

# 사람 동작 표현을 위한 3D 모델 뷰어의 구현

조경은°, 조영원, 조형제  
동국대학교 컴퓨터공학과  
e-mail : nina70@cakra.dongguk.ac.kr

## Implementation of a 3D Model Viewer for Human Body Movement Representation

Kyung-Eun Cho°, Young-One Cho, Hyung-Je Cho  
Dept. of Computer Engineering, Dongguk University

### 요 약

본 논문은 사람의 동작을 생성하고 표현해야 하는 다양한 응용 분야에서 사용할 수 있는 3차원 모델 뷰어 개발에 관한 것으로, 3차원 캐릭터 애니메이션, 가상현실, 행동과학분야에서 사용 가능한 시스템이며, 액터의 움직임을 그대로 재생하고자 하는 시스템에서나 시뮬레이션된 동작을 표현하고자 할 때 사용할 수 있다. 이 시스템은 실시간으로 입력되어진 액터의 데이터를 실시간으로 출력해줄 수 있으며, 표현에 사용되어지는 모델의 크기 및 종류를 사람 각 몸 구성 성분별로 쉽게 추가, 대체할 수 있는 장점이 있다. 현재 사람의 윗몸 동작만을 입력하고 이로부터 가상 인물의 윗몸 동작으로 디스플레이하는 기능이 구현되어 있어 그 원리와 결과를 소개한다.

### 1. 서론

사람의 동작을 정확하게 표현하고자 하는 연구는 이미 여러 응용 분야에서 다양한 방법으로 이루어지고 있다. 이에 해당하는 응용분야로는 3차원 캐릭터 애니메이션, 가상현실, 행동과학 (behavior science) 등이다. 3차원 캐릭터 애니메이션에서는 사람의 동작을 표현하기 위한 대표적인 방법으로 움직임의 중요 장면(키프레임)들을 수작업에 의해 생성하고 이들 사이의 중간 장면들을 채우는 방식으로 제작되는 키프레임(key frame) 애니메이션과, 물리 법칙 등 일련의 규칙들에 의해 대상체의 움직임을 자동적 또는 반자동적으로 생성하는 동작 제어(motion control) 기술에 의한 애니메이션으로 나뉜다. 여기에 실시간 애니메이션의 문제를 해결하기 위해 사용되는 방법으로 모션캡처(motion capture) 기술을 추가할 수 있는데, 이는 광학 센서나 마그네틱 센서를 사람의 관절에 부착 한 뒤 3차원 공간상에서의 실제 배우의 움직임을 직접 기록하여 캐릭터의 움직임을 생성하는 것이다[2,3].

가상현실분야에서도 사람의 동작을 정확히 표현하고자 하는 연구가 이루어지고 있다. 최근에는 WWW 가상공간에서 상호간에 커뮤니케이션하는 대중적인 방법이 실행됨에 따라 그 실제감을 증진시키기 위하여 사람의 모습을 지니는 아바타(avatar)를 사용하는 연구도 활발하다[2,4]. 그 외에도 가상현실 분야에서는 사람의 움직임을 포착하여 아바타를 동작시키는 다양한 응용분야가 있다. 가상 수화시스템, 가상 교사 시스템, 가상 여행 시스템(Dypers), 가상 연극 무대[6] 등이 여기에 해당한다.

인간 행동의 관찰에 바탕을 둔 심리, 사회학 분야인 행동과학(behavior science)에서도 인간의 행동을 분석하기 위한 도구로서 애니메이션 기술을 이용해야 한다. 운동력, 손짓, 몸동작, 얼굴 움직임과 같은 동적인 특징들과 이로서 발생하는 비언어적 행동들을 분석하기 위해 행동과학자들은 컴퓨터 애니메이션에 특별한 관심을 가지고 있다. 그 외에도 애니메이션을 생성하는 또 다른 방법은 최근에 많이 쏟아지고 있는 애니메이션 저작 소프트웨어를 이용하는 것이다. 여기에는 VRML과 같은 텍스트 스크립트방식 또는 시각적인 기법에 의한 3D sketching과 같은 방법들이 있다[5].

이 다양한 응용분야에서 주로 사용하게 되는 기술이 모션 캡처인데 이는 몇 가지 문제점을 가지고 있다. 아주 정밀하고 매끄러운 움직임을 얻기 어려운점과 액터의 각 신체 부위의 크기가 제어될 캐릭터와 다른 경우 액터의 움직임으로부터 얻어진 동작데이터를 캐릭터의 제어에 직접 사용할 수 없게 되는 경우도 발생한다는 점이다. 특히 데이터로 만들어진 움직임에 변형을 가하거나 합성하는 등의 조작이 어렵다는 점으로 이를 해결하기 위한 동작 분석 기술이 활발하게 연구되고 있으며 이러한 동작 분석기술을 이용하여 모션캡처 데이터에 기록되어있는 액터의 신체 크기와 다른 캐릭터에 동작 데이터를 적용하기 위한 기술도 연구되고 있다[3].

본 논문에서는 칼라마르크를 부착한 사람을 추적하여 각 관절의 3차원 좌표를 출력해주는 STABIL++ 시스템[10]을 이용하여 얻은 결과로부터 모델의 동작을 재현할 수 있는 3차원 모델 뷰어를 소개한다. 본 연구에서 구현한 3차원 모델 뷰어는 실시간에 입력되어진 3차원 움직임 좌표를 실시간으로 재생시켜줄 수 있으며, 또한 액터의 신체 크기가 변경되었을 경우나 모델을 교환시에 일반 사용자가 기존의 3차원 모델링 툴을 이용하여 새로 생성한 모델을 시스템에 용이하게 추가할 수 있다. 부가적으로, 본 연구팀이 구현한 시뮬레이터[5]의 데이터를 재생시켜주는 기능이 있다. 이 시뮬레이터는 가상 현실에서의 아바타 제어와 3차원 캐릭터 애니메이션에서의 동작생성을 쉽게 제공해주는 시스템이다. 시뮬레이터가 구현되어진 기본 이론[5]을 이용하여 실시간에 입력된 데이터의 보정 및 수정을 위한 방법론도 제시한다.

2장에서는 본 연구에서 제안한 시스템의 전체적 구성을 살펴보고 3장에서는 모델을 생성하고 제어하는 방법을 설명한다. 제 4장에서는 실험한 결과를 언급한 후 결론을 맺는다.

## 2. 3차원 모델 뷰어의 시스템 구성

본 연구에서 구현된 3차원 모델 뷰어는 시뮬레이션 편집기에 의해 생성된 코드를 실세계 좌표값으로 변환한 것 또는 실시간으로 기록된 좌표 값을 입력으로 받아 모델의 움직임을 표현한다. 구현된 3차원모델 뷰어의 시스템 구성도는 그림1과 같다.

3차원 모델 생성 부분은 모델링 단계, 자료변환단계, 모델 생성 단계로 구성된다. 모델링 단계는 Lightwave 5.5에서 작업이 이루어지는 것으로 사람을 이루는 각 몸 구성 성분별로 객체를 만든다. 이 단계는 VRML 노드로의 변환이 가능한 3차원 모델링 툴이라면 모두 사용 가능하다. 여기서 만들어진 몸 구성 성분 객체들을 VRML 노드로 변환시켜주는 단계가 자료변환 단계이다. 이는 대부분의 3차원 모델링 툴에서 제공해주는 기능이다. 마지막 단계인 모델 생성 단계는 본 논문에서 구현된 프로그램에서이루어지는 작업으로 각각의 몸 구성 성분 노드들을 전체 사람을 표현해주는 scene graph에 삽입한다.

3차원 모델 뷰어는 이미 앞서서도 언급했던 것처럼, 실시간으로 포착되어지는 사람 동작의 데이터자료를 실시간으로 표현해주는 기능과 사람 동작 시뮬레이션 편집기에 의해 편집된 데이터를 3차원 가상 모델로서 표현해주는 2가지 방식을 제공한다. 실시간 데이터에 의한 실험은 STABIL++의 자료를 이용한다. STABIL++은 스테레오 카메라 입력 처리를 해주는 사람 포착 및 추적 프로그램이다. 이 시스템은 칼라마크가 부착된 위치의 3차원 좌표를 결과로 출력해준다. 두 번째 입력 방식으로는 사람의 동작을 분석하고 생성하기 위해 만들어진 사람동작 시뮬레이션 편집기에 의한 입력 데이터이다. 이 편집기에서 작성되어진 베너 코드[11]를 실세계 좌표 변환기를 통해 3차원 좌표를 생성하여 입력하면 3차원 가상모델의 동작으로 표현된다.

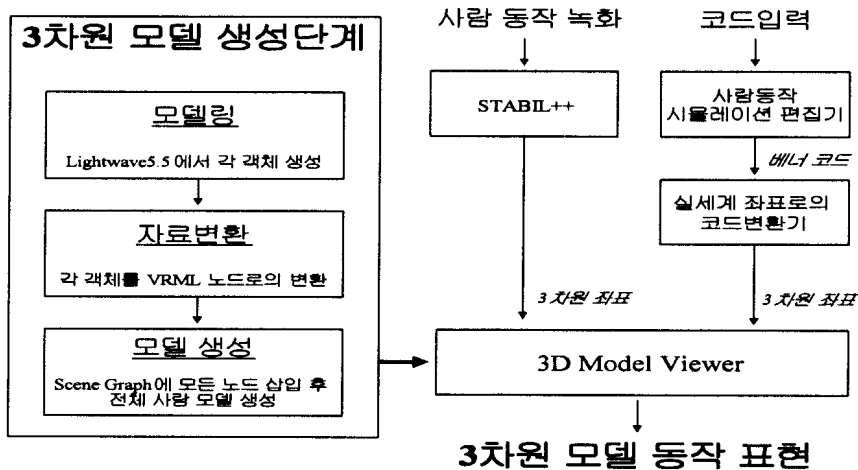


그림 1. 시스템 구성도

### 3. 3차원 모델 생성 및 모델 제어

여기서는 3차원 모델 뷰어에서 사용하게 되는 모델의 생성방법과 이 모델을 프로그램에서 제어하는 방법을 자세히 설명한다. 본 논문에서는 모델을 표현하고 제어하는 방법으로 Silicon Graphics의 Cosmo3D 라이브러리를 이용하여 시스템을 구축하였다.

#### 3.1 VRML과 Cosmo3D Library

VRML은 인터넷에서 상호 작용하는 3차원 멀티미디어 표현을 위한 파일 포맷이다. VRML은 node로 정의되고 이들 node는 그 성질이나 속성을 지정하기 위한 field와 사용자와의 인터페이스를 제공하는 event로 구성되어 있다. 예를 들어 박스 모양의 노드를 정의할 때 해당 노드의 field를 이용하여 색깔이나 크기 등을 정의하고 상자에 클릭하거나 상자에 마우스를 대는 등의 event를 주고받을 수 있다. 참고로 VRML 파일은 .wr1의 확장자를 가지며 웹브라우저에 Cosmoplayer와 같은 viewer를 설치한 후 불러서 볼 수 있다[9].

이 VRML의 노드는 Visual C++ 6.0에서 Cosmo3D Library를 이용하여 직접 다룰 수 있다. 필요한 헤더 파일을 선언하면 VRML에서 제공하는 파일 타입이나 필드 값 등을 선언하여 VRML 파일을 직접 다루는 것처럼 손쉽게 조작할 수 있다[8].

#### 3.2 모델의 생성

##### 3.2.1 사람 모델의 생성

Lightwave 5.5의 Modeler를 이용하여 사람 모델을 구성하는 객체들(objects), 즉 머리, 몸통, 위팔, 아래팔, 손등을 따로 모델링하고 이들을 각각의 VRML 파일로 변환한다. 각 객체들은 VRML에서 하나의 노드로서 그림 2와 같은 구조의 사람 모델을 구성하게 되며 트리 구조에서 위쪽 노드가 parent node, 아래쪽 노드가 child node의 관계를 형성함으로써 움직임에 있어 child node가 parent node에 종속하는 관계를 가진다. 즉, 위팔이 움직이면 아래팔도 따라서 움직이게 된다. 그림 2는 본 시스템에서 사용한 scene graph이다.

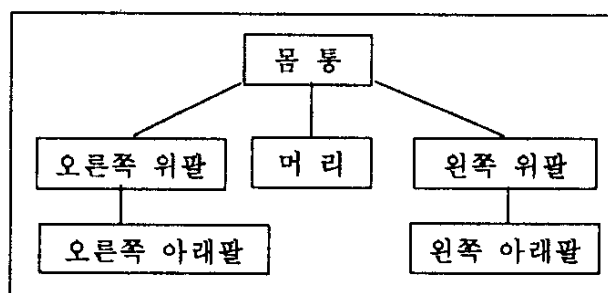


그림 2. Scene graph

### 3.2.2 각 객체의 생성

Lightwave 5.5에서 모델링할 때 원점의 좌표가 생성 객체의 기준점이 되므로 기준이 되어야 하는 점을 원점에 맞춘다. VRML에서는 노드를 회전할 때 기준점을 중심으로 회전하므로 각 객체의 기준점을 설정하는 것은 중요하다. 예를 들어 머리의 경우 목을 중심으로 회전을 하게 되므로 목에 가까운 아래쪽 면에 기준점을 두어야하고 위팔의 경우 어깨점을 중심으로 회전하고 아래팔의 경우 팔꿈치를 중심으로 회전하므로 위쪽 면에 기준을 두어야 한다. 다른 모델을 추가하거나 변경하여 사용하기 위해서는 관절이 있는 동일한 위치에 기준점을 설정하고 새로운 모델을 생성하여 교체시키면 된다.

### 3.3 모델 제어

VRML에서 이동 또는 회전을 적용하기 위해 필요한 Transform 노드의 구조는 다음과 같다.

```
Transform {  
    Translation tx ty tz  
    Rotation    rx ry rz theta  
}
```

노드의 구성요소를 필드라고 하는데 Translation 필드는 노드를 (tx, ty, tz)만큼 이동시키고 Rotation 필드는 벡터 값을 가지는 (rx, ry, rz)를 축으로 하여 노드를 theta 만큼 회전시킨다. 본 논문에서는 입력으로 각 관절의 좌표값, 즉 VRML 노드에서 특정지점의 3차원 좌표값이 들어오는데 좌표값의 차이만큼 노드를 이동시키는 건 어렵지 않지만 회전을 시킬 경우 좌표 값을 이용하여 3차원상의 회전축과 회전각도를 직접 구하는 작업을 해주어야 한다. 모델의 움직임은 이동과 회전의 순서로 적용하고 각각의 변형을 수행할 때도 parent 노드를 먼저 움직이고 child 노드의 움직임은 parent 노드의 움직임에 대해 상대적인 움직임만을 고려하여 적용하여야 한다.

#### 3.3.1 이동

입력 값으로 주어지는 좌표에 대해 최상 parent node의 이동만을 적용한다. 즉, 사람 모델에서 팔과 머리는 몸통의 child node이기 때문에 몸통의 이동에 의해 자연스럽게 몸통과 같이 이동하게 되므로 따로 고려해 줄 필요가 없다.

### 3.3.2 회전

주어지는 좌표값을 이용하여 회전축과 회전각도를 구해서 노드에 회전을 적용한다. Parent node의 회전후에 child node를 회전할 때는 parent node에 의해 움직여진 child node의 위치를 상대적으로 계산하여 적용해야 한다. 즉 그림 3에서와 같이 좌표 p1에서 p2로 회전해야 할 child node(아래팔)가 parent node(위팔)의 회전에 의해 p1'로 회전이 되었다면 p1'에서 p2로 회전되도록 하기 위하여 p1'의 좌표를 계산해야 하는데 이는 임의의 축에 대해 회전한 3차원상의 점의 좌표를 구하는 특정 알고리즘을 써서 구해야 하므로 아래에서 따로 설명하도록 한다.

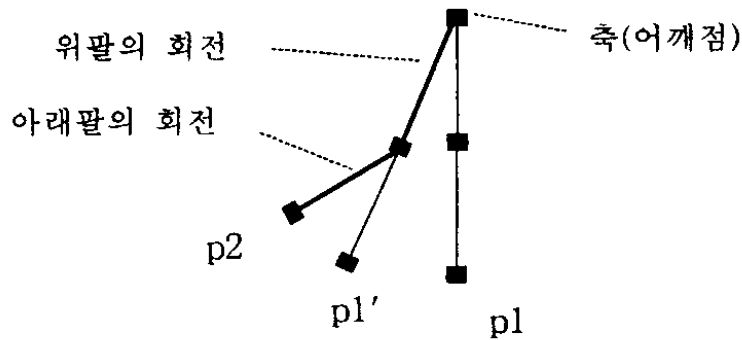


그림 3. 팔의 회전

#### ㉠ 회전축 구하기

회전축은 그림 4와 같이 기준좌표 po를 중심으로 좌표 p1, p2를 회전하였을 때 생성되는 평면에 대한 수직 벡터를 이용한다.

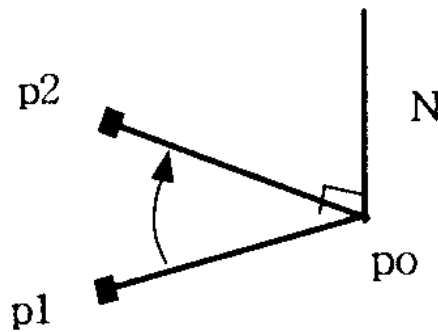


그림 4. 회전축 N

평면상의 두 벡터  $P1 = (x_1, y_1, z_1)$ ,  $P2 = (x_2, y_2, z_2)$ 에 대해 평면에 수직인 법선 벡터  $N = (x, y, z)$ 은

$$\begin{aligned}x &= y_1 * z_2 - y_2 * z_1 \\y &= z_1 * x_2 - z_2 * x_1 \\z &= x_1 * y_2 - x_2 * y_1\end{aligned}$$

이다. 참고로 벡터 P1은 각 좌표 요소 (x, y, z)에 대해 p1 - po를 수행한 결과이고 벡터 P2는 p2 - po를 수행한 결과 좌표 값이다.

### ㉠ 회전각도 구하기

회전각도  $\theta$ 는 공간벡터의 내적을 구하는 다음식을 이용한다. 벡터 P1 = (x1, y1, z1), P2 = (x2, y2, z2)가 이루는 각을  $\theta$ 라 하면

$$\cos \theta = \frac{P1 \cdot P2}{|P1| \cdot |P2|} = \frac{x_1x_2 + y_1y_2 + z_1z_2}{\sqrt{x_1^2 + y_1^2 + z_1^2} \sqrt{x_2^2 + y_2^2 + z_2^2}}$$

이므로 위 식을 변형하여

$$\theta = \arccos \left( \frac{x_1x_2 + y_1y_2 + z_1z_2}{\sqrt{x_1^2 + y_1^2 + z_1^2} \sqrt{x_2^2 + y_2^2 + z_2^2}} \right)$$

에서 회전각  $\theta$ 를 구할 수가 있다.

### ㉡ 3차원 공간상에서 임의의 축에 대해 회전한 점의 좌표 구하기

이는 그림 5와 같이 어깨점을 중심으로 아래팔이 p1에서 p2로 회전해야 할 때 위팔을 회전함으로써 child node인 아래팔이 p1에서 p1'의 좌표로 회전되어 p1'의 좌표를 구하기 위한 작업이라 할 수 있다. 3차원 상에서 임의의 축에 대해 회전된 좌표를 구하기 위해서는 회전축을 z축과 일치하게 한 후 좌표를 회전시켜야 하고 다시 회전축을 원래의 자리로 되돌리는 작업을 수행해 주어야 한다.

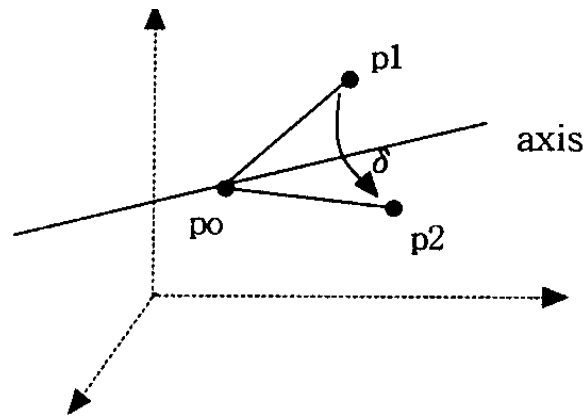


그림 5. axis를 중심축으로  $\delta$ 만큼 회전

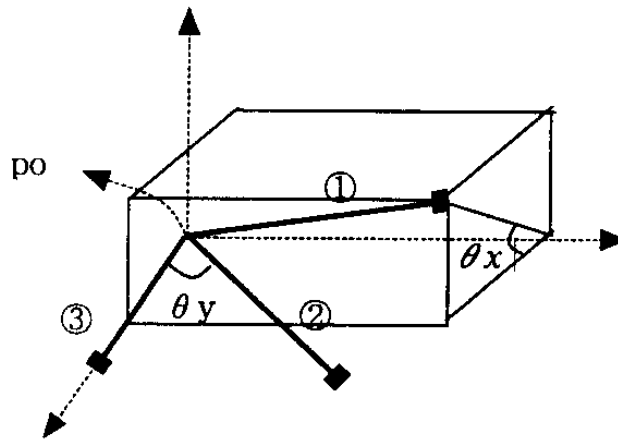


그림 6. 축(axis)의 회전

그림 5에서  $p_0$ 를 지나는 axis를 기준으로 좌표  $p_1$ 을  $p_2$ 로 회전시킬 경우 먼저  $p_0$ 를 그림 6의 ①과 같이 원점으로 이동시킨 후(T) ②와 같이 x축을 중심으로  $\theta_x$ 만큼 회전시켜 축이 xz 평면에 오게 하고(Rx) ③과 같이 y축을 중심으로  $\theta_y$ 만큼 회전하여 회전축 axis를 z축과 일치하도록 하여( $R_y$ ) z축을 중심으로  $\delta$ 만큼의 회전을 수행하였으므로 z축과 일치시켰던 축을 원래자리로 되돌리는 역과정을 수행한다. 원래 좌표  $p_1$ 에 대해 이러한 일련의 과정을 적용하여 얻은 회전된 좌표  $p_2$ 는

$$p_2 = p_1(T \cdot R_x \cdot R_y \cdot F_{xy} \cdot R_y^{-1} \cdot R_x^{-1} \cdot T^{-1})$$

이 되고 좌표  $p_1$ 에 대해 적용하는 연산들은 행렬의 형태로 이 값들은 부록 1에 있다[7]. 이상은 3차원 공간상의 VRML 모델에 대해 주어진 3차원 좌표값들을 이용하여 이동, 회전을 수행하는 방법을 보였는데 이들의 움직임은 계산상의 오차등에 의해 실제 좌표값들과 미세한 차이를 보일 수는 있다.



#### 4. 구현 및 실험결과

본 시스템은 Visual C++ 6.0환경에서 Silicon Graphics의 Cosmo3D 라이브러리를 이용하여 구현하였다. 각 객체의 모델링은 Lightwave 5.5에서 이루어졌다. 현재 시스템에서는 사람의 윗몸 동작만을 디스플레이 할 수 있는 기능이 구현되어 있다. 입력 데이터도 앉아있는 사람의 윗몸 동작만을 녹화하고, 시뮬레이터에 의해 입력되어진 데이터도 윗몸동작만으로 제한을 두었다.

다음의 그림 7은 실제 녹화장면을 나타낸 그림이고, 그림 8은 실시간 데이터를 출력한 결과를 디스플레이한 것이다. 그림 7의 실제 녹화장면은 스테레오 카메라로 촬영한 장면 중의 한쪽 카메라 장면이고 그림 8은 정면에서 바라보는 시점으로 출력되어진 결과이므로 이 두 그림은 시각적으로 약간의 차이를 보인다. 그림 9는 사람동작 시뮬레이션 편집기에 의해서 입력되어진 데이터를 실행시킨 결과이다.

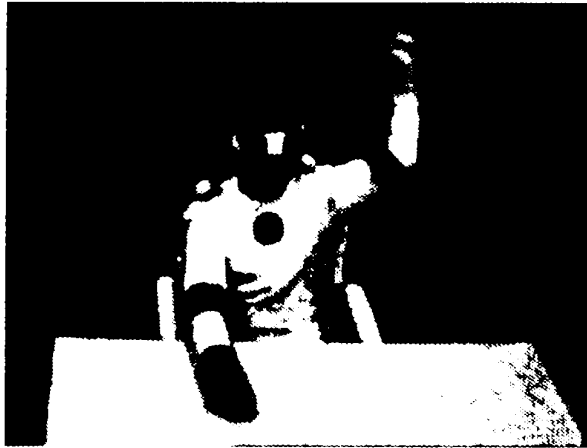


그림 7. 실제 녹화 장면

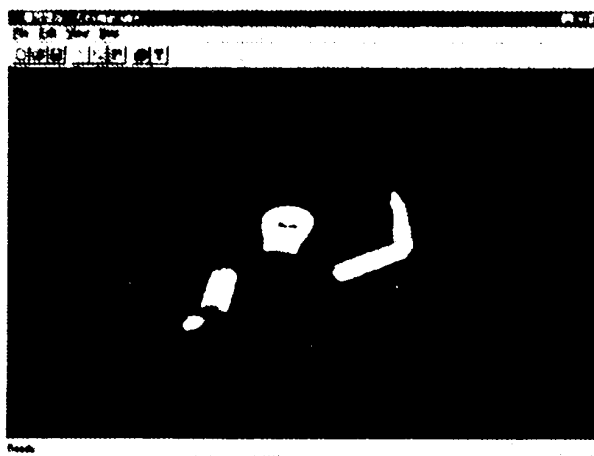


그림 8. 실시간 데이터의 출력 결과

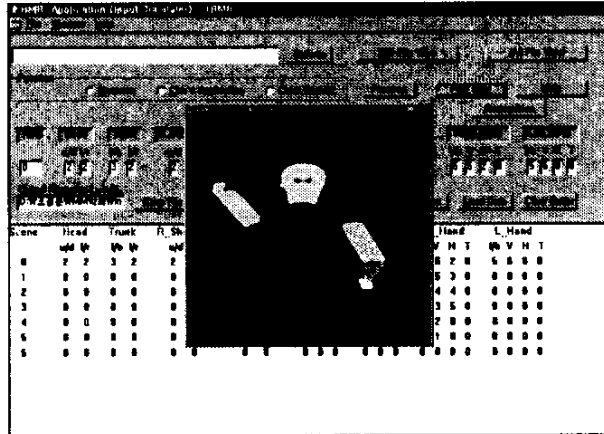


그림 9. 시뮬레이션 편집기 데이터의 출력 결과

## 5. 결론 및 향후 연구 과제

본 논문에서는 실시간 입력과 시뮬레이션된 입력을 3차원 가상 모델로 표현해주는 3차원 뷰어 시스템을 소개했다.

향후 연구과제로는 전신을 나타내는 모델로의 확장과 모델을 좀 더 정교하게 모델링해야 하는 점이다. 실시간 데이터의 입력을 재생시에는 칼라마크가 부착된 위치와 모델 사이의 오차 때문에 동작에도 약간의 오차가 발생할 수가 있는데 이것을 최소화시켜주는 알고리즘 개발이 요구된다. 또한 시뮬레이터가 구현되어진 기본 이론을 바탕으로 실시간에 입력된 데이터의 보정 및 수정을 위한 인터페이스도 추가 구현해야한다. 즉, 실시간 데이터를 베너코딩 체계로 변환하면 사용자가 쉽게 실시간 데이터를 수정할 수 있을 것으로 본다.

## 참고문헌

- [1] 강구, 김진수, "웹상에서 VRML을 이용한 수화학습 시스템", 1999년 한국정보처리학회 추계학술발표논문집 제6권 제2호, pp. MM158-MM162
- [2] 고 옥, "첨단 디지털 영상 제작 기술", 1999년 정보과학회지 제 17권 제 2호, pp.4-17
- [3] 김응순, 김영수, "3차원 캐릭터 애니메이션 기술 동향", 1999년 정보과학회지 17권 2호, pp. 48-59
- [4] 김정각, 박찬중, 김동현, 조맹섭, "가상세계를 위한 분신의 행동양식과 상호작용", 1997년도 한국정보과학회 가을 학술발표논문집 Vol. 24, No.2, pp.363 -366
- [5] 조경은, 조영원, 조형제, "유한 오토마타를 이용한 사람 동작 시뮬레이션 편집기 개발에 관한연구", 1999년 한국정보처리학회 추계학술발표논문집 제6권 제2호, pp. MM135-MM139
- [6] A. Azarbayejani, C. Wren, A. Pentland, "Real-time 3-D Tracking of the Human Body", IMAGE'COM 96, 1996, MIT Technical Report No. 374

- [7] D.F. Rogers, J.A. Adams, "Mathematical Elements for Computer Graphics", 2nd Ed., Macgraw-Hill, 1989, pp. 101-127
- [8] George Eckel, "Cosmo 3D(TM) Programmer's Guide", Silicon Graphics, 1998
- [9] International Standard ISO/IEC 14772-1:1997, "The Virtual Reality Modeling Language"
- [10] O. Munkelt, C. Ridder, D. Hansel, W. Hafner, "A model driven 3D image interpretation system applied to person detection in video images", 14'th ICPR 1998, pp. 70-73
- [11] S.Frey, H.P. Hirsbrunner, J. Pool & W. Daw, "Das Berner System zur Untersuchung nonverbaler Interaktion", P. Winkler Edition. Methoden der Analyse von Face-to-Face -Situationen, Stuttgart, Metzler, 1981, pp. 203-236

### 부록 1. 3차원 변환 행렬

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_0 & -y_0 & -z_0 & 1 \end{bmatrix}$$

$$Rx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_x & \sin \theta_x & 0 \\ 0 & -\sin \theta_x & -\cos \theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Ry = \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta_y & 0 & \cos \theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Fxy = \begin{bmatrix} \cos \delta & \sin \delta & 0 & 0 \\ -\sin \delta & \cos \delta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$