

호환성 있는 인체 애니메이션을 위한 모션 데이터 정의

정철희 이명원

수원대학교 인터넷정보공학과

whiskerfc@hanmail.net, mwlec@suwon.ac.kr

A Motion Data Definition for Compatible Human Animation

Chul Hec Jung Myeong Won Lee

Dept. of Internet Information Engineering, The U. of Suwon

요 약

H-Anim 은 웹3D 컨소시엄의 휴머노이드 애니메이션 워킹 그룹 (Humanoid Animation Working Group)에서 제정한 휴먼 애니메이션에 필요한 인체 구조를 정의한 국제표준안이다. 이 구조에 따라 인체를 움직이기 위한 다양한 라이브러리가 제작되고 있으나 인체의 사실적인 움직임을 표현하는 데에는 한계를 갖는다. 본 연구에서는 H-Anim 구조를 바탕으로 인체 애니메이션 생성에 필요한 모션 정의를 위해 모션 캡처 데이터를 활용하여 인체의 사실적인 움직임을 생성하는 방법을 제안한다. 이를 위해 기존의 H-Anim 명세와는 달리 모션 캡처 데이터를 입력 받을 수 있도록 휴머노이드 모션을 위한 데이터 형식을 정의하고 이를 입력 받아 모션을 디스플레이하는 브라우저를 개발하였다. 본 연구에서의 휴머노이드 모션 데이터 형식은 X3D 기반의 데이터 형식으로서 네트워크상 혹은 서로 다른 브라우저들 사이에서 호환성을 갖도록 하는 것을 목적으로 한다.

Abstract

H-Anim is an international standard that Humanoid Animation Working Group in Web3D Consortium defined the data structure necessary for human animation. Various libraries and tools have been generated according to the structure, but they still have restrictions to represent realistic humanoid motions. This paper presents the method of generating realistic human motion using motion capture data in order to define motion for humanoid animation based on H-Anim standard. In order to implement this, we have defined a data structure capable of receiving motion capture data and implemented a motion browser. The human motion data structure defined in this paper is based on X3D and intended to have compatibility through networks and various browsers.

키워드 : H-Anim, 모션 캡처 데이터, 모션 모델링, 휴먼 애니메이션, 휴머노이드 모션

Keywords : H-Anim, Motion Capture Data, Motion Modeling, Human Animation, Humanoid Motion

1. 서론

H-Anim 이란 3D 인터넷이 성장하면서 온라인 가상환경에서 인간을 표현하고자 하는 필요성이 증대함으로 인해 웹 3D 의 휴머노이드 애니메이션 워킹 그룹(Humanoid Animation Working Group)에서 제정한 네트워크상에서의 인체 애니메이션을 위해 필요한 인체의 추상형과 구조화에 대한 명세이다 [1]. H-Anim 에 의해 인간과 같이 관절이 있는 캐릭터 애니메이션에

있어서 서로 다른 표현 시스템들간의 인체 데이터의 호환이나 공유가 가능하게 되었다. 이와 같이 H-Anim 은 다양한 방법으로 인체 애니메이션 생성을 가능하게 하고 인체 움직임을 위한 데이터 공유를 목적으로 한다고 할 수 있다.

인체 애니메이션 제작에 있어서 데이터 호환을 목적으로 H-Anim 을 기반으로 한 애니메이션 제작 도구에 관한 연구 결과가 있으나, 애니메이션 생성을 위해서 키 프레임 애니메이션 방식으로 되어 있어서 각 키 프레임을 위한 파라미터 입력을 해야 하며 애니메이션의

사실적 움직임 생성을 위해서는 반복되는 수많은 입력 작업으로 인한 시간적 낭비의 한계를 갖는다 [2][3]. 사실적 애니메이션 표현을 위해 수많은 연구들이 진행되어 왔으나 인간의 움직임과 같이 복잡한 움직임을 운동학적 알고리즘 형식으로 표현하는 데는 자연스럽게 많은 결과를 가져오기도 한다 [4][5].

본 연구에서는 인간의 자연스러운 움직임 생성을 위해서 기존의 휴먼 애니메이션에서 써왔던 키 프레임 방식이나 운동학적 혹은 물리학적 알고리즘 방식으로 휴먼 모션을 생성하는 것이 아니라 인간 모션의 호환과 재생성에 목적을 두고, 모션 캡처 데이터를 이용하여 이를 H-Anim 구조에 적용하는 방식으로 인간형 모션 데이터를 추상화하는 방법에 대해 소개한다.

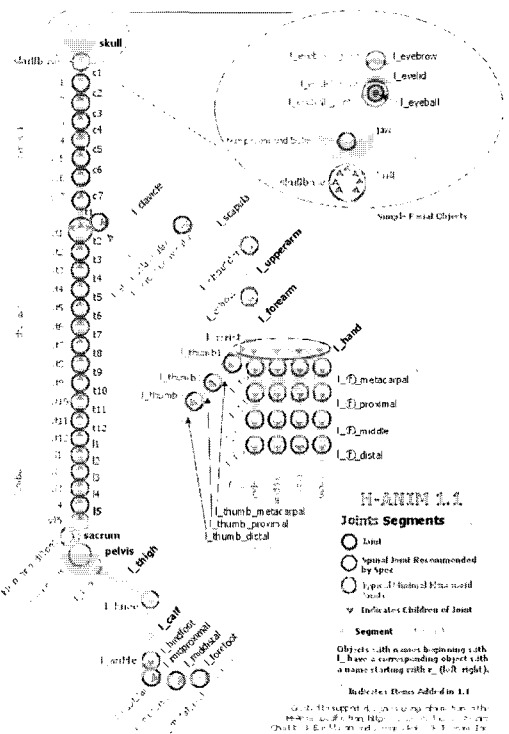
2. H-Anim 개념

H-Anim은 웹3D의 휴머노이드 워킹 그룹에서 제정한 표준안으로 휴머노이드, 즉, 인간형 모델의 표현을 위해 필요한 관절과 세그먼트 등의 구조를 위한 추상화된 데이터 형식에 대해 정의하며 ISO/IEC에 의해 국제표준으로 채택되었다 [1]. H-Anim은 한 저작도구에서 생성된 휴머노이드를 다른 저작도구에서 움직이게 하는 것을 가능하게 하는 호환성을 목적으로 한 추상적 데이터 표현이라고 할 수 있다. 또한 키프레이밍 (Keyframing), 인버스 키네마틱스 (Inverse Kinematics), 퍼포먼스 애니메이션 시스템 (Performance Animation System), 그리고 그 외의 다른 기술에 의해 동작할 수 있도록 한다[6][7]. 기존 연구 중에도 H-Anim의 기반이 되는 XML, X3D 데이터를 사용하여 인체를 애니메이션 하는 방법을 연구한 논문들이 있다[8][9]. 본 연구와 기존 연구와의 차이점은 본 논문에서는 H-Anim 데이터를 사용하여 인체를 애니메이션 할 뿐만 아니라, H-Anim 구조에 정의되어 있지 않은 애니메이션 노드를 추가로 정의하여 국제표준인 H-Anim 구조를 수정하여 H-Anim 국제표준을 개선시키고자 한다.

H-Anim의 휴머노이드 생성을 위해서는, 인체의 각 관절을 표현하고 인체의 각 부위와 상위 부위와의 관계를 정의하는 Joint Node, 신체 부위를 저장하는 Segment Node, 역기구학 체계에 의하여 사용될 수 있는 “중단 작용자”의 위치를 정의하는 Site Node, 움직임의 범위에 관한 정보를 포함하는 Displacer Node, 관절, 인체부위, 시야의 참조 점을 제공하고 전체 인체모형의 용기 역할을 수행하는

Humanoid Node를 도입해야 한다.

H-Anim 형태의 골격구조는 HumanoidRoot Joint에서 휴머노이드의 각 부속물들의 최하위 부속물들까지의 변환을 정의하여 Joint 객체들을 트리 구조로 표현하여 구성하게 된다. 이때 휴머노이드는 통상적으로 중첩된 Joint 노드의 연속체로 구성되며, 각각의 Joint는 그에 관련된 세그먼트를 가진다(그림 1)



(그림 1) H-Anim 구조

3. 모션캡처 데이터

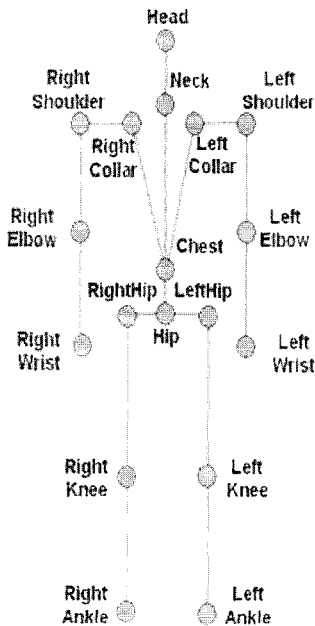
H-Anim 구조에 의해 휴머노이드의 움직임에 필요한 데이터 구조 형태가 갖추어져 있다고 해도 이 휴머노이드를 움직이거나 세그먼트를 구성하는 일은 다시 어플리케이션에 의해 애니메이션 특성에 따른 알고리즘으로 생성되어야 한다. 이렇게 구성된 애니메이션 데이터는 움직임 생성에 있어서 관절간의 움직임의 연관성과 복잡성으로 인해 어플리케이션 알고리즘에 종속하게 될 수밖에 없어서

휴머노이드 데이터 활용을 어렵게 만든다.

본 논문에서는 휴머노이드의 움직임 정의 데이터를 모션 캡처 데이터로부터 얻고 이를 H-Anim 관절 구조를 이용하여 휴머노이드의 모션을 위한 구문을 정의하여 휴머노이드 애니메이션 생성을 위한 데이터 전송을 가능하게 하고 모션 파일을 여러 브라우저 혹은 뷰어에서 공유하는 것을 목적으로 한다.

모션 캡처 파일 형식으로는 Biovision Hierarchy (BVH), Hierarchical Translation-Rotation (HTR), Alias Skeleton (ASK)/Alias Scene Description Language (SDL), Acclaim Skeleton File (ASF) / Acclaim Motion Capture (AMC) 등의 골격 기반의 데이터 형식과 C3D, FBX, GMS 등과 같이 특정한 골격이 미리 정의되지 않는 파일 형식들로 구분된다. 본 연구에서는 이들 모션 캡처 데이터 중에서 많이 활용되고 있는 Biovision 사의 BVH 를 이용하여 모션 정의에 필요한 입력 데이터로 사용하였다.

BVH 데이터는 크게 헤더부분과 데이터부분으로 나눌 수 있다. 헤더부분은 인체의 영덩이를 루트로 하는 계층구조의 정의 및 초기 뼈대(skeleton)의 포즈에 대한 위치정보를 포함한다(그림 2).



(그림 2) BVH 계층 구조에 의한 모델

4. 모션캡처 인터페이스

H-Anim의 휴머노이드 관절 구조에서 모션캡처 데이터를 수용하기 위해서는 모션 캡처 데이터에서의 관절의 수보다 같거나 많은 관절을 가지도록 H-Anim 휴머노이드 모델을 생성하고, 모션 캡처 데이터의 각 관절에 H-Anim 휴머노이드 관절과 세그먼트의 정보를 넘겨주어야 한다.

모션 캡처 데이터의 관절명이 고정되어 있는 것은 아니지만 기본적인 모션 캡처 데이터의 관절에 대응하는 H-Anim의 관절은(표 1)과 같고, 지속적인 모션 캡처 데이터 관절명 추가로 모든 모션캡처 데이터의 관절과 H-Anim의 관절이 매치될 수 있도록 한다.

(표 1)

모션캡처(BVH) 관절 이름	H-Anim 관절 이름
Hips	HumanoidRoot
LeftHip	l_hip
LeftKnee	l_knee
LeftAnkle	l_ankle
RightHip	r_hip
RightKnee	r_knee
RightAnkle	r_ankle
Chest	vc1
LeftCollar	l_sternoclavicular
LeftShoulder	l_shoulder
LeftElbow	l_elbow
LeftWrist	l_wrist
RightCollar	r_sternoclavicular
RightShoulder	r_shoulder
RightElbow	r_elbow
RightWrist	r_wrist
Neck	vc4
Head	Skullbase

H-Anim 휴머노이드가 모션 캡처 데이터를 받아들이기 위해서는(표 1)과 같이 매치 시킨 모션 캡처 관절에서 발생한 움직임 데이터를 휴머노이드 모델의 각 관절에서 일어날 수 있도록 모션 캡처 데이터의 인터페이스가 필요하게 된다. 다음 절에서는 이러한 인터페이스로서 본 논문에서 정의하는 모션 캡처 컴포넌트에 대해 기술한다.

5. 모션 캡처 컴포넌트 정의

기존의 H-Anim 구조에는 인체를 애니메이션 할 수 있도록 정의되어 있는 컴포넌트가 존재하지 않는다. 따라서 본 절에서는 H-Anim 휴머노이드 모델의 움직임을 표현하기 위하여 모션 캡처 데이터를 이용할 수 있도록 애니메이션 생성에 필요한 모션 캡처 컴포넌트를 정의하였다. 이를 위해 H-Anim의 Joint Node에 새로운 필드를 추가하고, 모션 캡처 파일에서 기술하고 있는 모션을 생성하기 위해 Motion Node를 새롭게 정의하였다. 이로써 H-Anim 구조에서도 모션캡처 컴포넌트를 이용하여 인체의 자연스러운 애니메이션이 가능하게 된다.

5.1 Joint Node

H-Anim 휴머노이드 모델이 모션 캡처 데이터를 받아들이기 위해서는 휴머노이드 모델의 각 관절에서 모션 캡처에서의 움직임이 발생할 수 있도록 변환 정보를 가져올 수 있는 인터페이스가 필요하게 된다. 이를 위해 H-Anim의 Joint Node에 다음과 같은 Channels와 ChannelsNumber 필드를 추가한다:

```
Interface Joint {
    // 기존과 동일한 필드
    ...
    // 추가 필드
    int[2] ChannelsNumber
    sequence<string> Channels
}
```

ChannelsNumber 필드는 현재 관절의 채널의 개수 즉, 자유도 (Degree of Freedom)을 정의한다. Channels 필드는 각각의 채널에 고유한 ID를 부여한다. Channels와 channelsNumber 필드는 Joint Node가 파싱될 때 미리 정의된 채널의 개수 및 채널의 고유한 ID를 통해 각각의 채널에 대한 정보를 정의된 순서대로 처리한다. (그림 3)은 기존 H-Anim의 Joint Node를 보여준다.

```
interface Joint {
    float[3] bboxCenter 0 0 0
    float[3] bboxSize -1 -1 -1
    float[3] center 0 0 0
    sequence<Object> children []
    sequence<Object> displacers []
    sequence<float[3]> llimit []
    float[4] limitOrientation 0 0 1 0
    string name ""
    float[4] rotation 0 0 1 0
    float[3] scale 1 1 1
    float[4] scaleOrientation 0 0 1 0
    sequence<integer> skinCoordIndex []
    sequence<float> skinCoordWeight []
    sequence<float[3]> stiffness [0 0 0]
    float[3] translation 0 0 0
    sequence<float[3]> ulimit []
}
```

(그림 3) 기존 H-Anim의 Joint Node

앞에서 기술된 확장 Joint Node에 따라 모션 캡처 데이터에서 추가 필드 값을 설정하면 다음과 같다:

```
Joint {
    // 기존과 동일한 필드
    ...
    // 추가 설정된 필드 값
    ChannelsNumber [ 1, 3 ]
    Channels Xrotate Yrotate Zrotate
}
```

위의 예에서 ChannelsNumber 및 Channels 필드는 Joint Node의 1 번째 채널에 Xrotate 정보, 2 번째 채널에 Yrotate 정보, 3 번째 채널에 Zrotate 정보가 들어감을 의미한다.

5.2 Motion Node

앞 절에서의 Joint Node에 추가된 필드들은 휴머노이드 모델의 각 관절에서의 모션의 종류를 명시하며, 본 절에서의 Motion Node는 각 관절에서의 모션 캡처 데이터로부터 실제 모션 변화를 일으키는 데이터들에 대한 추상 형식을 정의한다. 여기에는 모션 캡처 데이터의 프레임, 프레임 수, 변환 값을 적용하기 위해 Frames, FrameTime, transformation 필드들이 포함된다.

```

Interface Motion {
    int           Frames
    float         FrameTime
    sequence<float> transformation
}

```

Frames 필드는 애니메이션되는 프레임의 수 혹은 모션 샘플들의 수를 지정한다. FrameTime 필드는 샘플링 비율 (Sampling Rate)를 지정하는 필드로, 만약 이 필드 값이 0.033333으로 지정되어 있을 경우에는 초당 30프레임의 샘플링 연산을 수행함을 의미한다. transformation 필드는 각 프레임당 관절의 변환 값을 지정하고, Joint Node에서 새로 정의한 ChannelsNumber 과 Channels 필드의 채널 정보에 따라 각 관절에 적용된다. 다음은 모션 캡처 데이터로부터 모션 파라미터를 추출하여 휴머노이드 모델의 모션을 정의한 예를 보여준다:

```

Motion {
    Frames           601
    Frametime       0.033333
    Transformation  [11.623, 31.312, 64.121, .....
                   11.616, 31.313, 64.107, |
}

```

이 모션 노드에서는 프레임 수 601개, 초당 30프레임의 샘플링 연산, 각 프레임당 관절의 변환 값은 ChannelsNumber 와 Channels 필드에서 정의한 관절 순서대로 11.623, 31.312, 64.121 순으로 회전됨을 의미한다.

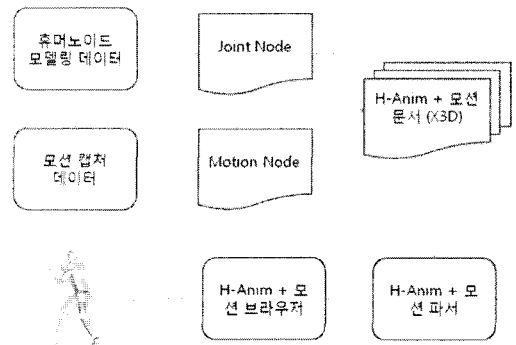
6. 구현

본 연구에서는 휴머노이드 모델 및 애니메이션 생성 및 호환성 구현을 위해서, H-Anim 구조에 모션 캡처를 이용한 모션을 정의하는 노드를 포함시키고 이 방법으로 생성한 휴머노이드 모션 파일을 디스플레이 하는 브라우저를 개발하였다.

모션 캡처 데이터에는 관절의 초기위치와 각 프레임에 대한 회전 정보만 기술되어 있다. 따라서 인체의 움직임을 표현할 수는 있지만 인체의 형태까지는 표현할 수 없다. 이러한 모션 캡처 데이터에 H-Anim 구조의 휴머노이드를 적용시킴으로써, 서로 다른 브라우저나 네트워크 상에서

도 공유 가능한 인체 모델 데이터와 자연스러운 움직임을 제공할 수 있다.

본 연구에서 개발된 브라우저는 기존의 H-Anim구조에 추가된 Motion Node를 파싱하여 휴머노이드 각 관절의 프레임 별 회전 정보를 읽어온다. 이 회전 정보를 샘플링 연산 시간에 맞춰 모든 프레임에 적용함으로써 인체는 자연스러운 애니메이션을 동작하게 된다. (그림 4)는 본 연구에서의 시스템 구성을 보여준다.



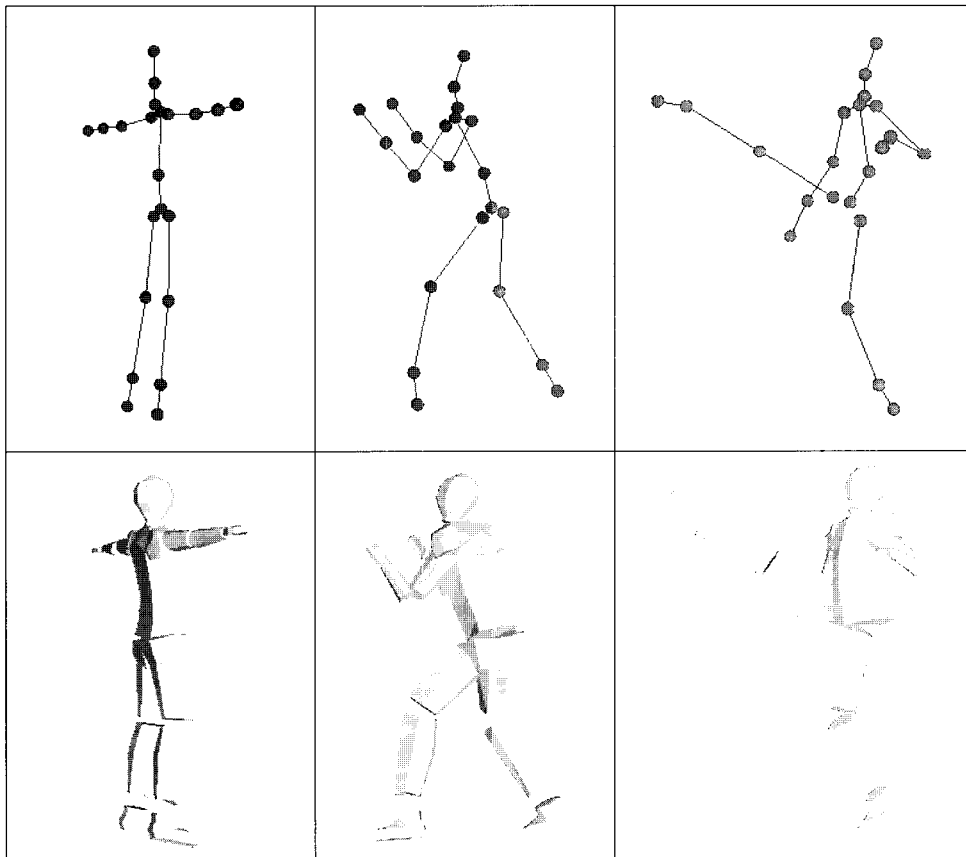
(그림 4) 휴머노이드 모션 생성 시스템

(그림 5)는 모션 캡처 데이터와의 인터페이스로 Joint Node에서의 필드 추가와 Motion Node 가 추가된 샘플 코드를 보여준다.

(그림 6)은 본 시스템을 통해 개발된 휴머노이드 모션 브라우저에서 위 샘플 데이터를 읽어 들여서 디스플레이 한 결과를 보여준다. 이 그림에서 상단은 모션 캡처 데이터를 인체 골격 구조 상에서 관절에 적용시킨 모습을 나타내고 하단은 3D 인체 모델링 데이터와 모션 캡처 데이터를 이용하여 휴머노이드 모션을 생성한 결과를 나타낸다.

<pre> <HAnimHumanoid DEF="sample" name="humanoid" version="1.1"> <HAnimJoint DEF="hanim_HumanoidRoot" center="0 -3.596 -91.49" name="HumanoidRoot" containerField="children" ChannelsNumber= 1 3 Channels= Xrotate Yrotate Zrotate > <HAnimSegment DEF="hanim_pelvis" name="pelvis" containerField="children"> </HAnimSegment> <HAnimJoint DEF="hanim_l_knee" center="10.4 0.7352 -49.93" </pre>	<pre> name="l_knee" containerField="children" ChannelsNumber= 4 6 Channels= Xrotate Yrotate Zrotate > <HAnimSegment DEF="hanim_l_calf" name="l_calf"> </HAnimSegment> </HAnimJoint> </HAnimJoint> <HAnimMotion DEF= hanim_Hips Frames= 601 Frametime 0.03333 Transformation= 11.623 31.312 11.616, 31.313 </HAnimMotion> </HAnimHumanoid> </pre>
---	---

(그림 5) 모션 캡처에 의한 모션 정의 코드



(그림 6) H-Anim 기반의 휴머노이드 모션 생성

7. 결론

본 논문에서는 네트워크 상에서의 호환성 있는 모션 파일 형식을 정의하기 위해 H-Anim 기반의 모션 정의 방법을 제안하고, 휴머노이드 모션 브라우저 개발에 대해 기술을 하였다. 휴머노이드 모션 정의를 위해서 모션캡처 데이터를 이용하였고 H-Anim의 휴머노이드 모델에 적용하여 인체 애니메이션을 구현하였다. 모션 정의를 위해서 H-Anim의 기존 노드에 새로운 노드와 필드를 추가하여 이 모션 데이터만으로 표준화된 휴머노이드 모델에 인체 애니메이션을 동작할 수 있게 되었다.

향후 연구로는 모든 모션 캡처 데이터 형식을 H-Anim의 휴머노이드에 적용하여, 모션 캡처 데이터 포맷에 제한되지 않고 인체의 자연스러운 애니메이션을 구현할 수 있도록 할 예정이다. 또한, 모션 재구성 (Motion Retargetting)을 위한 호환성 있는 데이터 파일 구성에 대해 연구할 예정이다[13][14].

참고 문헌

- [1] ISO/IEC FDIS 19774 – Humanoid animation (H-Anim), Web3D Consortium, 2005.
- [2] 김영신, 이민근, 이명원, “H-Anim 기반의 3차원 캐릭터 애니메이션 제어”, 한국컴퓨터종합학술대회 논문집, pp.194-198, 2007년 06월
- [3] Fabio Buttussi, Luca Chittaro, Danicle Nadalutti, “H-animator: a visual tool for modeling, reuse and sharing of X3D humanoid animations”, Proceedings of the eleventh international conference on 3D web technology, ACM, pp.109-117, April 2006.
- [4] M. Cobo , H. Bieri, “A Web3D Toolbox for Creating H-Anim Compatible Actors”, Computer Animation 2002, pp.120, June 2002
- [5] A.Ortiz, D. Oyarzun, I. Aizpurua, J. Posada, “Three-dimensional Whole Body of Virtual Character Animation for its Behavior in a Virtual Environment Using H-Anim and Inverse Kinematics”, Proceedings of Computer Graphics International, 2004.
- [6] Lucio Ieronutti , Luca Chittaro, “A virtual human architecture that integrates kinematic, physical and behavioral aspects to control H-Anim characters”, Proceedings of the tenth international conference on 3D Web technology (Web3D '05), pp.30, March 2005.
- [7] A.Ortiz, D. Oyarzun, I. Aizpurua, J. Posada, “Three-dimensional Whole Body of Virtual Character Animation for its Behavior in a Virtual Environment Using H-Anim and Inverse Kinematics”, Proceedings of Computer Graphics International, 2004.
- [8] Jeffrey D. Weekley, Curtis L. Blais, Don Brutzman, “Composing behaviors and swapping bodies with motion capture data in X3D”, Web3D '07: Proceedings of the twelfth international conference on 3D web technology, pp. 195-221.
- [9] Hyun-sook Chung and Yilbyung Lee, “MCML: motion capture markup language for integration of heterogeneous motion capture data”, Computer Standards & Interfaces Volume 26, Issue 2, March 2004, Pages 113-130
- [10] Jinxiang Chai, Jessica K. Hodgins, “Performance animation from low-dimensional control signals”, SIGGRAPH '05: ACM SIGGRAPH 2005 Papers, pp. 686-696, 2005.
- [11] T. Igarashi, T. Moscovich, J. F. Hughes, “Spatial keyframing for performance-driven animation”, Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 107-115, 2005.
- [12] Katsu Yamane, James J. Kuffner, Jessica K. Hodgins, “Synthesizing animations of human manipulation tasks”, ACM Transactions on Graphics (TOG), pp.532-539, 2004.
- [13] Michael Gleicher, “Retargetting motion to new characters”, Proceedings of the 25th annual conference on Computer graphics and interactive techniques (SIGGRAPH '98), pp.125-134, July 1998
- [14] Alexander Savenko , Gordon Clapworthy, “Using Motion Analysis Techniques for Motion Retargetting ”, Sixth International Conference on Information Visualisation (IV'02), pp.110, July 2002.

