

# MPEG-4 SNHC 기반의 Body Animation 구현

안정환\*, 김성열\*, 허영\*, 한규서\*\*, 전병태\*\*, 호요성\*

\*광주과학기술원 정보통신공학과

전화: (062) 970-2268/ 팩스: (062) 970-2204

\*\*한국전자통신연구원 컴퓨터소프트웨어연구소

전화: (042) 860-5257/ 팩스: (042) 860-4844

## Implementation of Body Animation based on MPEG-4 SNHC

Jeong-Hwan Ahn\*, Sung-Yeol Kim\*, Yung Hur\*, Kyuseo Han\*\*, Byung-Tae Chun\*\*, Yo-Sung Ho\*

\*Kwangju Institute of Science and Technology

E-mail: {jhahn, sykim75, yunghur, hoyo}@kjist.ac.kr

\*\*Electronics and Telecommunications Research Institute

E-mail: {kyuseo, chunbt}@etri.re.kr

### Abstract

The MPEG-4 SNHC subgroup has developed an international standard to support multi-user communications at low bit rates by encoding and transmitting feature parameters of the human face and body in virtual environments. In this paper, we have implemented a body animation system using body definition parameters (BDP) and body animation parameters (BAP), following the MPEG-4 SNHC standard. In addition, the developed system can animate 3-D avatars by BAP values.

### I. 서론

최근 다중 사용자가 네트워크를 통한 삼차원 가상환경에서 소리와 시각으로 통신을 하며, 가상 환경에 있는 멀티미디어 정보를 서로 상호 작용 하는데 많은 관심이 쏟아지고 있다. 즉, 사용자가 가상환경 속에서 그래픽적인 표현 방법에 의해 하나의 일부분으로 융화되어 마치 사람처럼 표현된다.

이렇게 가상환경에서 사람을 표현하기 위해 MPEG-4 (SNHC) Synthetic and Natural Hybrid Coding 그룹[1]에서는 사용자를 대신하는 아바타(Avatar)의 구현 및 압축 전송 방법에 대한 표준안을 제시하였다. 현재 SNHC에서는 얼굴(Face)과 몸통(Body)을 렌더링(Rendering)이 가능한 3D 폴리곤 메시(Polygon Mesh) 형태의 객체로 정의물 하고 있으며, 얼굴의 형태와

텍스처 및 표정은 비트열 속에 포함되어 있는 FDP (Facial Definition Parameter)와 FAP (Facial Animation Parameter)에 의해서 결정된다. 또한 몸통의 형태와 움직임은 BDP (Body Definition Parameter)와 BAP (Body Animation Parameter)로 정의된다 [2].

본 논문에서는 MPEG-4 표준에 맞게 간략화된 BDP를 만들고 OpenGL을 이용하여 렌더링하였다. 또한 표준 규격에 맞는 BAP를 읽어, 이에 맞게 움직일 수 있는 시스템을 구현하였다.

### II. 몸통 애니메이션의 구현

본 장에서는 MPEG-4 SNHC에서 정의한 BDP와 BAP에 대해 살펴본 후, 본 논문에서 사용한 몸통, 표면 정보(Segment)와 관절(Joint)구조에 대해 알아본다. 그리고 OpenGL을 사용하여 아바타를 렌더링하고, BAP에 따라 애니메이션하는 방법에 대해 알아본다.

#### 1. Body Definition Parameter (BDP)

BDP는 수신단에서 사용자가 정의하는 특정한 몸으로 특성화하거나 몸통을 움직이게 하는 정보에 따라 다운로드하여 사용한다. BDP는 일반적으로 사용자가 어떤 응용물을 이용하는 동안 한번만 전송된다. 만일 수신단이 BDP를 전송받지 않는다면 기본적인 모델을 사용하며 BAP를 해석할 수 있어야 한다. 따라서 방송이나 영상회의에서 최소한의 작동을

보장 해주게 된다. 그리고 BDP의 몸통 표면 기하학 정보와 관절은 H-Anim[2]에서 정의한 문법대로 표 1의 VRML[3] PROTO Segment[3]와 표 2의 VRML PROTO joints[3]를 사용하여 규정되어야 한다 [1].

표 1. Segment PROTO

```
PROTO Segment [
  exposedField SFString name          ""
  exposedField SFfloat mass           0
  exposedField SFVec3f centerOfMass   0 0 0
  field SFVec3f bboxCenter            0 0 0
  field SFVec3f bboxSize              -1 -1 -1
  exposedField MFNode children        [ ]
]
```

표 2. Joint PROTO

```
PROTO Joint [
  exposedField SFString name          ""
  exposedField SFVec3f translation     0 0 0
  exposedField SFRotation rotation     0 0 1 0
  exposedField SFVec3f scale          1 1 1
  exposedField SFRotation scaleOrientation 0 0 10
  exposedField SFVec3f center         0 0 0
  exposedField MFNode children        [ ]
  exposedField MFFloat ulimit         [ 0 0 0 ]
  exposedField MFFloat llimit         [ 0 0 0 ]
  exposedField SFString eulerSequence "XYZ"
]
```

인간의 몸통은 얼굴과 다리 각각의 관절을 중심으로 회전하고, 비틀리고 움직인다. 또, 몸체의 계층 구조상으로 어느 특정 부분을 움직이면, 상위의 모든 계층들까지 영향을 미치게 된다. 따라서, 각 부분을 알맞은 계층 구조를 가지게 구성하고, 이에 맞는 적절한 관절의 위치 설정은 몸통에 있어서 가장 중요한 요소로 적용하게 된다.

본 논문에서는 SNHC에서 정의하는 몸통에 대한 특징점들과 호환성을 유지하면서, 어느 특정 부분을 움직이면, 상위의 모든 계층들까지 영향을 미치는 계층 구조로 구성하였다.

표 3은 설계된 모델에 대한 설명이다. 표 3에서 보듯이, 몸통 표면은 16개로 개별적으로 설계하였다. 표 3에서 설명하는 특징점은 SNHC 표준에서 정의하는 특징점과 대응되는 점 위치이다.

그림 1은 실제 구현된 Nancy 모델을 보여준다. MPEG-4 SNHC에서 정의한 모델의 초기 자세는 시선은 앞을 바라보고 손등은 바깥쪽을 향하고 있어야 한다. 구현된 모델은 표준에서 기술한 모습을 하고 있다. 그림 1에서 보듯이, 구현된 모델은 골반을 기준으로 상반신과 하반신으로 나뉜다. 상반신에는 척추가 있고 척추에는 가슴과 허리로 구분된다. 여기에 각각의

왼쪽과 오른쪽 어깨에 대해 상박(upperarm), 하박(forcarm), 손이 연결된다. 하반신에는 엉덩이, 넓적다리, 정강이, 발이 있다. 현재 손과 발은 5개의 손가락과 발가락의 관절이 하나로 이루어진 단순한 형태를 하고 있다.

표 3. 몸통 표면의 정의

번호	특징점	설명
1	Pelvis	골반
2	l_thigh	왼쪽 넓적다리
3	l_calf	왼쪽 정강이
4	l_foot	왼쪽 발
5	r_thigh	오른쪽 넓적다리
6	r_calf	오른쪽 정강이
7	r_foot	오른쪽 발
8	c7	목
9	l_upperarm	왼쪽 상박
10	l_forearm	왼쪽 하박
11	l_hand	왼쪽 손
12	r_upperarm	오른쪽 상박
13	r_forearm	오른쪽 하박
14	r_hand	오른쪽 손
15	c4	몸통
16	Skull	얼굴

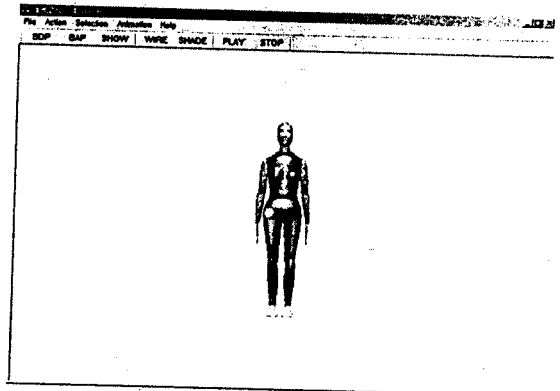


그림 1. 구현된 모델

표 4는 본 논문에서 사용한 관절의 위치이다. 표 4에서 보듯이, 17개의 관절로 이루어져 있으며, 특징점에서 기술한 점의 위치는 표준과 대응된다. 크게 몸통은 골반을 기준으로 상반신과 하반신으로 나뉘며, 상반신에는 간략한 구조를 가진 척추가 있다. 실제 척추는 경추, 흉추, 요추로 구성되지만, 현재는 각 부분이 서로 연결되는 부분을 vl1, vc4, vc8로 정의하고 단순한 척추의 움직임만을 기술한다. 따라서 실제 복잡한 척추 움직임은 지금 단계에서는 표현 할 수 없다. 척추를 기준으로 어깨, 팔꿈치, 손목이 있고, 목과 얼굴이 연결된다. 하반신에는 엉덩이 무릎, 발목이 있다. 또 17번째 중심점(Root)은 전체적인 몸통 움직임의 기준이 된다.

표 4. 몸통 관절

번호	특징점	설명
1	pelvis	골반
2	vl1	허리
3	vc4	목
4	vc8	얼굴
5	l_shoulder	왼쪽 어깨
6	l_elbow	왼쪽 팔꿈치
7	l_wrist	왼쪽 손목
8	r_shoulder	왼쪽 어깨
9	r_elbow	왼쪽 팔꿈치
10	r_wrist	왼쪽 손목
11	l_hip	왼쪽 엉덩이
12	l_knee	왼쪽 무릎
13	l_ankle	왼쪽 발목
14	r_hip	오른쪽 엉덩이
15	r_knee	오른쪽 무릎
16	r_ankle	오른쪽 발목
17	root	중심점

그림 2는 표 4에서 기술한 관절의 위치를 보여준다.

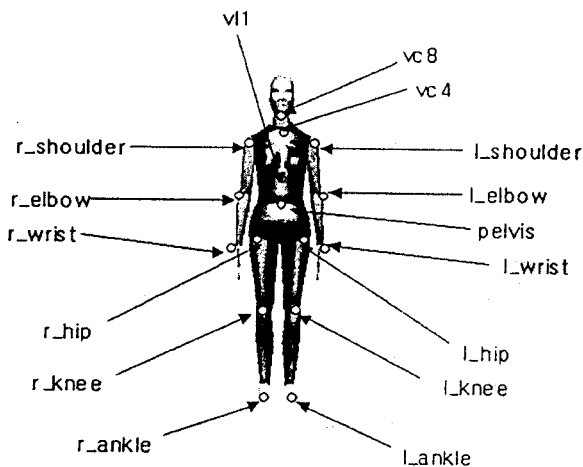


그림 2. 관절 구조

## 2. Body Animation Parameter (BAP)

MPEG-4 SNHC그룹에서는 몸통의 전체적인 움직임을 위해 6개의 Degree of Freedom (DOF)와 간단한 척추구조를 가진 내부적인 움직임을 위해 66개의 DOF를 정의하고 있다. 손가락 움직임은 별도의 선택사항으로 두고 있다.

BAP는 몸통의 애니메이션을 정의하는 변수로 어떤 몸통 부분이 어떤 관절에 대해서 매 프레임마다 얼마나 회전(Rotation)하거나 이동(Translation)하는지에 대한 움직임 정보를 가지고 있다. 현재 BAP는 169개로 구성되어 있으며, 표 5와 같이 19개의 그룹으로 나뉘며 각 그룹에 해당하는 BAP개수는 비트 마스크에 나와있다.

BAP 파일은 헤더정보, BAP 마스크, BAP 데이터 3부분으로 구성되어 있다.

(1) 헤더 설명:

3.0 example 30 400

→ <버전 정보> <파일이름> <fps> <시퀀스 길이>

(2) BAP 마스크:

→ 169개의 BAP 변수중 사용되는 변수만 '1'로 설정하고 나머지는 '0'으로 설정한다.

(3) BAP 데이터

→ <프레임 넘버> <BAP 데이터>

라인의 처음에는 몇 번째 프레임인지 프레임 넘버가 나온다. 그리고 BAP mask에서 1로 설정된 BAP에 대한 움직임 값이 나온다. 따라서 이 값을 읽어 애니메이션 시킨다.

표 5. BAP 구성

Group	Group Name	Bit Mask
1	Pelvis	3
2	Left leg1	4
3	Right leg1	4
4	Left leg2	6
5	Right leg2	6
6	Left arm1	5
7	Right arm1	5
8	Left arm2	7
9	Right arm2	7
10	Spine1	9
11	Spine2	15
12	Spine3	18
13	Spine4	18
14	Spine5	12
15	Left hand1	16
16	Right hand1	16
17	Left hand2	9
18	Right hand2	9
19	Global positioning	6

## 3. OpenGL을 이용한 몸통 애니메이션 구현

### 3.1 OpenGL

OpenGL[4]은 삼차원 그래픽 프로그램을 지원하는 그래픽 라이브러리(Library)이다. 삼차원 객체를 구현할 수 있는 OpenGL은 뷰(View)를 설정하는 함수, 상태 변수를 바꾸는 함수, 기하학적 정보를 제공하는 함수, 이미지의 일부를 선택하는 함수 등을 제공한다. 또한 몸통을 구현하고 움직이는데 있어 상위계층의 애니메이션이 하위 계층의 애니메이션에 영향을 주는 계층구조로 구현하기 위해 OpenGL에서 제공하는 glPushMatrix()와 glPopMatrix()을 사용하여 계층적으로 표현하였다.

### 3.2 시스템 구조

그림 3은 본 논문에서 만든 전체적인 시스템 구조로 나타내며, 크게 Model Analyzer, Parser, Display의 3부분으로 구성되어 있다.

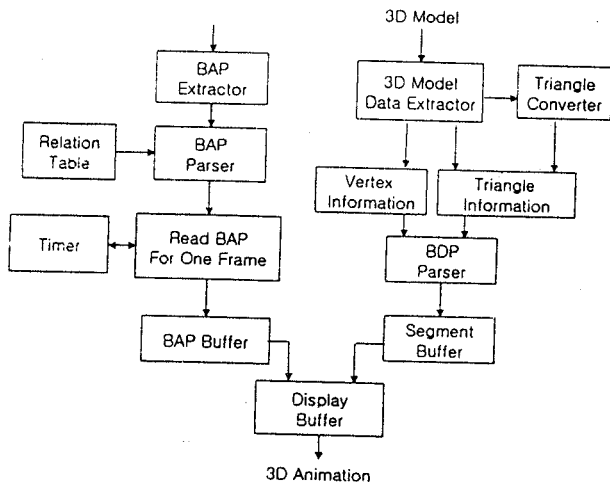


그림 3. 전체적인 시스템 구성도

표 1과 표 2에서 기술한 구조를 가지고 있는 VRML 데이터가 들어오면 입력 데이터로부터 세그먼트(Segment)값과 관절값을 추출한다. 이때 세그먼트값에 사각형 이상의 다각형이 있는 경우 삼각형으로 만들어준다. 그리고 이를 세그먼트버퍼에 저장한다.

움직임을 기술한 BAP가 입력되면 BAP Parser를 통해 움직임 정보를 얻는다. BAP에는 프레임의 수와 각 프레임마다 어떤 BAP가 얼마나 움직이는지에 대한 정보가 있다. 따라서 이 값들을 분석하여 BAP와 관련된 상호관계표(Relation Table)를 만든다. 이 표에는 관절 데이터 표, 세그먼트 데이터 표, 유형 데이터 표, BAP 관계 데이터 표 등의 4개의 표로 구성되어 있다.

먼저 관절 데이터 표에는 관절 번호, 관절 이름, 관절 정보가 있다. 세그먼트 표에는 세그먼트 번호, 세그먼트 이름을 가지고 있다. 유형 표는 BAP의 변환 유형을 정의한 것으로 유형 번호와 유형 이름, 축 정보, 그리고 회전인지 이동 인지에 대한 정보가 들어간다. 마지막으로 BAP 관계 표는 나머지 3가지 데이터들을 연결시켜 주는 역할을 한다.

Display에서는 실제 모델을 움직이는 일을 한다. 우선 애니메이션을 위한 타이머(Timer)가 있어야 한다. BAP에는 fps에 대한 정보가 들어있으며, 이에 맞게 타이머를 설정해 놓았다. 타이머가 가동되면 해당 프레임 시간에 BAP Parser로부터 그 프레임에 맞는 BAP를 읽어온다.

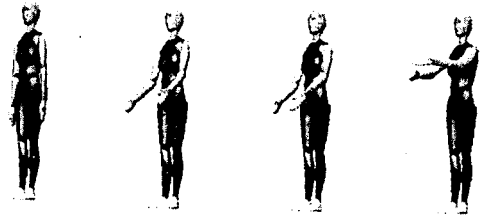
읽어온 BAP 정보는 BAP 버퍼에 임시적으로 저장된다. BAP 버퍼는 3D 모델의 세그먼트 번호와 해당 Joint 정보를 포함하고 있고, 회전 혹은 이동여부에 대한 정보와 적용할 축 정보 그리고 적용할 크기 정보를 포함하고 있다. BAP 버퍼에 있는

정보를 이용하여 결과 값을 Display 버퍼에 저장하고 저장된 내용이 화면에 보여준다.

### III. 실험결과

본 논문에서 구현한 시스템에 EPFL에서 제공한 applause.bap를 검사하였다. 이 BAP는 58프레임으로 박수치는 행동을 한다.

그림 4는 실제 구현된 모델의 움직임을 보여준다. 그림 4(a)는 첫번째 프레임으로 기본자세로 서 있다. 그림 4(b)는 30번째 프레임으로 오른쪽 어깨와 왼쪽 어깨를 기준으로 팔을 약간 올리고 있는 상태이다. 그림 4(c)는 40번째 프레임으로 박수를 치기 위해 양손을 모으는 행동을 하고 있다. 그림 4(d)는 마지막 58번째 프레임으로 박수치는 모습을 하고 있다.



(a) 0<sup>th</sup> frame (b) 30<sup>th</sup> frame (c) 40<sup>th</sup> frame (d) 58<sup>th</sup> frame

그림 4. 움직이는 Nancy 모델

### IV. 결론

본 논문에서는 MPEG-4 SNHC에서 정의한 BDP와 BAP 변수를 이용하여 가상공간에서의 애니메이션을 구현하였다. 또한 OpenGL을 이용하여 구현된 모델이 표준에 맞는 BAP에 따라 움직이도록 하였다. 그러나 현재는 단순한 척추 구조를 하고 있어, 복잡한 신체의 움직임을 표현하기는 어렵다. 따라서 추후 좀더 실제적인 사람의 움직임을 표현하기 위한 척추 구조에 대해 연구할 예정이다.

### 감사의 글

본 연구는 광주과학기술원(K-JIST) 초고속광네트워크 연구센터(UFON)를 통한 한국과학재단 우수연구센터(ERC)와 교육부 두뇌한국21(BK21) 정보기술사업단의 지원에 의한 것입니다.

### 참고문헌

- [1] ISO/IEC JTC1/SC29/WG11/MPEG98/W2068, "SNHC Verification Model 6.0," Feb. 1998
- [2] <http://ece.uwaterloo.ca/~h-anim/spec.html>
- [3] A.L. Ames, D.R. Nadeau and J.L. Moreland, *VRML 2.0 Source Book*, John Wiley, 1997
- [4] M. Woo, J. Neider, T. Davis and D. Shreiner, *OpenGL Programming Guide: Third Edition*, Addison Wesley, 1999