

모션캡처 시스템을 위한 추상레이어의 설계

이 회 만*

The Design of Abstract Layer for Motion Capture System

Hee-man Lee *

요 약

본 논문은 다양한 모션캡처 하드웨어 장비에 독립적인 응용 프로그램을 작성할 수 있도록 단일화된 추상 레이어를 설계하고 구현 하였다. 설계한 추상레이어의 기능을 테스트하고 검증하기 위하여 광학식 모션캡처 시스템과 기계식 모션 캡처 시스템을 각각 AOA 파일과 BVH 파일을 이용하여 에뮬레이션 하였다. 응용 프로그램에서 추상레이어 함수의 호출에 의해 광학식 및 기계식 모션캡처 시스템을 각각 구동하고 시간에 따른 프레임 데이터를 Direct3D를 이용하여 애니메이션 하였다.

Abstract

In this paper, the abstract layer for motion capture system is designed and implemented to meet the various hardware and different capturing method. The abstract layer can offer the unified programming by providing device independent API(Application Programming Interface). The device drivers of the optical system and mechanical system are emulated to verify the designed abstract layer. The optical system employs the AOA file while the mechanical system uses BVH file. An application program is written to call the abstract layer functions to drive both optical and mechanical drivers and receive the frame data, simulated motion data, that are displayed sequentially on the computer screen by utilizing Direct3D.

▶ Keyword : 모션캡처(Motion Capture), 추상레이어(Abstract Layer), 3D 애니메이션(3D Animation), 광학식 모션캡처(Optical Motion Capture) 기계식 모션캡처(Mechanical Motion Capture),

• 제1저자 : 이회만
• 접수일 : 2007.1.24, 심사일 : 2007.2.5, 심사완료일 : 2007. 3.12.
* 서원대학교 컴퓨터공학부 교수

I. 서론

모션캡처는 컴퓨터 그래픽 애니메이션을 하는데 현재 가장 주목을 받고 있는 기법중 하나이다. 실제 인물과 동물, 또는 그 외에 동작하는 물체의 움직임을 카메라 또는 장비 등을 사용하여 동작을 캡처한 후 데이터 가공을 통하여 컴퓨터에서 애니메이션에 이용하는 기법이다. 이 모션캡처 기법의 특징은 실시간으로 애니메이션이 가능하고, 키 프레임 을 통하여 애니메이션 하던 과거의 방법보다 빠르게 애니메이션 할 수 있다. 모션캡처는 애니메이션의 대상이 되는 가상캐릭터(Virtual Character)와 실제연기자(Performer)가 있다. 실제연기자 몸의 동작을 알아내어 가상 캐릭터(3D 모델)가 같은 동작으로 애니메이션 되도록 하는 것이 모션캡처 기술이다. 그러므로 모션캡처는 애니메이션이 그려 진다기 보다는 행위(Perform)되어지는 방법이다. 즉, 컴퓨터가 생산한 모델의 움직임을 애니메이터가 대략적으로 규정짓는 것 대신에 연기자의 실제적 행위로부터 얻어진 데이터에 의해 3D 모델이 애니메이션 되는 것이다. 이러한 방법은 이전에 볼 수 있었던 그 어떤 방법보다도 자연스러운 움직임을 만들어낼 수 있을 뿐 아니라, 모델로 하여금 몸동작의 언어적 표현과 그 밖의 실제 공연에서만 볼 수 있었던 특징들을 표현할 수 있게 한다.

모션캡처는 1차적으로 중요 동작을 수집하는데 사용된다. 즉 수집한 모션의 데이터는 기본동작 애니메이션 트랙을 만드는데 사용되고, 추후 다른 애니메이션 기법을 사용하여 보강하는 것이 일반적 수순이다. 모션캡처 기법에 근기한 애니메이션에 추가되는 2차적 작업은 얼굴표정 등에 관한 세부 애니메이션은 1차 동작의 기본 트랙 위에 추가된다. 모션캡처는 군사, 의료 목적으로 꾸준히 활용되어 왔으며 영화 산업을 중심으로 광범위하게 퍼져 나갔다. 특히 영화의 특수효과 제작에 있어서 모션캡처의 역할은 점차 증대되고 있으며, 이러한 추세는 당분간 계속될 것으로 보인다 [1][2][3].

본 논문에서는 많은 하드웨어 장비나 구성방식에 상관없이 모션 데이터를 모션캡처 시스템으로 부터 받고 이를 애니메이션 할 수 있도록 하는 추상 레이어(Abstract Layer) 설계에 관한 것이다. 추상 레이어는 운영체제 설계자들이 많이 사용하는 방법으로 프로세서와 그래픽 카드 등의 다양한 하드웨어에서 운영체제를 안정적으로 포팅할 수 있도록 하는데 사용되고 있다. 본 논문은 3D 프로그래머가

하드웨어 장비에 독립적인 프로그램을 작성할 수 있도록 하는 추상 레이어를 설계하였다.

II. 모션캡처 장비

2.1. 기계식 방식 (Mechanical System)

기계식 모션캡처 시스템은 연기자의 관절 움직임을 측정하기 위한 전위차계와 슬라이더의 복합체로 구성되어 있다. 이 방식은 액터에게 뼈와 같이 관절이 있는 철제 장치를 몸에 입고 캡처하는 방식으로서 액터의 각 관절에 해당하는 곳에 센서(가변저항)를 붙여 기계장치를 입은 액터의 팔이나 다리처럼 움직임이 있는 부위에 움직임이 있을 때 얼마나 움직였는지를 측정 할 수 있게 해준다.

기계식 시스템의 장점은 기구적인 방법을 사용하기 때문에 외부의 영향(빛, 급속체, 자성체 등)을 받지 않으므로 실내 및 실외 어디에서든지 사용이 간편하고, 단순히 사람이 입는 것만으로 모든 세팅이 끝난다. 그래서 셋업 과정이 거의 필요 없으며 상당히 높은 샘플링 빈도로 모션 데이터를 획득할 수 있다. 무엇보다도 장비의 가격이 가장 저렴하다.

기계식 시스템의 단점은 매우 부담이 되는 기계장치를 연기자의 몸에 부착해야 하므로 자연스러운 동작을 연출할 수 없고, 기계 장치가 연기자의 각 관절에 얼마나 정확하게 위치했는지에 따라 정확도가 달라진다.

2.2. 자기식 방식 (Magnetic System)

자기식 모션캡처 시스템은 액터의 각 관절 부위에 자기장을 계측할 수 있는 수신기(Sensor)를 부착하고 중앙에 위치한 송신기를 사용하여 자기장 발생장치 근처에서 연기자가 움직일 때 각 센서에서 측정하는 자기장의 변화를 다시 공간적인 변화량으로 계산하여 움직임을 측정하는 방식이다. 각 센서와 자기장 발생장치 및 본체는 케이블로 연결되어 있다.

자기식 시스템의 장점은 셋업이 쉽고, 기계식 방식에 비해서 자연스러운 동작을 획득할 수 있으며, 광학식에 비해 가격이 저렴하다[4].

자기식 시스템의 단점은 센서를 연기자의 몸에 부착할 본체와 연결되는 케이블로 인하여 연기자의 동작에 제한을 주고 이것으로 인해 복잡하고 빠른 움직임을 자연스러운 모션을 캡처하는 것이 불가능하다. 이러한 단점을 보완하기 위

해서 최근에는 무선시스템을 개발하였으나 송신기를 부착하여야 하기 때문에 자연스러운 모션을 받기에는 여전히 무리가 있다. 또한 자기식 시스템은 자기장을 이용하는 만큼 외부의 영향을 가장 심하게 받는다[5].

2.3. 광학식 방식 (Optical System)

광학식 시스템은 광원, 카메라, 반사물체(마커)를 이용하여 3차원 공간에서 마커의 위치를 결정한다. 즉, 이들 시스템은 카메라로부터 들어온 정보들을 각 카메라의 2차원 위치 데이터를 제공하면 모션캡처 소프트웨어가 3D 데이터로 계산한다[6].

광학식 시스템의 장점은 자기식 모션캡처 시스템과 달리 금속에 의해 영향을 받지 않으며, 액터에게 부착되는 마커의 크기가 작고 케이블이나 다른 것에 고정이 되어 있지 않아 동작에 제한이 없어 자유스럽고 자연스러운 동작을 캡처 받을 수 있다. 뿐만 아니라 2인 이상의 액터의 모션캡처를 할 때에도 각자 마커만 부착하면 여러 명의 액터의 모션캡처가 가능하기 때문에 편리하다.

광학식 시스템의 단점은 액터의 몸에 부착하는 마커가 카메라로부터 보이지 않을 때에는 데이터의 손실의 우려가 있다. 광학식 시스템은 카메라로부터 나가는 빛에 의한 마커의 반사된 정보로 데이터를 얻기 때문에 햇빛이나 자외선이 있으면 그러한 빛들에 민감하여 데이터의 오류가 발생할 우려가 있다. 또 다른 단점으로는 비용이 다른 시스템에 비해 월등히 높으며 가격은 카메라의 속도와 대수, 해상도에 비례하며 처리 소프트웨어의 종류에 따라 많은 차이를 보인다.

2.4. 음향식 방식 (Acoustic System)

음향식 시스템은 다수의 초음파 발생장치와 3개의 수신 장치로 구성된다. 액터의 각 관절에 부착된 초음파 발생장치가 초음파를 발생하면, 발생한 초음파가 수신 장치에 수신되기까지 걸린 시간을 이용해, 그때의 소리 속도를 이용해 발생장치에서 수신 장치까지의 거리를 계산한다. 각 전송장치의 3차원 공간상의 위치는 3개의 수신 장치에서 각각 계산된 값을 이용한 삼각 측량원리에 의하여 구할 수 있다.

음향식 시스템의 장점은 광학식 모션캡처 시스템에서 일반적으로 발생하는 차폐(Occlusion)문제나, 자기식 모션캡처 시스템에서 발생하는 금속물체에 의한 간섭 등의 문제가 발생하지 않는다는 장점이 있다. 또한 이 시스템은 위치 측정에 필요한 계산량이 적어 실시간 처리가 가능하며, 다른 시스템에 비해 값이 저렴하다.

음향식 시스템의 단점에는 시스템의 특성상 각 초음파 발생장치의 위치를 순차적으로 측정할 수밖에 없어, 원하는 순간 액터의 몸에 부착된 모든 초음파 발생장치의 위치를 동시에 측정하기가 힘들다는 것이다. 이 문제점은 샘플링 빈도(Sampling Rate)를 저하시키거나, 동시에 사용할 수 있는 초음파 발생장치의 수를 제한시키는 요인이 되어 정밀한 동작을 캡처하는데 어려움을 주기도 한다.

2.5. 소프트웨어 방식

이 방식은 다른 모션캡처 시스템과 다르게 소프트웨어 기반이라는 것이 가장 큰 특징이다. 액터가 장비를 입거나 센서를 붙이지 않고 디지털 캠코더를 통하여 받은 비디오 영상(AMI, MPEG 형식)을 통하여 캐릭터의 모션 데이터를 추출할 수 있다.

소프트웨어 시스템의 장점은 다른 시스템과 다르게 공간이 필요하거나 장비가 필요 없이 컴퓨터 시스템만 있으면 된다는 것이고, 그에 따라 가격이 저렴하다. 또한 디지털 캠코더로 통하여 받은 비디오 영상을 이용하여 모션캡처를 하기 때문에 사람뿐만 아니라 동물들의 움직임 동작을 쉽게 캡처할 수 있다.

소프트웨어 시스템의 단점은 다른 모션캡처 시스템을 이용하는 만큼의 정교한 데이터를 얻을 수 없다는 것이고, 비디오 영상을 이용해서 모션을 캡처하기 때문에 실시간 모션캡처가 되지 않는다[7].

III. 추상 레이어

전산학에서의 추상화란 지엽적이고 세부적인 사항들을 줄이고 보이지 않도록 하는 기법을 말한다. 이는 수학에서의 추상화의 개념과 유사하다. 추상 레이어는 특정 기능을 구현하기 위해 하부의 상세한 사항들을 감추는 하나의 방법이다. 추상화 레이어의 개념을 사용하는 잘 알려진 소프트웨어 모델은 컴퓨터 네트워크 프로토콜의 OSI 7 레이어라고 할 수 있다. 또한 OpenGL의 그래픽스 라이브러리, Unix나 Linux 또는 MSDOS 등의 운영체제에서 사용하는 바이트 스트림 I/O 모델들도 잘 알려진 추상 레이어를 사용하는 소프트웨어 모델이다[8].

UNIX 운영체제에서 대부분의 입출력 오퍼레이션은 장치로부터 읽어 오거나 또는 저장하는 바이트의 스트림으로 간주된다. 이 바이트 스트림의 모델은 파일의 I/O, 소켓에서의 I/O 등의 디바이스 독립적으로 작용할 수 있도록 사용된다.

응용 프로그램 레벨에서 장치의 데이터를 읽거나 쓰기를 할 때는 해당 장치를 열기 위해서는 함수를 호출한다. 다른 장치의 경우도 실제 물리적인 장치이거나 또는 가상의 장치인 경우에도 같은 함수를 호출하여 데이터를 저장하거나 읽어 올수 있다. 이는 각 장치의 물리적인 세부 특징이 운영체제에 의하여 은닉되어 있기 때문이다. 즉 운영체제는 응용프로그램에 추상화 레이어를 제공함으로써 프로그래머가 편리하고 일관성 있게 데이터를 저장하고 불러 올수 있도록 한다. 추상화 레이어는 일반적으로 계층적 구조를 가지고 있다. 각 층은 하부 층의 구체적인 구현을 은닉한다.

본 논문의 추상 레이어는 하드웨어 추상 레이어(HAL)의 일종이다. 하드웨어 추상 레이어는 컴퓨터의 물리적인 장치와 컴퓨터에서 실행되는 소프트웨어 사이에 존재하는 소프트웨어이다. 하드웨어 추상 레이어는 각각 다른 물리적인 하드웨어를 제어하고 데이터를 조작하는 기능을 운영체제의 커널이나 응용 프로그램에 은닉하는 기능을 제공한다. 그림으로 해서 많은 하드웨어 장비나 구성 방식에 상관없이 일관된 인터페이스를 제공한다. 하드웨어 추상 레이어는 또한 고급 수준의 언어로 저수준의 하드웨어를 직접 제어할 수 있도록 한다. 윈도우즈 운영체제에서는 하드웨어 추상 레이어를 커널에 포함시키고 있다. 이는 운영체제가 다양한 하드웨어에서 운용될 수 있도록 이동성(portability)을 높여준다. 대부분의 코드상의 변화가 없이도 다양한 프로세서나 그래픽 카드 등에서 잘 수행 될 수 있다.

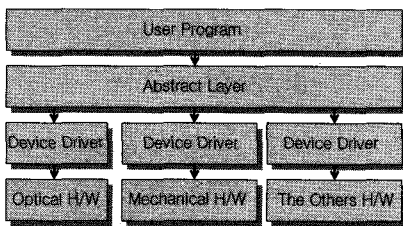


그림 1. 추상 레이어의 계층구조
Fig 1. Hierarchy of Abstract Layer

모션캡처의 다양한 장비와 파일형식은 이를 이용하는 사람들에게 많은 어려움을 주고 있어 통일되고 일관성 있는 방법이 연구 되고 있다[9]. 본 논문에서는 많은 모션캡처 하드웨어 장비나 구성방식에 상관없이 모션 데이터를 일관성 있게 읽고 처리하고 애니메이션 할 수 있도록 하는 추상 레이어(Abstract Layer)를 설계하였다. <그림 1>은 모션캡처를 위한 추상 레이어의 계층구조를 보인 것이다. 그림에서 알 수 있듯이 실제 하드웨어 레이어가 은닉되어 있

어 사용자 프로그램은 사용하는 하드웨어장비와 무관하게 추상 레이어를 통하여 모션캡처 장비를 제어할 수 있다. 추상 레이어는 각각 다른 하드웨어의 드라이버 상단에 위치하며 하위 층을 은닉한다. 모션캡처 시스템을 사용하는 사용자 프로그램은 추상 레이어의 인터페이스만을 이용한다.

추상 레이어는 C++의 추상 클래스로 구현할 수 있다. 추상 클래스는 <그림 2>과 같이 기본적인 기능만을 가지도록 설계하였다.

```

typedef struct
{
    //member variable
    MoCapClass* SensorLocation;//Data Read/Write
    Callback DrawFunction;//user call back function

    //member funtions
    virtual int GetDeviceCaps(MoCapCAPS &caps)=0;
    virtual int SetSensingHiearchy();
    virtual int SetUserDrawingCallback(Callback fname)=0;
    virtual int StartCaturingRec();
    virtual int StopRec();
    virtual int CreateBuffer();
    virtual MoCapClass* GetBuffer();
    virtual int InitSystem();
} IMoCapDevice;
    
```

그림 2. 추상 레이어를 위한 추상 클래스
Fig 2. Abstract Class for Abstract Layer

광학방식의 모션캡처 시스템은 마커의 위치를 캡처하는 모션캡처 시스템이며 기계식 모션 캡처 시스템은 계층적 구조를 갖는 관절의 결합 각도를 캡처하는 모션캡처 시스템이다. 서로 다른 이들 두 방식을 하나의 추상 레이어에서 통합 관리하기위하여 마커나 관절 센서의 위치를 계층적 구조로 일반화하고 각 프레임에서의 위치는 광학방식처럼 위치정보를 표시하도록 한다.

추상클래스의 멤버 변수는 두 개다. 즉 마커 및 관절의 계층적 구조와 모션 데이터의 저장 및 관리를 MoCapClass 객체에 저장하도록 하며, 하드웨어의 각 프레임 데이터가 얻어질 때마다 사용자 함수를 호출해 줄 수 있도록 DrawFunction변수에 콜백함수 포인터를 보관한다. 멤버함수는 기본적인 기능을 수행하도록 간략히 설계되었다. GetDeviceCaps()함수는 디바이스의 성과와 관련된 정보를 알려주는 함수이며 SetSensingHiearchy() 함수

는 마커의 계층적 구조 정보를 초기화할 때 설정하는 함수이며 SetUserDrawingCallback() 함수는 사용자 함수의 프레임 단위 콜백함수를 설정하는 함수이다. StartCaturingRec()함수는 모션캡처 시스템을 기동하여 모션 데이터를 얻어오는 함수이며 StopRec()함수는 모션캡처 시스템을 정지시키는 함수이다. CreateBuffer()함수는 멤버 변수인 SensorLocation에 메모리를 할당하는 함수이며 GetBuffer()함수는 SensorLocation 멤버변수의 주소를 알려주는 함수이고 InitSystem()함수는 하드웨어의 초기화 함수이다.

모션을 캡처한 데이터를 저장하고 관리하는 MoCapClass 객체의 단순화 시킨 클래스 선언은 (그림 3)과 같다.

```
class MoCapClass
{
public:
    BVHNode    m_bvhNode;
    MotionNode m_motionNode;
    int        m_dataPerFrame;
    int        m_numMarkers;
    float      m_frameRate;
public:
    MoCapClass(){/*초기화... */};
};
```

그림 3. 모션데이터 관리 클래스
Fig 3. Class for Motion Data

마커 또는 센서의 계층적 구조는 BVHNode에 저장하며 모션의 데이터는 MotionNode에 저장한다. 또한 각 프레임마다 필요한 데이터의 개수와 마커 또는 마커의 수 및 프레임 캡처 주기를 저장할 수 있도록 설계되어 있다.

IV. 모션캡처 시스템 에뮬레이션

본 논문에서는 기계식 모션캡처시스템과 광학식 모션캡처 시스템의 하드웨어를 에뮬레이션하여 추상 레이어의 기능을 검증하였다.

기계식 모션캡처 시스템을 에뮬레이션하기 위하여 기계식 모션캡처 데이터를 저장한 BVH 파일을 일정 주기 간격으로 읽어 들이도록 하는 가상의 디바이스 드라이버 프로그램

을 작성하였다. 기계식 가상 디바이스 드라이버는 추상 레이어의 함수를 구체적으로 구현하는 것이다. BVH 파일은 모션캡처 전문 회사인 Biovision사에 의해 개발된 파일형식이다. BVH 파일에서는 센서의 위치가 관절을 대표하며 관절의 구조는 계층적 구조로 되어 있다. 각 관절의 오프셋을 기준으로 프레임 별 각 관절의 3개축에 대한 회전각도(오일러 각도)가 저장되어 있다[10]. 각 관절의 부모 회전 각도 데이터는 부모노드 자신이 아닌 자식 노드에 영향을 준다. 자식 노드의 오프셋 정보와 3개축의 오일러 각도를 이용하여 자식 노드의 새로운 좌표를 구한다. 자식 노드의 좌표는 그 아래 손자 노드에 또한 영향을 준다. 다음은 좌표를 계산하는 방법이다.

회전행렬 Rr 은 BVH 파일에서 제공하는 각축의 회전행렬. Ry, Rx, Rz 의 매트릭스 곱으로 계산한다.

$$Rr = RyRxRz \dots\dots\dots (4.1)$$

새로운 오프셋을 구하기 위한 월드행렬 Nw 는 수식 (4.1)에서 계산한 매트릭스와 부모의 월드 매트릭스 Pm 의 매트릭스 곱이다. Pm 매트릭스는 계층적 구조에 의해 자식과 손자에게 계속 누적되어 계산된다.

$$Nw = RrPm \dots\dots\dots (4.2)$$

그러므로 새로운 오프셋 좌표의 벡터 매트릭스 No 는 BVH에서 제공하는 초기 오프셋 좌표의 벡터 매트릭스 Os 와 수식 (4.2)에서 구한 Nw 매트릭스의 곱이다.

$$No = OsNm \dots\dots\dots (4.3)$$

최종적으로 새로운 노드의 위치 벡터 매트릭스 는 Nc 는 수식(4.3)에서 구한 No 매트릭스와 계층적 구조상의 부모 노드 위치 벡터 매트릭스 Pc 매트릭스와의 합이 된다.

$$Nc = No + Pc \dots\dots\dots (4.4)$$

광학식 모션캡처 시스템을 에뮬레이션하기 위하여 광학식 모션캡처 데이터를 저장한 AOA 파일을 일정 주기 간격으로 읽어 들이도록 가상의 디바이스 프로그램을 작성하였다. 광학식 가상 디바이스 드라이버는 추상 레이어 클래스를 상속 받으며 버추얼 함수를 구체적으로 구현한다. AOA 파일은 Adaptive Optics사의 모션캡처 파일 포맷이며, 이 포맷은 마커를 기준으로 하는 모션캡처 파일로 각 프레임별 마커의 위치가 기록 되어 있다.

V. 실험결과

모션캡처 시스템을 위한 추상 레이어를 구체적으로 구현하고 실험하기 위하여 <그림 4>와 같이 구성하였으며 윈도우 XP 운영체제에서 구현하였다. 추상 레이어 및 가상의 드라이버는 마이크로소프트사의 COM 모델을 사용하여 각각 독립적으로 제작하였다. <그림 4>에서 보인바와 같이 가상의 드라이버는 기계식 장비를 에뮬레이션 하기위해서 BVH 파일을 사용하며, 광학식 장비를 에뮬레이션 하기 위해서 AOA 파일을 사용한다.

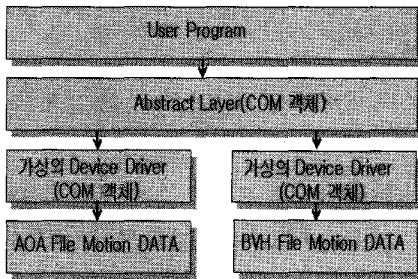


그림 4. 추상 레이어의 에뮬레이션
Fig 4. Emulation for Abstract Layer

광학방식 모션캡처 시스템의 마커 위치 및 기계식 모션캡처 시스템의 관절위치는 모두 계층적 구조로 기술되며 BVH 파일에 저장되고 SetSensingHierarchy() 함수를 통하여 추상 레이어에 설정된다. <그림 5>는 BVH 파일의 일부분을 보인 것이며, <그림 6>은 본 실험에서 사용한 마커 및 관절의 계층구조를 보인 것이다.

```

HIERARCHY
ROOT Hips
{
  OFFSET 0.000000 0.000000 0.000000
  CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
  JOINT LeftHip
  {
    OFFSET 5.000000 0.000000 0.000000
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT LeftKnee
    {
      OFFSET 0.000000 -17.500000 0.000000
      CHANNELS 3 Zrotation Xrotation Yrotation
      JOINT LeftAnkle
      {
        OFFSET 0.000000 -15.500000 0.000000
        CHANNELS 3 Zrotation Xrotation Yrotation
        End Site
        {
          OFFSET 0.000000 -3.500000 -1.500000
        }
      }
    }
  }
}
    
```

그림 5. BVH 파일의 일부분
Fig 5. A Part of BVH File

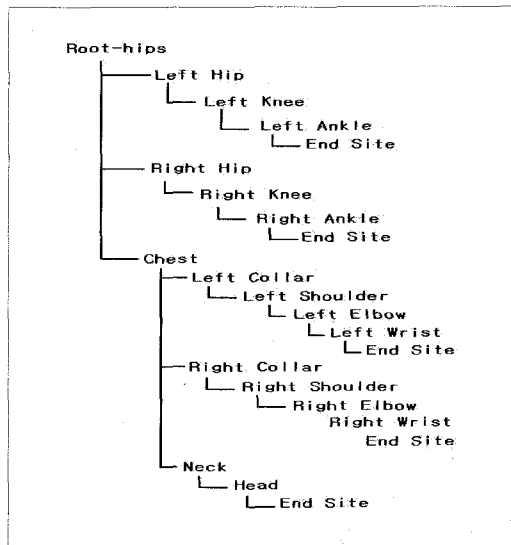


그림 6. 마커/관절의 계층구조
Fig 6. Markers/Articulations Hierarchy

광학식 및 기계식 모두 일관성 있는 데이터 양식을 유지하기 위하여 절대좌표를 사용한다. <그림 7>은 BVH에 저장된 관절의 초기위치를 계산한 것으로 상대좌표로부터 계산하여 얻은 각 관절의 절대 좌표를 보인 것이다.

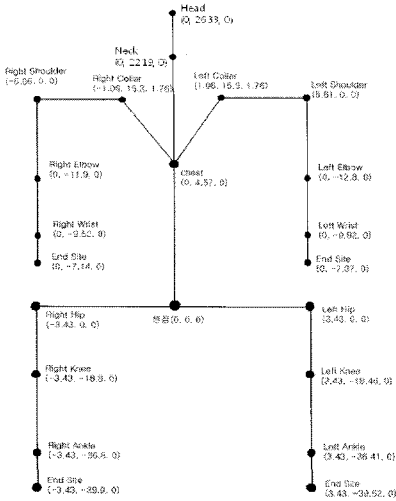


그림 7. 관절의 절대좌표

Fig. 7. Absolute Coordinate of Articulations

BVH에 정의된 계층구조와 모션데이터를 이용하여 각 관절의 절대 위치를 계산하는 재귀 알고리즘을 의사 코드로 표현하면 <그림 8>과 같다.

```

RecursiveCalc(input parameters)
{
    if nodeType is ROOT
    {
        Read Offset Data from Model Hierarchy
        such as, offset_x, offset_y, offset_z
        Read Position Data from Motion Data Part
        such as, acc_x, acc_y, acc_z
        Find New Coordinate
        new_x.y.z = offset_x.y.z * Parent Transform Matrix
        new_x.y.z = new_x.y.z + acc_x.y.z
        Read Euler Anis from Motion Data Part
        such as angle_z, angle_x, angle_y
        Find Local Transform Matrix
        such as, NewMatrix = RotationY * RotationX * RotationZ
        and NewMatrix= NewMatrix *Parent Transform Matrix
    }
    if nodeType is JOINT
    {
        Read Offset Data from Model Hierarchy
        such as, offset_x, offset_y, offset_z
        Read Position Data from previous Data
        such as, acc_x, acc_y, acc_z
        Find New Coordinate
        new_x.y.z = offset_x.y.z * Parent Transform Matrix
        new_x.y.z = new_x.y.z + acc_x.y.z
        Read Euler Angle from Motion Data Part
        such as angle_z, angle_x, angle_y
        Find Local Transform Matrix
        such as, NewMatrix = RotationY * RotationX * RotationZ
        and NewMatrix= NewMatrix *Parent Transform Matrix
    }
    if nodeType is ENDSITE
    {
        Read Offset Data from Model Hierarchy
        such as, offset_x, offset_y, offset_z
        Find New Coordinate
        new_x.y.z = offset_x.y.z * Parent Transform Matrix
        new_x.y.z = new_x.y.z + acc_x.y.z
    }
    Call RecursiveCalc with Child node, new_x.y.z and NewMatrix
    Call RecursiveCalc with Sibling node, acc_x.y.z ParentMatrix
}
    
```

그림 8. BVH 좌표계산 재귀 알고리즘
Fig 8. BVH Recursive Algorithm for Coordinate

광학방식의 경우에는 마커위치의 절대 좌표를 사용하여 계층적 트리구조의 마커 정보와 함께 사용하였다.<그림 9>.

AOA file written by Seovon Pro Motion
frames = 1227 nummarkers = 23 hz = 30

-0.004	36.508	-0.087
4.990	36.508	-0.323
5.042	19.008	-0.276
5.087	3.508	-0.241
5.033	-0.000	-1.721
-4.999	36.508	0.148
-4.996	19.008	0.197
-4.994	3.508	0.239
-5.064	0.002	-1.244
-0.051	39.508	-1.086
0.054	58.502	0.017
6.547	58.484	-0.289
6.512	46.484	-0.321
6.484	36.984	-0.344
6.464	29.984	-0.343
0.054	58.502	0.017
-6.439	58.520	0.323
-6.472	46.520	0.331
-6.496	37.021	0.378
-6.518	30.021	0.344

그림 9. AOA 파일의 일부분
Fig 9. A Part of AOA File

<그림 10>는 응용 프로그램에서 추상 레이어의 함수를 호출하여 가상의 디바이스 드라이버를 구동하고, 가상의 드라이버에서 전해주는 모션데이터를 Direct3D를 이용하여 애니메이션을 실시한 결과 화면이다.

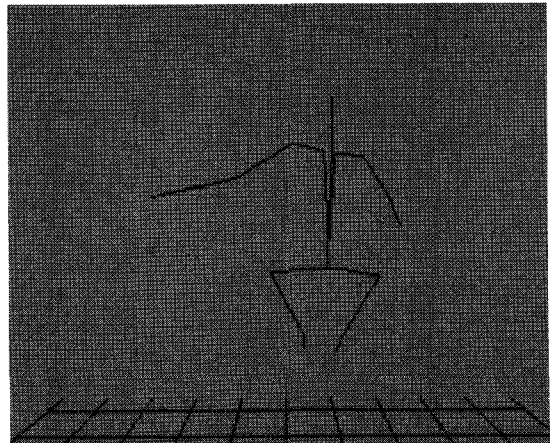


그림 10. 추상 레이어를 이용한 응용프로그램의 애니메이션 결과
Fig 10. Animation Result of Application Program using Abstract Layer

VI. 결론

본 논문에서는 모션캡처 하드웨어 장비에 독립적인 프로그램 작성할 수 있도록 하는 추상 레이어를 설계하고 구현 하였다. 추상 레이어는 운영체제에서 많이 사용하는 방법으로 프로세서와 그래픽 카드 등의 다양한 하드웨어에서 운영체제를 안정적으로 포팅할 수 있도록 하는데 사용되고 있다. 본 논문의 추상 레이어는 하드웨어 추상 레이어(HAL)이며 하드웨어 추상 레이어는 고급 수준의 언어로 저수준의 하드웨어를 직접 제어할 수 있도록 한다. 본 논문에서는 광학식 모션캡처 시스템과 기계식 모션 캡처 시스템을 각각 AOA 파일과 BVH 파일을 이용하여 애니메이션 하여 추상 레이어의 기능을 실험하였다. 향후 본 연구에서 개발한 추상 레이어를 이용하여 특정 목적에 사용되는 광학식 또는 기계식 모션 캡처시스템을 설계하고 구현할 예정이다.

참고문헌

- [1] 최홍진, "모션캡처 기술 및 특허동향 보고서," 한국특허정보원, 2004.
- [2] 이계희, "모션 캡처의 과거, 현재, 그리고 미래," 정보과학회지, 제21권, 제7호, pp.24-29, 2003.
- [3] 이인호, 박찬중, "모션 캡처 기술의 현황과 응용분야," 한국멀티미디어학회지, 제3권, 제1호, pp.38-46, 1999.
- [4] "http://www.parhae.co.kr/," (주)발해게이트
- [5] Yabukami, S., "Motion capture system using magnetic marker", Magnetics Conference, 2000. INTERMAG 2000 Digest of Technical Papers. 2000 IEEE International, 2000.
- [6] 이희만, 서정만, 정순기, "모션 캡처 애니메이션을 위한 거리 측정방법," 정보처리학회논문지, vol.9-B, no.1, pp.129-137, 2002.
- [7] Liu, Xiaoming, "Video based human motion capture," IEEE Third Workshop on Multimedia Signal Processing, Multimedia Signal Processing, 1999.

- [8] <http://en.wikipedia.org/>, 온라인백과사전
- [9] 정현숙, "모션 캡처 데이터 통합을 위한 새로운 마크업 언어의 구현," 정보과학회논문지, 제30권 제3-4호, pp.219-230, 2003.
- [10] 유흥근, "실시간 기계식 모션캡처 시스템에 대한 연구," 서원대학교 석사학위 논문, 2007.

저자 소개



이 희 만

1994년 6월 : Texas A&M Ph.D
 1996년 ~ 현재: 서원대학교 교수
 관심분야: 컴퓨터 그래픽스, 가상
 현실, 3D 애니메이션