐릭터 애니메이션 기술을 크게 분류하면 전통적인 셀 애니메이션 기법과 같이 움직임의 중요 장면들을 수작업에 의해 생성하고 이들 사이에 중간 장면들을 채우는 방식으로 제작되는 「키프레임(Key Frame) 기술」, 일련의 규칙에 의해 대상체의 움직임을 자동적 또는 반자동적으로 생성하는 「동작제어(Motion Control)기술」에 의한

*서울사이버대학교 IT학부 게임&애니메이션학과

**조선대학교 전자계산학과

***조선대학교 컴퓨터공학부 우수논문

접수일자 : 2002. 11. 1

애니메이션 그리고 3차원 공간상에서 실제 배우의 움직임을 직접 기록할 수 있는 장비로 캐릭터의 움직임을 생성하는 「모션 캡처(Motion Capture)」으로 나눌 수 있다. 이밖에도 인간의 감정 차이를 잘 나타낼 수 있도록 표정 및 입술 움직임 등을 미세하게 표현하는 「얼굴 애니메이션 기술」과 팔, 다리 등 관절 부위로 이루어진 전신의 자세를 제어해 동작을 표현하는 「다관절체 애니메이션 기술」등도 있다.

2차원 정보만을 기반으로 있던 웹이 이제 3차원 입체화면을 통하여 시각적 효과를 높인 가상현실에서 캐릭터를 생성해서 캐릭터 애니메이션을 할 수 있다.

그런데, 3차원 캐릭터 모델링 도구는 캐릭터의 외형 및 골격 구조를 표현하는 점, 선, 면의 구조를 생성, 편집, 삭제 등의 연산 및 이의 변형, 간단화 등을 효율적으로 수행할 수 있도록 구현되어야 한다. 본 논문에서는 최근 웹에서 사용이 증가하고 있는 XML과 VRML 언어를 이용한 차세대 확장가능 언어인 X3D를 이용하여 Web3D기반 3D Humanoid 의 모델링과 애니메이션 기법을 제안하고, Simulation을 통해 검증한다.

2. Humanoid 애니메이션 모델링

인간의 몸체는 얼굴과 달리 각각의 관절을 중심으로 회전하고, 비틀리고, 움직이게 된다. 또, 몸체의 계층 구조상 어느 특정 부분을 움직이면 상위의 모든 계층들까지 영향을 미치게 된다. 따라서 각 부분을 알맞은 계층구조를 갖도록 구성하고, 이에 맞는 적절한 관절의 위치 설정은 몸체에 있어서 가장 중요한 요소로 작용하게 된다.

2.1 Humanoid 기능 분석 및 모델링

인체모델은 근 골격구조, 근력, 인체치수 등 인간의 신체특성을 고려하여 개발되어야 하며, 설계상의 검증, 평가를 위하여 컴퓨터를 이용한 3차원 인간공학적인 인체모델의 개발과 3차원 Dummy모형의 개발이 요구된다.

인체 모델의 개발에 대한 연구에서는 인체특성을 수집, 확보하고 이를 데이터베이스화 한다. 또한 인체 동작의 기초자료로서 인체 주요 관절의 불편도(Discomfort)에 대한 연구 등의 인체의 동작에 대한 특성 분석 및 평가 연구를 수행하며, 인체 동작을 컴퓨터 시뮬레이션 할 수 있는

모델을 구현한다. 그후 구축된 데이터베이스와 기구학적 기법을 이용하여 인체를 모델링하고, 근력의 모형 및 인체치수, 근골격구조 등을 고려한 소프트웨어적 휴먼 모델을 개발한다.

2.2 Humanoid의 신체특성 분석

현재 미국의 NASA나 우리나라의 표준 과학 연구원에서 인체의 다양한 Dimension을 측정, 제시하고 있으나, 이들 모두가 인체의 표현에 사용되지는 않아 이들 자료로부터 인체 모델을 현실감 있게 표현하는데 필요한 인체 치수의 파악이 필수적이다. 인체모델의 개발을 위해서는 각 지체의 회전축의 위치와 방향을 측정하고 이로부터 각 지체의 길이를 도출하며 여기에 체적(Volume)을 부과해야 한다. 인체 표면을 기준으로 측정된 인체측정 자료로부터 이를 도출하기는 불가능하므로 실험을 통하여 위의 항목들을 조사하고 이를 기존의 인체측정자료와 함께 데이터베이스화한다. 그림 1은 인체의 관절을 표현한 것이다.

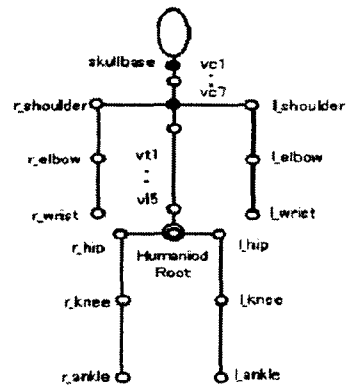


그림 1. X3D Humanoid joint node

2.3 Humanoid 동작 분석 및 시뮬레이션 연구

인체 동작의 기초자료로서 인체 주요 관절의 동작범위와 불편도(Discomfort)에 대한 연구 등, 인체의 동작에 대한 특성을 분석 평가하는 연구를 수행한다. 다양한 자세에 대한 인체 관절의 불편도는 동작 자세 예측 등의 인체 자세 예측 및 자세 평가의 기초자료로 활용될 수 있으며, 제품의 설계에도 기초자료로 활용될 수 있다. 제품 사용시의 인간의 동작 범위 및 사용편의성, Reach, 근력, Clearance 등의 여러 인간공학적인 평가 요소를 기준으로 평가방안을 수립하고 이를 정량적으로 평가하는 평가모델을 개발한다.

3. Humanoid 모델링의 노드

물체의 애니메이션을 VRML 브라우저 상에서 구현할 경우 있어서 먼저 필요한 것이 물체의 형태에 대한 표현이다. 간략화된 물체의 모델을 먼저 VRML에서 구현한 후 애니메이션을 추가하여야 한다.

물체의 애니메이션의 구현을 위해 사용되는 VRML의 노드들은 다음과 같다. 지오메트리(geometry)노드는 물체를 표현하는데 사용하고, 그룹(group)노드는 계층적 구조의 물체를 만들며, 인터polator(interpolator)노드는 키 프레임 방식의 애니메이션에 사용한다. 그밖에 조명과 배경에 관련된 노드들도 사용한다.

본 논문에서는 PROTO 타입을 정의한 후, 지오메트리노드를 사용하여 대리인을 구성하였다. 물체에 있어서의 애니메이션은 각 부분들이 하나의 객체로서 움직이기 때문에 서로의 링크 정보를 가지고 있어야 한다. 그러므로 그룹노드를 사용하여 이러한 기능을 가능하게 한다. 또한 인터polator 노드는 물체의 애니메이션 파라미터를 키와 킷값을 이용하는 키프레임 방식으로 애니메이션을 발생시킨다. : 2002. 2. 15

```
Humanoid Definition
DEF Humanoid Humanoid{
  humanoiBody[
    DEF hanim_humanoidRoot Joint{
    .....
    }
  ]
}
```

그림 2. 아바타 Humanoid의 정의

```
DEF r_ankleRotInterp_BasicWalk
OrientationInterpolator{
  key[...]
  keyValue[...]
}
DEF r_ankleRotInterp_BasicWalk
OrientationInterpolator{
  key[...]
  keyValue[...]
}
```

그림 3. 관절에 대응하는 애니메이션 데이터

```
ROUTE syntax
ROUTE
  r_ankleRotInterp_Stand.value_changed TO
  r_ankle.set_rotation
ROUTE
  r_kneeRotInterp_Stand.value_changed TO
  r_knee.set_rotation
```

그림 4. ROUTE구문을 이용한 아바타 데이터와 애니메이션 데이터를 연결

3.1 Humanoid 모델 개발 및 동작

인체자료 데이터베이스를 바탕으로 연립면, 나이별로 신체 크기와 동작 특성이 다를 수 있으므로 다양한 층의 인체 크기를 반영할 수 있는 소프트웨어적인 인체모델을 개발한다. 또한 인체 애니메이션 기술은 역운동학 기반 애니메이션 제어시 인체의 특성을 고려한 계산식의 적용으로 실시간 제어가 가능하며, 모션 캡처 자료의 수치적 해석에 의하여 적은 양의 데이터만으로 동작을 재구성할 수 있고, 나아가 새로운 동작을 생성할 수도 있다.

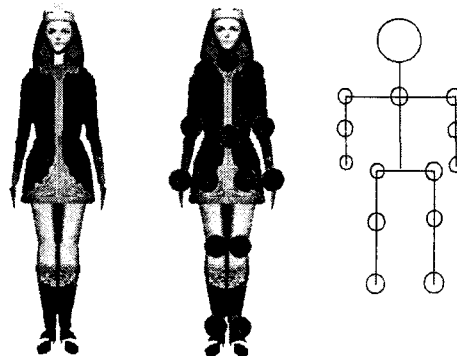


그림 5. 분절구조의 모델링과 피벗의 설정

3.2 Humanoid 모형의 방법

Humanoid 형태의 기하형상은 두 가지 방법으로 서술할 수 있다. 첫 째는 관절 계층으로 이루어진 장면 그래프로써, 이것은 Humanoid 노드의 skeleton 필드(field)에 서술된다. 관절 계층의 Segment노드에서 정의된 (Shape의 형태를 한)

기하형상은 신체를 분리된 기하학적 부분으로 서술한다. 이 방법은 기계적으로 계산할 때에는 효율적이지만, (이어 붙인 자극이나 주름같은) 시각적인 부자연스러움이 있기 때문에 인체모형 형태의 외형에서 품위를 떨어뜨린다. 두 번째 방법은 Humanoid 인체모형 형태의 기하형상을 연속된 하나의 기하형상인 것처럼 서술하는 것으로, Humanoid 노드의 skin 필드를 사용한다. 이 방법은 (각각 Coordinate와 Normal 노드의 형태를 하고 있는) skinCoord와 skinNormal 필드에 위치점과 법선 벡터(normal vector) 집합을 먼저 정의하여야 한다. 이러한 방식으로 정의된 데이터는 Humanoid 노드의 정의에서 두 가지 다른 객체에 의해 참조될 수 있도록 한다. 첫 번째 객체는 IndexedFaceSet 노드로, 이것은 skin 필드의 인체모형 형상의 기하형상 표면을 정의한다. 대부분의 경우에 이 표현은 단일한 IndexedFaceSet 노드이지만, 여러 개의 IndexedFaceSet 노드로 정의된 것처럼 만들어 질 수도 있다. IndexedFaceSet의 구현 방법과 인체모형 형태의 구성에 따라서, 연속적인 망 구조의 변화를 제한된 표면으로 국한시킴으로써 복수의 IndexedFaceSet 노드로 나누는 편이 더 좋은 수행성을 제공할 수도 있다.

4. Humanoid 모형 Simulation

인체는 사용자가 이동하거나 회전할 시에 걷는 동작을 실행한다. 걷는 동작을 위한 애니메이션 정보는 사용자가 처음에 서버에 접속할 때 적재하는 아바타의 외형정보에 포함되어 있으며 이들 정보는 CoordinateInterpolator는 TimeSensor로부터 입력되는 시간값을 실행시간에 계산하며 인체의 동작을 보간해서 애니메이션이 이루어진다. 클라이언트의 VRML 브라우저 상에서 나타나는 인체 애니메이션을 위해서 특정 메시지는 필요하지 않다. 서버로부터 전송되는 인체의 이동정보만 있으면 클라이언트에서는 이동정보에 맞추어서 애니메이션을 동기화 하면 되기 때문이다.

```
#VRML
DEF SyncTouchSensor Script{
    eventIn SFBool start
    eventOut SFTIME stime
    field SFBool state FALSE
```

```
    uel "vrmlexport:
        function start(value) {
            state=value
            stime=new
SFTIME();
        }"
```

다음은 3차원 Humanoid 구조를 VRML로 처리 하기 위한 알고리즘이다.

```
1: #VRML V2.0 utf8
2: PROTO XVL_EDGE [
3:   field SFFloat round_val 0
4:   field SFVec3f round_str 0 0 0
5:   field SFVec3f round_end 0 0 0
6: ]
7: { Text { string [ " weight of adge " ] } }
8: PROTO XVL_STATUS [
9:   field SFString status of shape" ] }
10: Group{
11:   children [
12:     Group {
13:       children [
14:         shape {
15:           geometry IndexedFaceSet {
16:             }
17:         }
18:       }
19:     XVL_STATUS {
20:       status "XVL_GREGORY"
21:     }
22:   XVL_EDGE {
23:     round_val 0.5
24:     round_str 0 0 0
25:     round_end 0 0 0
26:   }
27:   IndexedLineSet {
28:     coordIndex [ 24 103 ]
29:   }
30: ]
31: whichChoice -1
32: }
33: }
```

그림 6. Humanoid object 구조를 VRML로 처리하기 위한 알고리즘

위의 VRML코드는 애니메이션 시작점을 결정하는 TouchSensor를 동기화가 가능하도록 스크립트 노드를 정의하였다. 사용자가 시스템에 전송하면 이 노드를 통해서 이벤트를 호출하는 것을 시작으로 해서 애니메이션의 시작을 결정할 수 있다. 즉, 애니메이션 정보를 가지는 모든 객체들은 그들의 행위를 시작하는 방법으로 두 가

지의 라우팅 경로를 가지게 되는 것이다.

VRML 변환 규칙에 의하여 3차원 Graphic을 VRML로 구성한다. 구성된 각 객체의 노드 유형은 관련된 기능을 가진 노드에 맞도록 추상(abstract) 노드 기반으로 되어있다. 객체 계층 구조에 대한 그림은 추상 노드 유형(객체 계층구조에서 "Node"의 뒤에 붙어있는 명칭들)에 기반을 둔 각각의 노드 간의 연관성과 기원을 보여준다. 예를 들어 VRML 97 규약에서 정의되고 X3D DTD에서 다시 되풀이 되는 GeometryNode, LightNodes, SensorNode, BindableNode 등이 이에 해당 된다. Tetrahedron 노드 정의는 GeometryNode 유형으로부터 파생되어 나올 것이므로, 이 노드는 Box, Sphere 또는 다른 기하(geometry) 노드가 놓일 수 있는 장면 그래프의 어느 곳이나 놓일 수 있다.

VRML로 구성된 3D Graphic은 XML에 의해서 다이나믹한 형태의 X3D로 구현이 되어진다. 다음은 X3D의 Humanoid Object 구조의 DTD(Document Type Definition)이다.

```

1: <!ENTITY % XvlShellTypes "CS_POLYLN |
2: CS_BEZIER | SS_POLYGON | SS_LATTICE |
3: SS_GREGORY " >
4: <!ELEMENT XvlShell (XYZ, UV, Faces*,
5: Edges*,
6: Vertices*0)>
7: <!ATTLIST XvlShell
8: Type (%XvlShellTypes;) #REQUIRED
9: DEF ID \#IMPLIED
10: USE IDREF \#IMPLIED
11: >
12: <!ELEMENT Faces (Material?, Texture?,
13: Face+)>
14: <!ELEMENT Edges (Material?, Edge+)>
15: <!ELEMENT Vertices (Material?, Texture?
16: %XvlG1T1FaceChild;) >
17: <!ATTLIST Face
18: ix %MFInt32; #REQUIRED
19: tx %MFInt32; #IMPLIED
20: >
21: <!ELEMENT Edge (Material?
22: %XvlG1T1EdgeChild; ) >
23: <!ATTLIST Edge
24: ix %MFInt32; #REQUIRED
25: vec %MFInt32; #IMPLIED
26: VRnd %SFFloat; #IMPLIED
27: >
28: <!ELEMENT Vertex (Material?) >
29: <C
30: ix %MFInt32; #REQUIRED
31: VRnd %SFFloat; #IMPLIED

```

```

30: >
31: <!ELEMENT Texture (EMPTY) >
32: <!ELEMENT Texture
33: url CDATA \#IMPLIED
34: repeaatS %SFBool; "true"
35: repeaatT %SFBool; "true"
36: url IDREF \#IMPLIED
37: url ID \#IMPLIED
38: url IDREF \#IMPLIED
39: >
40: <!ELEMENT XYZ \#PCDATA>
41: <!ELEMENT UV \#PCDATA>

```

그림 7. X3D의 Humanoid Object 구조의 DTD

5. 결 론

본 논문에서 제시한 기술은 미래 국내 인터넷 기반 기술, 3차원 쇼핑몰, 가상사무실, 3차원 시뮬레이션 게임, 가상 은행등에 이르는 폭 넓은 영역에 기본 모듈로써 응용 될 수 있을 것으로 생각된다. 이를 이용하여 교육 및 의료에 활용한다면 지금까지 이차원에서 표현되는 한계를 극복하여 좀 더 사실적이고 효과적인 인체 애니메이션이 될 것이다. 현재 기술로는 임의 객체들의 움직임을 완전히 자동적으로 생성하거나 제어할 수 없지만 일정한 규칙하에서 어느 정도 변화폭을 가지는 특정한 유형의 움직임은 알고리즘화 될수는 있다. 그러므로 우리가 차후에 할 일은 임의 객체들의 움직임을 완전한 규칙하에서 변화할 수 있도록 해야 할 것이며, 3D 애니메이션으로 관심이 쏠리고 있는데 반해 3차원 모델/애니메이션 데이터를 저장하는데 많은 메모리를 필요로 하고 전송시간 또한 오래 걸린다. 그러므로 메모리와 전송시간의 향상 및 처리속도를 향상시키기 위해 3차원 애니메이션 부호화 기술 개발이 필수적이다. 그리고 멀티미디어를 지원하기 위한 Sound가 요구되며 현재 진행되고 있다. 삼차원 브라우저는 X3D 파일을 처리할 수 있도록 개선되어야 할 것이다.

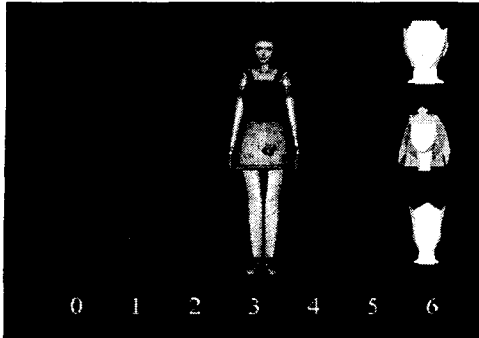


그림 8. 3차원 Humaniod의 Animation 동작 시스템

참고문헌

- [1] Tim Bary, et. al. XML 1.0 Specification.
- [2] B. Roehl, Specification for a Standard VRML Humanoid Version 1.0 August 1997; [http:// ece.uwaterloo.ca:80/p h-anim/](http://ece.uwaterloo.ca:80/p/h-anim/)
- [3] International Standard ISO/IEC Virtual Reality Modeling Language, 1997.
- [4] W.W. Armstrong, M. Green, R. Lake, Near-realtime control of human figure models, IEEE Comput. Graphics Appl. 7 (6) (1987) 52 61.
- [5] VRML97, International Standard: ISO/ IEC 14772-1, December, 1997; [http:// www.vrml.org/ Specifications /VRML97/index.html](http://www.vrml.org/Specifications/VRML97/index.html).
- [6] World Wide Web Consortium : Extensible Markup Language(XML) [http://www.w3.org/XML/\(2000\)](http://www.w3.org/XML/(2000)).
- [7] "VRML Script Working Group Clarifications," Availableas "http://www.vlc.com.au/~justin/vrml/script-wg/".
- [8] Robert Tarjan, "Depth-First Serach And Linear Graph Algorithms," SIAM Journal on Computing, Vol. 1, No. 2, June 1997, pages 146~160

저자소개



이성태(Sung-tae Lee)

1994년 2월 광주대학교 전자계산학과 공학사
 1998년 8월 조선대학교 전자계산학과 이학석사
 2002년 2월 조선대학교 전자계산학과 박사과정 수료

1999년 5월~2002년 8월 (주) 베네테크 연구소 선임연구원

2002년 9월~현재 서울사이버대학교 IT학부 게임&애니메이션학과 전임강사

※관심분야: 가상현실, 3차원캐릭터, 멀티미디어

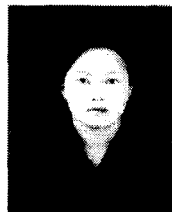


오근탁(Guan-Tack Oh)

1998년 2월 동신대학교 컴퓨터공학과 공학사
 2001년 2월 조선대학교 전자계산학과 이학석사
 2002년 3월~현재 조선대학교 전자계산학과 박사과정

1997년~2002년 현재 서린정보시스템 대표

※관심분야: 모바일 프로그래밍, 의료영상처리



김이선(Yi-Sun Kim)

2001년 2월 대불대학교 컴퓨터교육학과
 2001년 3월~현재 조선대학교 전자계산학과 석사과정
 ※관심분야: 3차원 캐릭터, 가상현실, 멀티미디어



이윤배(Yun-bae Lee)

1980년 2월 광운대학교 전자계산학과 이학사
 1983년 2월 광운대학교 전자계산학과 이학석사
 1993년 12월 숭실대학교 전자계산학과 공학박사

1988년~현재 조선대학교 컴퓨터공학부 교수

※관심분야: Artificial Intelligence, Expert Systems, Multimedia, Image Processing, Virtual Reality