

웹3D 국제표준화 기술동향

수원대학교 | 이명원*

1. 서론

컴퓨터와 인터넷 기술의 발전과 더불어 웹브라우저 기술로 웹은 이제 우리 생활에서 중요한 정보 제공 및 전송의 수단으로서 활용되는 영역이 급속하게 확대되어 가고 있다. 웹에서의 미디어의 형태도 기존의 2D 멀티미디어에서 3D 멀티미디어로 발전되어 가고 있다(그림 1)¹⁾. 3D 멀티미디어의 특징은 기존의 텍스트, 이미지, 비디오와 같은 2D 미디어로 구성된 단방향성 미디어에 3D 특성인 상호작용성과 3차원 현실감을 제공한다. 이러한 3D 멀티미디어가 웹을 통해 전송 및 통신하기 위해 필요한 기술이 웹3D 기술로서 미국의 Web3D Consortium(www.web3d.org)을 중심으로 관련 기술 개발과 표준화가 진행되고 있다(그림 2)²⁾.

웹3D는 웹에서의 3D 그래픽스 구현에 관한 기술을 말하며 웹을 기반으로 하는 응용 프로그램에서 3차원의 그래픽스 장면의 생성, 전송 및 표현을 가능하게 하는 기술이다. 유비쿼터스 네트워킹 환경에서 3D 디지털 미디어가 통신되도록 하기 위해서는 웹3D와 관

The Micro Universe of 3D Standards

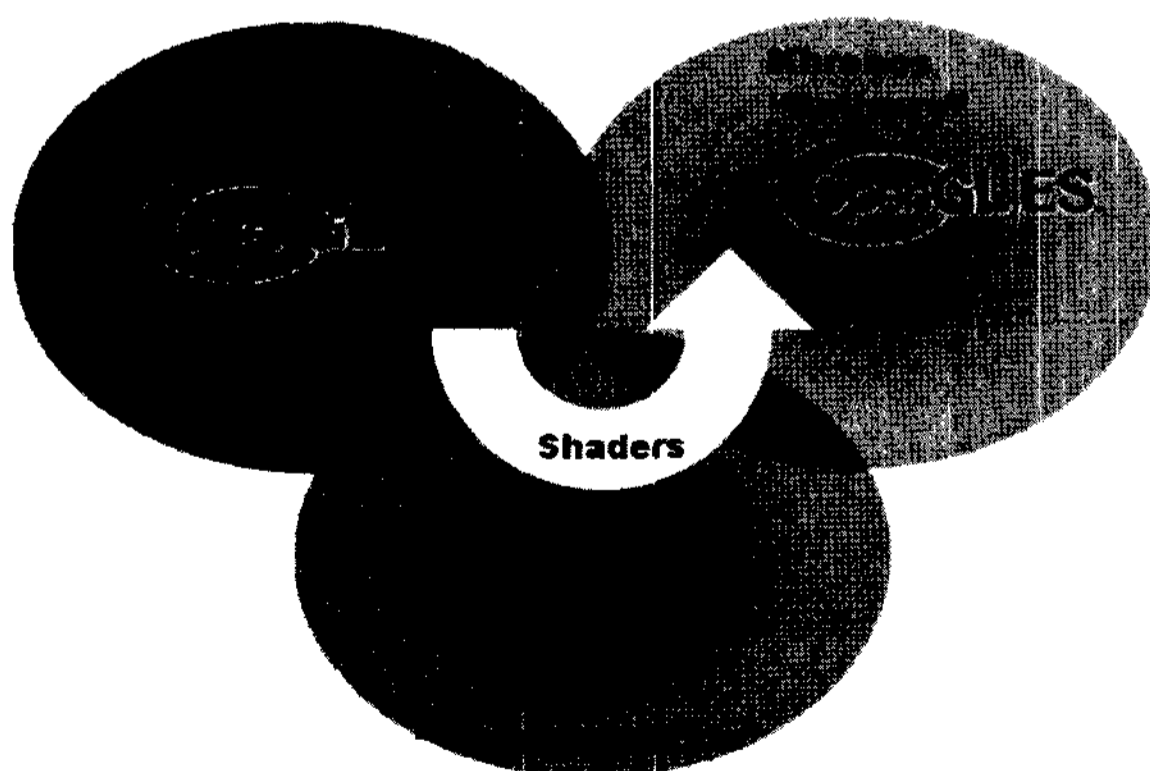


그림 1 웹3D 세계

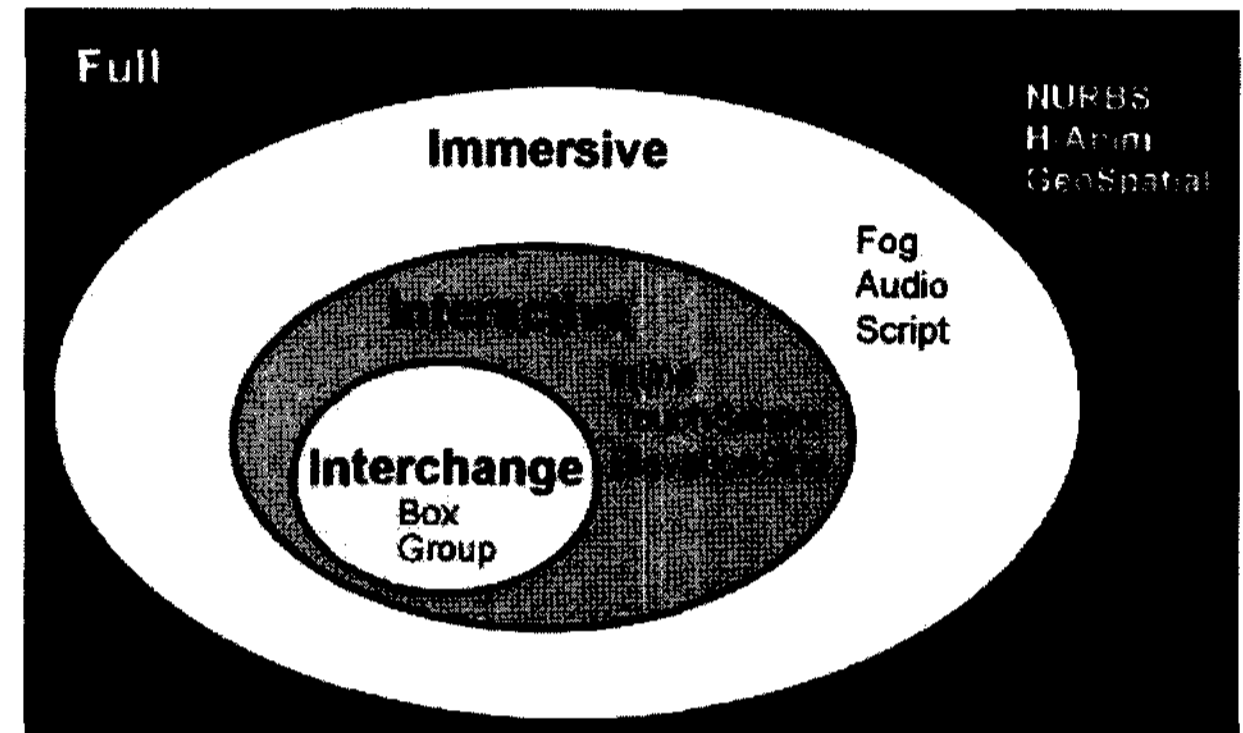


그림 2 X3D 기본 프로파일

련한 표준화 기술이 밀반침되어야 한다. 본 고에서는 웹3D 표준화 기술의 현황에 대해서 소개한다.

2. 웹3D 컨소시엄(Web3D Consortium)

웹3D 컨소시엄은 웹3D 명세에 대한 표준화를 진행하는 국제 민간표준화 단체로서 X3D 규격을 개발하여 응용들 사이에서나 분산 네트워크나 웹서비스를 통하여 웹에서의 3D 통신을 가능하게 한다[1]. 웹3D 컨소시엄은 ISO 및 W3C와의 공동협력을 통하여 표준화 활동을 지속적으로 진행하고 있다. 웹3D 컨소시엄의 워킹그룹은 X3D와 관련하여 특정한 기술적 문제들에 대한 해결책을 제안하거나 연구하는 기술위원회이다. 현재 워킹그룹의 종류에는 X3D Earth, X3D Networking, User Interface, CAD, Medical, VizSim, X3D Conformance Program, Programmable Shaders, GeoSpatial, DIS-XML, H-Anim, X3D Source들이 있다.

해외의 웹3D 컨소시엄에 대응하여 국내에도 한국 표준협회에서 지원하는 표준화포럼인 “웹디지털콘텐츠 표준화포럼”이 있다. 이 포럼은 2007년 6월에 구성되어 웹에서의 디지털콘텐츠의 국제표준화를 진행하고 있다. 본 포럼의 관련 분야로는 2D 및 3D 그래픽스 영상과 관련된 모든 분야가 포함되며, 현재는 웹3D 기반의 휴먼 애니메이션, 모바일 미디어, 의료, GIS, 이러닝 분야를 중심으로 국제표준화 방향을 모색하고

* 종신회원

† 본 연구는 한국표준협회에서 지원하는 웹디지털콘텐츠 표준화포럼 사업의 일환으로 수행됨.

1,2) 2004년 Web3D Consortium 자료에서 발췌

있다. 또한, 수원대, KT, ETRI, 충북대, 동덕여대 등에서도 웹3D 표준화 활동과 기술개발을 추진하고 있다 [9-18].

3. 웹3D와 가상현실

웹3D는 가상현실 모델링 언어인 VRML(Virtual Reality Modeling Language)로부터 시작되었다[2]. VRML은 웹에서 3차원 가상공간 장면을 생성하는 모델링 언어이면서 파일 형식이라고 할 수 있다. 초창기 버전인 VRML 1.0은 웹브라우저에서 3D로 구성된 장면의 인터랙션이 기본 기능으로 제공되었다. 이 때에는 웹브라우저에서 가동시키기 위해 HTML 문서에 VRML 파일(*.wrl)을 삽입하여 웹브라우저에서 VRML 브라우저가 플러그인으로 작동하면서 VRML 브라우저 인터페이스를 이용하여 3차원 장면을 제어하도록 하였다. VRML은 1997년도에 2.0버전인 VRML97이 완성되었다. VRML97은 Web3D Consortium과 ISO/IEC JTC1 SC24의 공동 작업에 의해 ISO 14772 국제 표준안으로 채택되었다.

웹에서의 문서 표준이 HTML에서 XML로 진전됨에 따라서 VRML을 삽입하는 문서 표준도 XML로 진화하게 된다. 그러나 HTML이 여전히 웹에서의 표준 문서로 많이 사용되고 있는 것과 같이 VRML도 웹3D 표준 문서로 여전히 많이 사용하고 있다. 또한, VRML97의 차기 버전인 X3D에서의 3D 기술 명세는 VRML97을 기반으로 하고 있다.

4. X3D 표준 명세

4.1 X3D란

X3D(Extensible 3D)는 저작권 사용료가 없는 공개된 표준의 파일 형식이며, XML을 이용하여 3D 장면과 물체들을 표현하고 서로 통신한다. 이것은 ISO 표준으로 어플리케이션에 삽입되어 실시간 그래픽스 콘텐츠를 저장, 검색, 재생하는 기능을 제공하여 광범위한 영역과 사용자 시나리오를 지원하는 개방형 구조를 가진다. X3D는 공학 및 과학적 가시화, CAD 및 건축, 의료 가시화, 훈련 및 시뮬레이션, 멀티미디어, 오락, 교육 등 많은 분야에서 이용될 수 있도록 부품화되어 있는 특징을 갖는다. 3D 데이터를 모든 어플리케이션과 네트워크 어플리케이션 사이에서의 실시간 통신 개발은 초창기 VRML부터 전개되어 X3D 표준에 와서 더욱 성숙해지고 개선되었다.

X3D는 대화형 웹과 멀티미디어와 통합된 방송 기반 3D 콘텐츠를 위한 소프트웨어 표준이다. X3D는 다양한 하드웨어 장치에서 공학 및 과학적 가시화, 멀티

미디어 프리젠테이션, 오락 및 교육용 타이틀, 웹 페이지, 공유된 가상월드 등과 같은 광범위한 어플리케이션 영역에서 사용될 수 있다. 또한, X3D는 3D 그래픽스와 멀티미디어가 통합된 보편적인 교환 형식을 제공한다. X3D는 웹 기반 3D 그래픽스 표준인 VRML의 후속 버전이다. X3D는 VRML을 개선시켜서 새로운 기능들과 향상된 어플리케이션 프로그램 인터페이스, 추가적인 데이터 인코딩 형식, 엄격한 적합성 및 부품화된 구조로 되어 있다.

X3D는 특정 시장과 기술적 요구사항을 충족시키기 위하여 다음의 설계 목적을 가진다: 런타임 구조와 데이터 인코딩을 분리시키고, XML을 포함하는 다양한 인코딩 형식을 지원한다. 새로운 그래픽, 행동 및 대화형 객체를 추가할 수 있고, 다른 API를 3D 장면 안으로 제공할 수 있다. 그리고, 다양한 서비스 레벨에서 구현되도록 허용하며, 필요할 경우 지정되지 않은 행동들은 제거할 수 있다.

X3D는 폴리곤 기하, 파라미터 기하, 계층적 변환, 조명, 재질, 다중 단계의 텍스처매핑 기능을 제공하는 3D 그래픽스를 포함하며, 텍스트, 2D 벡터 및 3D 변환 계층 내에서 디스플레이되는 텍스트, 2D 벡터 및 평면의 형상을 지원한다. 또한, 휴머노이드 애니메이션이나 모핑을 위한 연속적인 애니메이션에 필요한 타이머와 보간자(interpolators)를 지원한다. 마우스 기반의 픽킹 및 드래깅과 키보드 입력과 같은 사용자 상호작용 기능을 제공하며 카메라, 3D 장면 내에서의 사용자의 움직임, 충돌, 접근 및 가시성 탐지 등과 관련된 네비게이션 기능을 지원한다.

또한 X3D는 사용자가 정의하는 자료형을 생성하여 내장된 브라우저 기능을 확장할 수 있다. 프로그래밍 및 스크립트 언어를 사용하여 장면을 동적으로 변화시킬 수 있으며 네트워크 상에 있는 다른 장면이나 데이터들을 이용하여 하나의 X3D 장면을 구성할 수도 있다. 휴머노이드 애니메이션, 지구공간 데이터, 분산 상호작용 시뮬레이션 프로토콜과 통합하는 물리적 시뮬레이션 기능도 지원한다. 지구 공간적으로 X3D 장면의 객체들을 정확하게 위치시킬 수 있으며 CAD 시스템과 매핑되도록 CAD 모델을 나타낼 수 있다. 그리고, X3D 장면들을 렌더링 그룹들로 구성하여 각 레이어의 객체들이 하위 레이어들의 객체들을 겹칠 수 있도록 하였다. 프로그래밍을 지원하는 셰이더로 X3D 라이팅 모델을 주문형 셰이더 프로그램으로 교체할 수 있으며, 파티클 시스템에 의해 불과 연기와 같은 파티클을 생성할 수가 있다.

X3D 의미론은 시간 기반의 대화형 3D와 멀티미디어 정보의 추상적 기능 행동을 기술하는 것이다. X3D에서는 물리적 장치나 다른 구현 의존적인 개념은 정의하지 않으며, 다양한 장치와 어플리케이션에서의 사용을 목적으로 하고, 기능성에 있어서의 광범위한 해석과 구현 범위를 제공한다.

각 X3D 어플리케이션에서는 정의된 모든 물체와 어플리케이션에 포함된 모든 물체의 월드좌표계를 암시적으로 설정하며 모든 3D 및 멀티미디어 객체들을 명시적으로 정의하여 구성한다. 그리고, 다른 파일이나 어플리케이션으로의 하이퍼링크를 설정할 수 있으며 프로그래밍이나 데이터에 의해 객체의 행동을 정의할 수가 있다. 프로그래밍 혹은 스크립트 언어에 의해서 외부 모듈이나 어플리케이션에 연결될 수 있으며 프로파일이나 컴포넌트를 지정하여 기능적 요구사항을 명시적으로 선언할 수가 있다.

4.2 X3D 프로파일

X3D의 모듈화 구조는 레이어로 되어 있는 프로파일(profile)을 지원하여 몰입형 환경이나 향상된 상호작용을 위한 추가 기능이 제공되고, 모듈화 블록 단위인 컴포넌트로 구성된 작은 크기로 다운로드할 수 있는 공간에서의 데이터 교환 형식에 초점을 두어 어플리케이션이나 콘텐츠 개발자에 의해 쉽게 이해되고 구현될 수 있도록 하였다.

컴포넌트 기반의 구조는 개별적으로 지원될 수 있는 서로 다른 프로파일의 생성을 지원한다. 컴포넌트들은 새로운 레벨을 추가하여 개별적으로 확장되거나 수정될 수 있으며 새로운 컴포넌트들이 스트리밍과 같이 추가되어 새로운 특성을 도입할 수 있다. 이러한 메커니즘으로 한 영역에서의 개발이 전체적으로 규격을 느리게 하지 않으므로 규격의 향상이 신속하게 이루어지도록 한다. 특정 콘텐츠에 대한 적합성 요구사항이 콘텐츠에 의해 요구되는 프로파일, 컴포넌트 및 레벨에 의해 애매하지 않게 정의된다는 것이 중요하다.

그림 2는 X3D 기본 프로파일을 보여준다. 교환(Interchange) 프로파일에서는 어플리케이션들 사이에서 통신하기 위한 기본 프로파일이다. 이것은 기하, 텍스처, 기본 조명과 애니메이션을 지원한다. 렌더링을 위한 런타임 모델이 없으므로 사용하기 쉽고 어떠한 어플리케이션에도 통합되기 쉽다. 상호작용(Interactive) 프로파일에서는 사용자 네비게이션이나 상호작용 (Plane-Sensor, TouchSensor 등)을 위하여 다양한 센서 노드들을 추가하여 3D 환경과의 기본 상호작용을 가능하게 한다. 향상된 타이밍과 추가적인 조명 효과를 위

해서 Spotlight, Pointlight 등이 제공된다. 몰입(Immersive) 프로파일에서는 오디오 지원, 충돌, 안개, 스크립트를 포함하여 전체 3D 그래픽스와 상호작용을 가능하게 한다. 전체(Full) 프로파일에서는 NURBS, H-Anim, GeoSpatial 컴포넌트를 포함하여 정의되어 있는 모든 노드들을 포함한다.

MPEG-4 상호작용 기능은 방송, 휴대장치나 휴대폰을 위해 설계된 상호작용 프로파일의 일부 버전이다. CDF(CAD Distillation Format) 기능은 CAD 데이터를 출판하거나 대화형 미디어를 위해 공개 형식으로 변환할 수 있도록 하는 것으로 현재 개발 중이다.

4.3 X3D와 VRML97의 차이

X3D와 VRML97과의 차이점은 다음과 같다:

- X3D는 VRML 2 월드를 대부분 작동시키는 Classic VRML이 있어서 VRML과 호환된다. VRML의 대부분이 X3D에 포함된다.
- XML은 현재 기업이나 정부 데이터베이스 정보를 위해 필수적으로 되어가고 있다. X3D는 XML 인코딩으로 정보의 관리, 제어, 교환 등이 쉽게 이루어질 수 있도록 한다[3].
- VRML의 가장 큰 문제는 모든 브라우저나 플레이어에서 작동하는 VRML 환경을 개발하기가 어렵다는 점이다. 이것은 VRML 표준에서는 VRML 행동을 기술하는 적절한 명세가 부족하기 때문이다. X3D에서는 장면과 환경들이 브라우저들 사이에서 서로 작동할 수 있는 방법을 제공한다.
- X3D는 부품화되어 있어서 CAD, 의료, 가시화 등 큰 시장 단위로 맞추어진 프로파일 명세를 제공한다. 산업 개발이 진행되면서 나타나는 새로운 기술 도입을 쉽게 할 수 있도록 구성되어 있다.
- X3D는 어떤 플레이어의 저작을 위해서도 일관되고 단순하게 작동된다. X3D 장면 저작 인터페이스는 내외부의 스크립트 언어를 위한 일관된 기능을 제공한다. 이것은 Java와 ECMAScript가 광범위하게 사용되는 VRML과는 다른 점이다. X3D SAI(Scene Authoring Interface)는 통합된 추상적 서비스 집합을 명시하여 어떤 프로그래밍이나 스크립트 언어에게 맵핑될 수 있어서 가상환경이 프로그래밍언어에 관계없이 일관되게 디스플레이될 수 있다. X3D는 Java와 ECMA를 위한 언어 바인딩도 제공하고 있다.
- X3D는 VRML에서 요구되었던 많은 특성들이 완전히 통합되어 제공된다. VRML에서 특정 업체가 제공했던 수많은 임시적인 해결책은 피하였다.

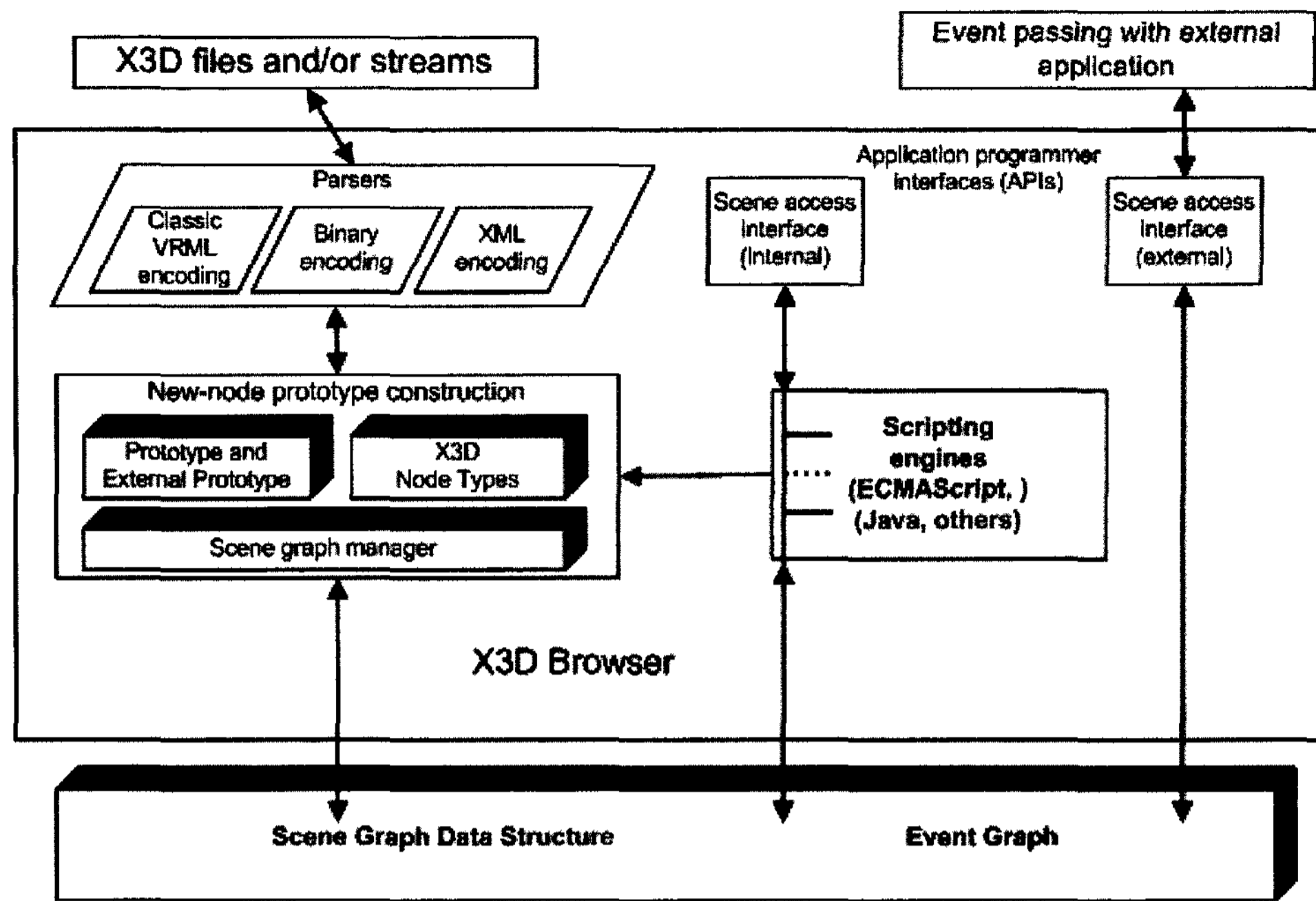


그림 3 X3D 시스템 구조

- X3D는 계속적으로 향상되고 업데이트되고 있다. 개정안에서는 3D 텍스처나 셰이딩 언어 등이 추가되었다. X3D 구조는 변화하는 그래픽스나 상업 시장을 적응되도록 되어 있어서 새로운 기능들을 쉽게 추가 가능하다.
- 웹3D 컨소시엄에 의해 X3D 적합성 프로그램이 개발되어서 X3D 소프트웨어의 적합성을 위해 제공되므로 브라우저나 플레이어 등 어플리케이션들의 신뢰성 향상을 가져왔다. 또한, 보안과 압축 기능을 포함하는 X3D 바이너리 포맷을 제공한다[4].

4.4 X3D 구조

X3D 어플리케이션은 정의된 모든 물체와 각 어플리케이션에 포함되는 물체들을 위한 월드 좌표계를 암시적으로 설정하며 2D, 3D 및 멀티미디어 물체들을 정의한다. 그리고, 다른 파일이나 어플리케이션으로의 하이퍼링크를 설정할 수 있고 물체의 행동을 정의할 수가 있다. 또한, 프로그래밍이나 스크립트 언어를 통하여 외부 모듈이나 다른 어플리케이션과 연결될 수 있다[5,6]. X3D 시스템의 전체 구조는 그림 3과 같다.

하나의 X3D 월드는 하나의 파일에서 개념적으로 구성된 여러 문장들로 정의된다. 파일의 첫 항목은 Header 문장으로 시작하며 두 번째는 PROFILE 문장이다. 다음은 선택적으로 하나 이상의 COMPONENT 문장들이 이어진다. 그 뒤에는 COMPONENT 문장들을 따르는 META 문장들이 선택적으로 나오며, 만약 COMPONENT 문장이 없다면 PROFILE 문장들을 따른다.

ROUTE 문장들은 이벤트 전송을 위한 경로를 지정하는데 사용되며 이 문장들에 의해 한 노드의 필드를 다른 노드의 같은 필드 형으로 연결시키게 된다. PROTO 문장들은 새로운 노드 형을 지정하는데 사용된다. EXTERNPROTO 문장들은 PROTO의 인터페이스 혹은 로컬 파일 외부에 놓여진 EXTERNPROTO 문장들을 지정하는데 이용된다.

4.5 X3D 장면그래프

X3D 런타임 환경의 기본 단위는 장면그래프(Scene graph)이다. 이 구조가 시스템의 모든 물체 및 물체들 간의 관계를 포함한다. 이 관계들은 장면그래프의 여러 축들을 따라 포함된다. X3D는 변환 계층에 의해 물체 렌더링의 공간적 관계를 기술하며 행동그래프(Behavior graph)에 의해 필드들과 이벤트 흐름과의 연결을 기술한다.

하나의 X3D 파일은 0개 이상의 루트(root) 노드를 가진다. X3D의 루트 노드는 다른 노드나 PROTO 문장들에 없는 노드 문장들이나 USE 문장들에 의해 정의된 노드들이다. X3D 장면그래프는 방향성의 비순환 그래프이며 노드들은 계층 구조상에서 자식 노드를 가지는 필드를 포함한다. X3D 변환의 계층구조는 모든 루트 노드들과 그들의 자손 노드들을 다 포함한다. X3D는 조상 노드 좌표계로부터의 변환에서 정의되는 지역 좌표계의 개념을 포함한다. 루트 노드의 좌표계를 월드 좌표계라고 한다.

X3D 브라우저의 일은 사용자에게 하나의 X3D 파일

을 보여주는 것이다. 이것은 X3D 브라우저가 사용자에게 변환 계층구조를 보여주는 것을 의미한다. 변환 계층구조는 가상 월드에서 직접적으로 보이는 부분을 기술한다. 센서나 환경 노드들은 장면 그래프에는 있으나 변환 계층구조에 의해 영향을 받지 않는다. Switch나 LOD 같은 노드들은 렌더링 중에 자식 노드들을 포함한다. X3D를 기술하고 있는 국제표준인 ISO/IEC 19775는 월드 좌표계의 측정 단위를 미터(m)로 정의한다[1]. 각도는 라디안으로, 시간은 초로, 컬러 공간은 RGB([0.,1.], [0.,1.], [0.,1.])로 정의한다. ISO/IEC 19775는 데카르트 좌표계, 오른손 좌표계 및 3차원 좌표계를 이용한다.

X3D의 이벤트 모델은 필드들 간의 연결(Routes)을 선언할 수 있다. 행동그래프(Behavior graph)는 이러한 필드 연결의 집합을 나타낸다. 이 행동그래프는 재연결, 추가, 해지 등에 의해 동적으로 변경될 수 있다. 이벤트들은 시스템 안으로 삽입되어 행동그래프를 통해 정해진 순서대로 전파된다.

4.6 X3D 런타임 환경

X3D 런타임 환경은 장면그래프의 현 상태를 유지시키고 필요할 때마다 장면을 렌더링하고 다양한 센서로부터 입력 데이터를 받아들이고, 행동시스템의 명령에 따라 장면그래프에 변화를 실행한다. X3D 런타임 환경은 내장된 객체와 사용자 정의 객체 및 프로그래밍 스크립트를 포함하여 물체의 생존 사이클을 관리한다. 런타임 환경은 X3D에서 행동 생성의 기본 수단인 이벤트 처리를 조정한다. 또한 런타임 환경은 파일 전송, 하이퍼링크, 페이지 통합 및 외부 프로그램 접근을 위해 X3D 브라우저와 호스트 어플리케이션간의 상호작용을 관리한다.

런타임 환경은 물체를 관리하며 런타임 환경에서 유용한 기능을 포함하는 여러 타입의 내장형 객체를 지원한다. 내장형 객체로는 3D 벡터값, 기하 노드, 노드간 경로 등과 같은 데이터 구조를 나타내는 것들이 있다. 각 노드는 0개 이상의 데이터 값을 저장하는 필드들이나 물체간의 메시지를 보내기 위한 0개 이상의 이벤트를 포함한다.

노드들은 파일에 선언되거나 런타임에 프로시저어 코드를 사용하여 인스턴스화 된다. X3D 저자는 프로토타입 메커니즘을 사용하여 새로운 노드 형을 생성할 수 있다. 이벤트는 X3D 런타임 환경에서 행동을 생성하는 기본 수단으로 시간 기반의 애니메이션을 실행하도록 하고 물체의 픽킹 처리, 사용자 이동과 충돌 탐지, 장면그래프 계층구조의 변경 등에 X3D 전체를

통해서 이용된다. 런타임 환경은 시스템과 잘 정의된 규칙에 의한 평가 순서에 따라 이벤트들을 전달한다. X3D 콘텐츠 저자는 장면의 생성과 관리, 렌더링, 행동, 미디어들의 로딩을 제어할 수 있다. X3D나 외부 언어에 의해 작성된 저작물의 로딩과 통합도 제어될 수 있으며 콘텐츠가 정의하는 확장 능력은 프로토타입 메커니즘을 지원하는 프로파일에서 제공된다.

X3D 시스템은 객체라는 추상적 개별 요소들로 구성되어 있다. ISO/IEC 19775는 각 객체 형의 기능적 명세를 정의하며 구현에 대해서는 지시하지 않는다. X3D 저자는 ISO/IEC 19776에서 기술하는 X3D 인코딩이나 내장형 스크립트 등을 이용하여 런타임에 장면 내 객체들을 정리한다. 필드는 X3DField 객체로부터 파생되며 데이터를 저장하거나 작동하게 한다. 노드는 X3DNode 객체로부터 파생되며 필드 보다는 완전한 공간적 시간적 처리 개념을 가진다. 노드는 데이터 값을 저장하거나 이벤트는 주고받는 하나 이상의 필드를 가진다. 어떤 노드들은 공통된 속성과 바운딩 박스나 그룹핑 노드와 같은 기능성 및 메타 데이터 명세를 나타내는 인터페이스에 의해 추가적인 기능을 구현한다. 또한, X3D는 ROUTE, PROTO, 컴포넌트, 프로파일, 메타데이터 등과 같이 필드나 노드에 저장되지 않은 장면그래프 정보를 위한 객체 형을 정의한다. 센서나 스크립트에 의해 초기 이벤트가 생성되면 이벤트는 ROUTE를 따라 필드로부터 다른 노드로 전달된다. 다른 노드들은 추가적인 이벤트를 생성하고 반응하며 모든 경로를 다 지나갈 때까지 계속한다. 어떤 센서들은 동시에 복수 개 이벤트를 생성하기도 한다. 이와 마찬가지로 비동기적으로 생성된 이벤트들이 동일한 시간에 도착할 수도 있다. 그림 4는 X3D의 실행 모델을 보여준다.

4.7 X3D 컴포넌트

X3D 컴포넌트는 다양한 X3D 객체와 서비스들로 구성되는 관련된 기능들의 집합이다. X3D 컴포넌트는 다음의 요구사항을 충족시켜야 한다: 한 컴포넌트의

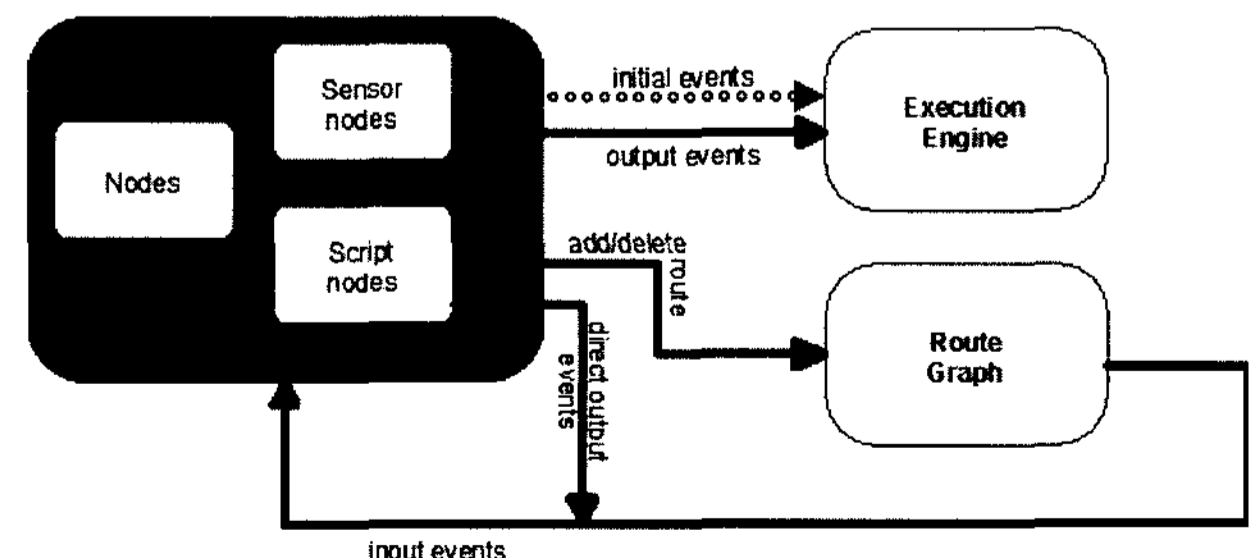


그림 4 X3D 실행 모델

표 1 X3D 컴포넌트 종류

CAD geometry	Core	Cube map environmental texturing	Distributed interactive simulation
Environmental effects	Environmental sensor	Event utilities	Followers
Geometry2D	Geometry3D	Geospatial	Grouping
Humanoid animation (H-Anim)	Interpolation	Key device sensor	Layering
Layout	Lighting	Navigation	Networking
NURBS	Particle systems	Pointing device sensor	Programmable shaders
Rendering	Rigid body physics	Scripting	Shape
Sound	Text	Texturing	Texturing3D
Time			

모든 노드 객체는 X3DNode 클래스로부터 직접 혹은 간접적으로 파생될 수 있어야 한다. 그리고 한 컴포넌트 내 모든 필드 객체들은 X3DField 혹은 X3DArrayField 클래스로부터 파생될 수 있어야 한다. 노드와 필드들의 이름은 국제표준에서 정하는 이름의 의미를 따라야 한다. 표 1은 X3D 컴포넌트의 종류를 보여준다.

4.8 장면 접근 인터페이스(Scene Access Interface)

X3D는 런타임 장면 접근을 정의하는 장면접근 인터페이스(SAI)라는 응용 프로그래머 인터페이스를 제공한다[7]. 개발자는 SAI를 이용하여 노드 생성과 해지, 이벤트 전송, 노드간 연결, 필드값 읽기와 설정, 장면그래프 읽기, 브라우저 작동 제어 등을 할 수 있다.

프로그래밍 접근은 내부적 혹은 외부적으로 할 수 있다. 내부적 접근으로는 장면그래프에서 주문형 요소들을 생성할 때 사용하고 외부적 접근으로는 장면 밖에서 웹브라우저와 같은 호스트 어플리케이션에서와 같이 프로그램 요소들을 연결할 때 사용한다. 내부적 접근은 스크립트 노드라고 하는 특수 노드를 통하여 지원된다. 스크립트 노드는 개발자가 프로그래밍 언어 함수와 장면 그래프의 객체 클래스를 서로 연결할 수 있도록 해준다. 스크립트의 필드는 그 스크립트와 연관되어 있는 객체의 속성이나 메소드에 자동적으로 매핑된다. 스크립트 노드는 런타임 환경에 의해 장면그래프로 역 전파될 수 있는 이벤트를 발생할 수도 있다. 외부적 접근은 X3D 런타임 시스템과 다양한 프로그래밍언어의 런타임 라이브러리 사이의 통합으로 지원된다.

X3D SAI는 언어와 독립적인 서비스로서 여러 프로그래밍 언어 및 스크립트 언어의 바인딩으로서 명시

화된다. X3D SAI 서비스와 컴포넌트 모델 인터페이스의 전체 명세서는 ISO/IEC 19775-2에 기록되어 있다. 여기에서 정의하는 서비스들을 위한 언어 바인딩은 ISO/IEC 19777에 기록되어 있다.

5. 휴머노이드 애니메이션 표준 명세

지난 10여년에 걸쳐서 3D 그래픽스에 대한 관심의 증가로 인해 3D 인체 형상의 생성과 애니메이션을 위한 캐릭터 모델링 SW가 지속적으로 개발되고 있다. 이와 같은 시기에 실제 인간의 움직임을 추적하는 많은 시스템들이 개발되어 왔다. 가장 큰 장애는 많은 소프트웨어 패키지와 시스템들이 캐릭터 애니메이션의 정보 교환이 어렵다는 점이다. 표준화된 골격 시스템이 없으므로 애니메이션이나 모션 캡처 제작사는 자신들만의 해결책으로 시스템과 SW 간의 데이터 이동을 진행해왔다.

ISO/IEC 19774 국제표준인 H-Anim 규격은 이러한 3차원 인체 형상의 모델링을 위한 추상적 표현을 기술한다[8]. 이것은 한 회사의 모델링 도구로 생성한 인체 형상을 다른 회사의 모션 캡처와 애니메이션 도구를 사용하여 애니메이션을 생성할 수 있도록 해주는 휴머노이드 표현에 있어서의 표준화된 방법을 제공한다.

H-Anim은 네트워크 3D 그래픽스와 멀티미디어 환경에서 휴머노이드 표현을 위한 구조적 시스템을 명시한다. 개념적으로 각 휴머노이드는 서로 다른 표현 시스템에 삽입될 수 있고 표현 시스템에 의해 제공되는 장치를 이용하여 애니메이션 될 수 있는 관절형 캐릭터를 말한다. 이 국제표준은 휴머노이드의 추상 형태와 구조를 정의한다. 또한 이 표준은 시간 기반, 대화형 3D, 멀티미디어 관절형 캐릭터의 추상 기능 행동으로서 휴머노이드 애니메이션의 의미를 명시한다. 이 표준은 각 캐릭터의 물리적 형태를 정의하지 않으나 그 캐릭터가 애니메이션을 위해 어떻게 구조화되어 있는지를 정의한다.

5.1 H-Anim 형상

H-Anim 형상은 애니메이션 되는 캐릭터를 묘사하기 위한 관절이 있는 3D 표현이다. H-Anim 형상은 인간과 같은 캐릭터를 나타내기 위한 것으로 인체와 같은 수의 팔다리, 머리, 혹은 다른 인체 부위로 제한하지 않는 일반 개념이다. 하나의 H-Anim 형상을 휴머노이드라고 한다.

H-Anim 형상은 휴머노이드(Humanoid), 관절(Joint), 세그먼트(Segment), 위치(Site), 치환(Displacer)으로 분류하는 H-Anim 객체를 이용하여 기술된다. 휴머노이

드 객체는 H-Anim 형상의 루트로서 휴머노이드의 모든 부위를 위한 프레임워크를 제공한다. 관절 객체는 부착된 몸체 부분과 연관된 기하를 따라 관절의 현재 상태를 명시하는 변환(transform)을 이용하여 휴머노이드 객체나 다른 관절 객체에 붙어있다. 세그먼트 객체는 휴머노이드 형상의 관절들 사이의 물리적 링크의 속성을 정의한다. 위치 객체는 알려진 의미를 관련시킬 수 있는 위치를 정의한다. 치환 객체는 삽입될 객체에 허용되는 움직임의 범위에 관한 정보를 지정한다. H-Anim 형상의 골격에 대한 기술은 Humanoid-Root 관절로부터 휴머노이드 부속물의 종단 효과기(end effector)까지의 변환을 정의하는 관절 객체들의 트리(Tree)로 구성된다. 이 국제표준의 골격 계층 구조의 정의를 위한 유일한 요구사항은 정의된 Humanoid-Root 관절 객체를 가져야 한다는 것이다. 다른 관절 객체들은 선택적이며 H-Anim 휴머노이드 형상을 위해 반드시 요구되는 것은 아니다. 그러나 대부분의 H-Anim 형상들은 HumanoidRoot 외에 더 많은 관절들을 가지고 있다.

관절 레벨은 LOA(Level of Articulation)을 의미하며 휴머노이드 형상에서 필요한 관절의 수를 말한다. 14개 관절을 갖는 휴머노이드 형상은 저수준 관절을 갖고 있는 경우를 말하며 72개 관절을 가지는 경우는 고수준 관절을 가지는 경우를 말한다. HumanoidRoot 관절 객체만 가지는 골격 계층구조는 H-Anim 형상으로 허용되는 가장 낮은 레벨의 관절이다. 이 국제표준의 부속서에는 일반적으로 많이 분류되어 사용되는 H-Anim 인체 형상의 전체적인 움직임 능력을 기술하는 네 종류의 서로 다른 LOA를 정의한다. 그러나, 하나의 H-Anim 인체 형상은 네 개 LOA 중 하나에 적합하도록 제한하나 어떠한 LOA를 사용해도 된다. 골격 계층 구조는 표준의 H-Anim 관절 객체 중에 어떠한 다른 수의 관절 객체들을 추가적으로 포함할 수 있다. 표준 관절 객체의 계층 구조상 조상의 순서만 유지된다면 표준 관절 객체들 사이에 다른 관절 객체들이 삽입될 수 있다.

H-Anim 형상은 관련된 치환(Displacer) 정보로 제한되면서 관절에서의 변환을 적용하여 애니메이션 된다. H-Anim 형상의 수족과 몸체를 재배치하는 능력은 각 수족과 몸체를 정의하는데 사용되는 관절과 세그먼트 수에 따라 다르다. 관절과 세그먼트 수가 많을수록 더욱 유연한 형상이 나오며 관절의 수가 적으면 유연성이 떨어지는 결과를 가져온다.

다음 각 세부 절에서는 다섯 종류의 H-Anim 객체에 대해서 설명한다.

5.2 휴머노이드 객체(Humanoid Object)

H-Anim 휴머노이드의 몸체를 정의하는 기하 데이터는 골격과 피부로 기술될 수 있다. 골격 몸체의 기하 정의는 휴머노이드 객체의 골격을 나타내는 필드에서 정의되며 골격 계층구조의 장면그래프 내에서의 기하 데이터를 기술한다. 이 계층 구조의 세그먼트 객체 내에 정의된 기하 데이터는 분리된 기하 조각들로 몸체를 정의한다. 몸체의 피부 정의를 위해서는 휴머노이드 객체에서 피부를 나타내는 필드 내에 연속적인 기하 조각으로 몸체를 정의한다. 휴머노이드 객체는 좌표와 법선 벡터 데이터를 이용하여 휴머노이드 형상의 피부 곡면을 구성하는 기하를 기술한다. 이 곡면은 하나의 인덱스 면 집합이나 혹은 복수 개의 인덱스 면 집합으로 구현될 수 있다. 휴머노이드 객체는 골격 필드의 장면그래프에서 발생할 수 있는 변화를 반영하기 위해 피부의 좌표와 법선 필드의 데이터를 이용하기도 한다.

5.3 관절 객체(Joint Object)

관절 객체는 휴머노이드 형상의 관절을 기술하기 위한 기본 구성 요소이다. 휴머노이드 형상의 각 관절은 관절 객체에 의해 표현된다. 이것은 골격의 관절 객체간의 부모와 자식간의 관계를 기술하고 골격의 각 관절에 특수한 정보를 위한 컨테이너를 제공한다.

관절 객체는 자신과 자식 필드에 정의된 객체들을 위한 좌표계를 설정한다. 이 좌표계는 대부분의 경우에 다른 관절 객체이면서 휴머노이드 객체인 부모 객체의 좌표계와 관련된다. 관절 객체는 다른 관절 객체의 자식일 수 있는 특수한 그룹 객체거나 HumanoidRoot 관절의 경우에는 휴머노이드 객체의 골격 필드에서 정의된 골격 계층구조의 첫 번째 객체일 수 있다.

관절 객체는 두 필드를 가지며 휴머노이드 객체의 피부 좌표 필드에서 정의되는 정점들을 조작할 수 있다. 관절 객체의 회전 필드 이벤트에 의해 피부 정점 좌표가 영향을 받는다.

관절 객체는 관절에 특수한 정보를 저장하기도 한다. 특히 관절 명칭이 제공되어 어플리케이션이 관절 객체를 확인할 수 있도록 한다. 또한 관절 객체는 H-Anim 형상을 제어하기 위한 역운동학 시스템의 속성값들을 가지고 있기도 한다, 이러한 속성값으로는 관절의 상하 제한 값이나 방향 등이 있다. 어플리케이션에 의해 이러한 제한 사항들의 사용 유무를 결정한다.

5.4 세그먼트 객체(Segment Object)

휴머노이드 형상의 전박, 종아리, 허벅지 등 각 몸체 부분은 세그먼트 객체에 의해 표현된다. 이러한 세

그먼트 객체들은 휴머노이드의 관절 객체 골격 계층 구조로 구성되며 몸체의 각 세그먼트 특유의 정보를 위한 컨테이너를 제공한다. 세그먼트 객체는 각 자식 필드의 객체를 위한 컨테이너를 제공하는 특수 그룹 객체이다. 세그먼트 객체는 관절 객체의 자식으로 정의될 수 있고 관절 객체와 정확히 매치되어야 한다.

5.5 위치 객체(Site Object)

위치 객체는 역운동학시스템에 의해 사용될 수 있는 종단 효과기(end effector) 위치, 보석이나 의복과 같은 액세서리를 위한 부착점, 그리고 세그먼트 객체의 참조 프레임에서 가상카메라를 위한 위치를 정의하기 위해 사용될 수 있다.

위치 객체는 좌우측 눈에서와 같이 어떤 투시적 뷰잉에서 보는 부착점으로서 사용될 때 카메라가 바라보는 방향으로 향할 수 있다. 위치 객체는 세그먼트 객체의 자식 필드 내에 정의될 수 있는 그룹 객체이다. 위치 객체의 회전과 평행이동 필드는 세그먼트의 좌표계 내에서 종단 효과기의 위치와 방향을 정의한다. 위치 객체의 자식 필드는 세그먼트 객체에 부착할 수 있는 어떠한 액세서리도 저장할 수 있다. 위치 객체는 부모 객체의 좌표계와 관련한 자식 필드의 객체들을 위한 좌표계를 정의한다.

5.6 치환 객체(Displacer Object)

각 메쉬 객체의 모양은 어플리케이션의 요구사항에 따라 변경될 수가 있다. 가장 기본적인 레벨에서는 메쉬 객체의 좌표 필드에 저장되어 있는 데이터를 조작하면 된다. 관절 구조의 아바타의 경우에는 메쉬 객체는 세그먼트 객체에 있다. 변형 가능한 메쉬 아바타의 경우에는 메쉬 객체는 휴머노이드 객체의 피부 필드에 의해 정의된다.

메쉬 내에 특정 그룹의 버텍스들을 구분하는 것이 필요하다. 예를 들어, 어플리케이션에 따라서는 머리 내에서 어떤 정점들이 좌측 눈썹을 구성하는가를 알 필요가 있다. 또한, 각 버텍스가 이동하는 방향에 관한 힌트를 제공하는 것도 필요하다. 이러한 정보는 치환 객체에 저장된다. 관절이 있는 아바타에 있어서 특정 세그먼트 객체에 대한 치환 객체는 세그먼트의 치환 필드에 저장된다. 변형 가능한 메쉬 아바타에 있어서 치환 객체는 아바타의 관절 객체의 치환 필드에 저장된다. 이러한 치환 정보는 특정 관절 객체의 지역 공간에서 정의되며 메쉬에 적용되기 전에 휴머노이드 공간으로 변환된다.

치환 객체는 세 가지 방법으로 사용될 수 있다. 가장 기본 레벨에서는 메쉬 상의 특징에 따른 버텍스를

구분하는데 사용될 수 있다. 다음 레벨에서는 여러 방향으로 버텍스들을 치환하는 특별한 근육 행동을 표현하는데 사용할 수 있다. 세 번째 방법으로는 메쉬 내 버텍스들의 완전한 구성을 나타내는데 사용될 수 있다. 얼굴의 경우에는 각 얼굴 표현에 대한 하나의 치환 객체가 정의될 수 있다.

각 치환 객체는 변형 목표인 위치를 지정하여 휴머노이드 형상의 치환 속성을 수정한다. 이러한 치환 객체들에 의한 치환에 필요한 스칼라 양은 보간기와 같은 외부 소스에 의해 동적으로 제공될 수 있다. 그래서 메쉬는 기본 메쉬와 치환 객체에 의해 정의된 치환들의 선형 조합으로 부드럽게 변형될 수 있다. 치환 객체들이 대부분 얼굴의 모양을 제어하는데 사용되나 다른 몸체 부분들을 위해서도 사용될 수 있다. 예를 들어 치환 객체들은 팔이 움직일 때 근육이 부푸는 효과를 시뮬레이션 하기 위해 팔이 변하는 모양을 제어하는데 사용할 수 있다.

5.7 휴머노이드 형상의 모델링

이 국제표준은 하나의 H-Anim 인체 형상을 위해 설계된 애니메이션을 다른 H-Anim 인체 형상으로 전개시킬 수 있도록 H-Anim 인체 형상의 모델링을 제한한다. 이러한 제한에 의해 애니메이션이 적용되기 전에 모델의 상태를 정의하고 H-Anim 인체 형상 모델들 간의 일관성을 유지시킨다.

휴머노이드는 서있는 자세로 +Z 방향으로 보고 +Y 방향으로 서있으며 휴머노이드 좌측이 +X가 되도록 모델링한다. 원점 (0, 0, 0)은 휴머노이드 발 사이에서 땅에 위치시킨다. 발은 땅에 펼쳐놓고 엉덩이 넓이와 같은 거리로 떨어뜨려 놓는다. 발의 바닥은 Y=0이 되도록 한다. 팔은 쪽 펴고 몸 옆에 평행하게 두며 손바닥은 허벅지를 향하도록 한다(그림 5). 그림 5는 휴머노이드의 기본 자세를 보여준다. 손은 펼쳐서 엄지 외

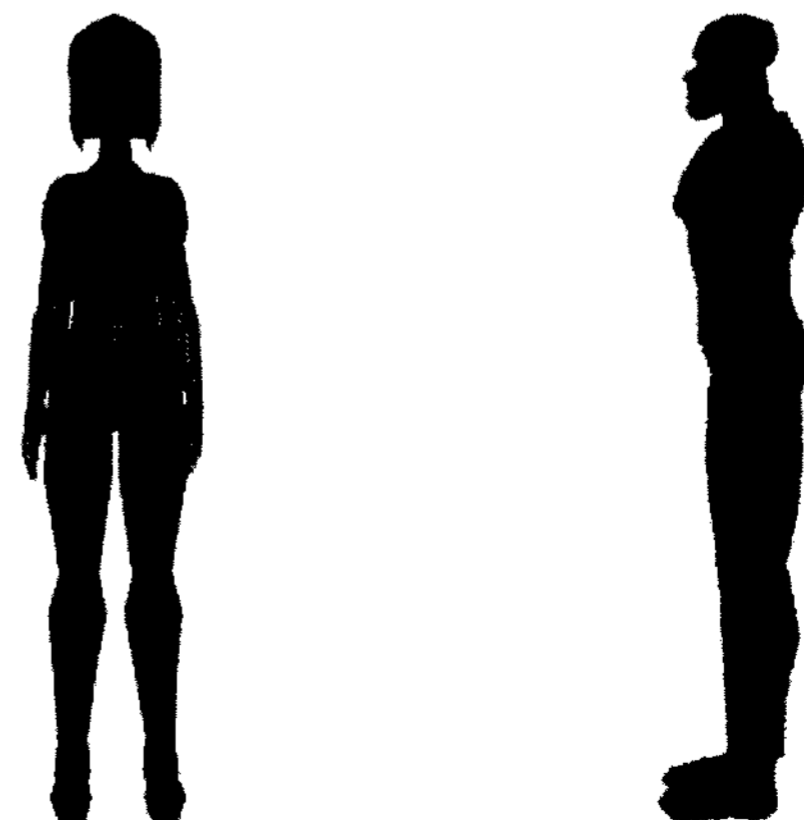


그림 5 휴머노이드 형상의 기본 자세

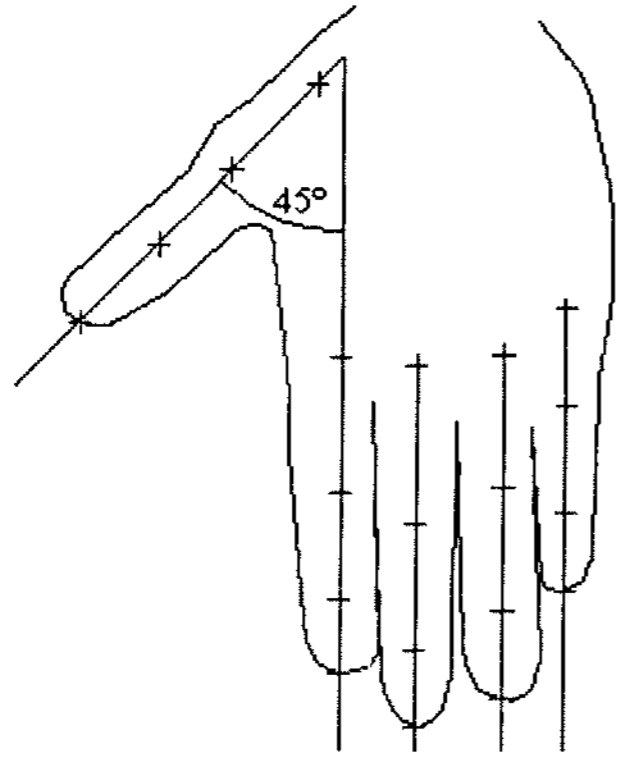


그림 6 손의 방향

손가락의 축은 Y축에 평행하도록 두고 엄지 손가락 축은 +Z 방향으로 $\pi/2$ 라디안으로 회전된 위치에 둔다(그림 6).

얼굴은 눈썹과 함께 모델링되며 입은 닫은 상태로, 눈은 크게 뜬 상태로 모델링한다. 휴머노이드는 실제 인체 크기 범위로 두며 모든 치수는 미터로 한다. 전형적인 인체의 크기로는 대략 1.75 미터로 한다.

5.8 휴머노이드 형상의 구조

인체는 팔꿈치, 손목, 발목 등과 같은 관절에 의해서 서로 연결되어 있는 전박, 손, 발 등의 많은 세그먼트들로 구성되어 있다. H-Anim 몸체는 연속적인 중첩된 관절 객체들로 만들어지고 각 관절 객체는 하나의 세그먼트와 연관되어 있다. 규격에서 휴머노이드 계층구조에 의해 관절과 세그먼트 객체의 쌍이 정의된다. 표준 관절 객체가 정의되고 관절 객체가 세그먼트 객체를 정의하면 이 세그먼트 객체는 계층 구조로부터 적절한 이름을 사용하면 된다. 즉, l_upperarm 세그먼트 객체가 정의되면 이것은 l_shoulder 관절 객체의 자식으로 정의될 것이다. 또한, r_knee 관절 객체가 자식 필드에 하나의 세그먼트 객체를 갖는다면 그것은 r_calf 세그먼트 객체가 된다.

휴머노이드 애니메이션에서 어떤 형태는 인체의 실제 치수와 무관한 경우도 있다. 예를 들어 얼굴을 어떤 각도로 기울이는 것은 두개골 관절 객체를 가지는 어떠한 휴머노이드에도 같은 효과를 갖는다고 할 수 있다. 그러나 어떤 애니메이션에 있어서는 각 세그먼트의 길이나 세그먼트 길이 비율에 의존하는 경우가 있을 것이다. 예를 들어, 손가락 끝으로 코 끝을 만지기 위해서는 인체의 치수를 알아야 할 것이다. 이 규격에서는 각 관절 객체에 있어서 관절 중심의 위치 및 종단 효과기로서 사용되는 위치 객체의 위치를 제공한다. 이러한 치수 기반으로 만들어진 휴머노이드는 다른 휴머노이드와 애니메이션을 공유할 수 있게

된다. 서로 다른 크기의 휴머노이드가 같은 비율의 세그먼트 길이들을 사용한다면 어플리케이션에서 애니메이션 값들을 그에 따라 조정하여 하나의 애니메이션을 공유할 수 있을 것이다. 어플리케이션에 따라서는 전체 관절 수를 필요로 하지 않기 때문에 적절한 관절의 레벨(LOA)을 이용한다. 특정 LOA로 맞추는 것은 휴머노이드가 애니메이션 호환성이 있다는 것이며, 같은 LOA를 가지는 휴머노이드 모델에 호환성을 갖는다는 것을 의미한다.

본 규격의 부속서에는 여러 가지 목적으로 사용하기 위해 인체의 표면 특징점 목록을 정의한다(그림 7). 이러한 특징점들은 치환 노드를 이용하여 기술하고 있으며 각 이름 필드에는 특징점의 H-Anim 이름을 가지며 인덱스 필드에는 특징점에 가장 가까운 벡터의 인덱스 값이 들어간다. H-Anim의 특징점 목록은 CAESAR 프로젝트에서 사용하던 특징점에서 유래한다. ISO 7250은 인체측정학에 사용되는 전통적 인체 측정을 포함하고 있으나 H-Anim과는 일치하지 않는다. 그러나 H-Anim, CAESAR 와 ISO 7250에서 해부학적인 몇 가지 공통적인 것을 포함하기도 한다.

6. 결론

본 고에서는 웹3D 컨소시엄에서 진행하고 있는 표준화 동향에 대해 기술하였다. 웹3D 컨소시엄의 대표적인 국제표준인 X3D와 H-Anim 표준화 기술은 웹에서 컴퓨터 그래픽스 영상을 사용하는 모든 디지털콘

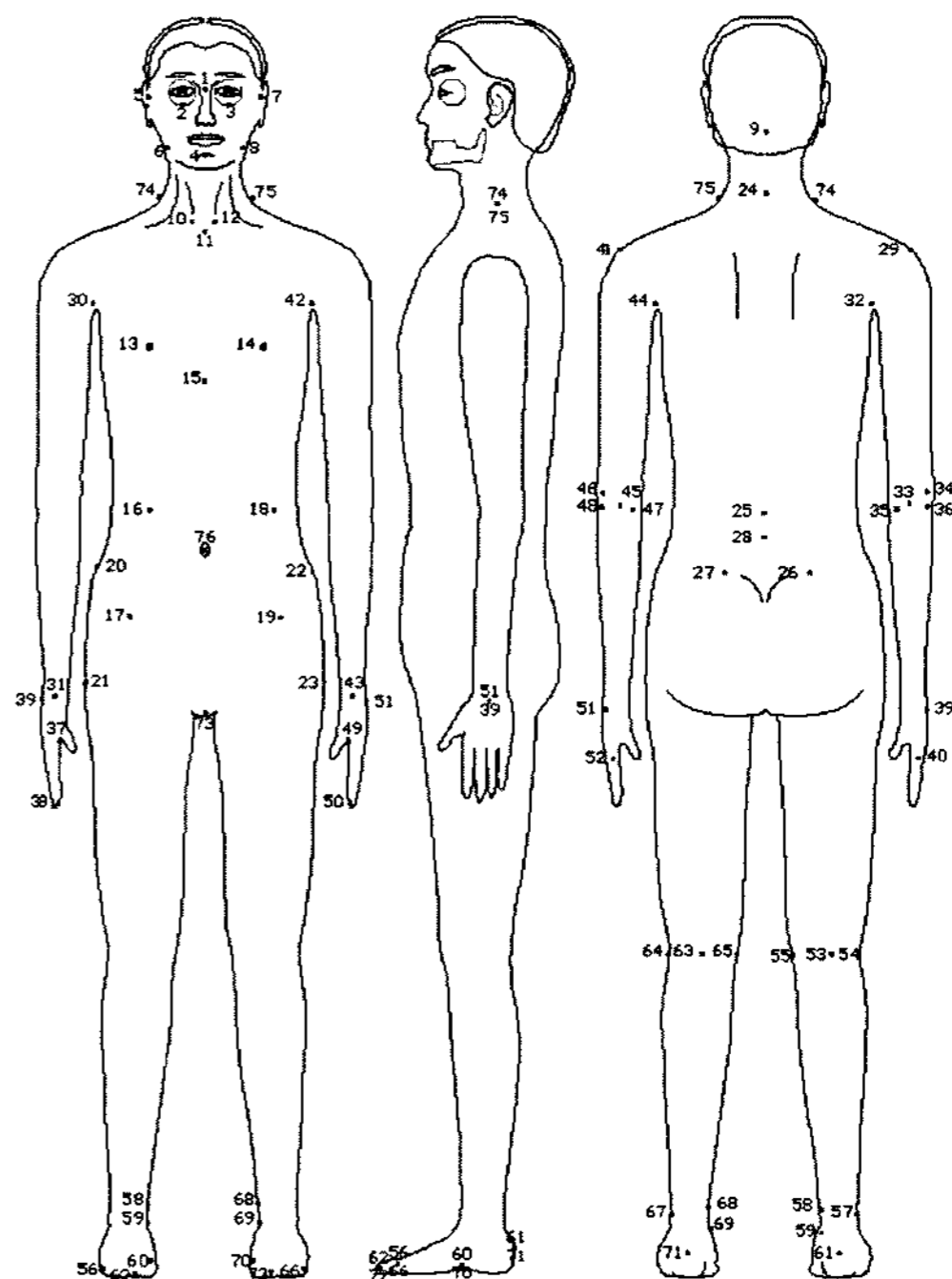


그림 7 인체의 특징점 정의

텐츠 구성시에 필수적인 기술이다. 웹에서 3D 그래픽스를 이용하는 의료, GIS, 이러닝, CAD, 휴먼 애니메이션 등의 분야의 서비스 어플리케이션 개발을 위해서는 X3D가 해결책이라고 할 수 있다.

앞으로 유무선 네트워크 환경의 전개와 가속화할 디지털 컨버전스 시대에서 디지털콘텐츠의 국제표준화 개발은 필수불가결한 경쟁 요소 중 하나가 될 것이다. 디지털콘텐츠에서 3D 시각적 정보 구성의 중요성과 웹의 발전 방향을 고려할 때 관련 산업분야에서의 웹3D 표준화 기술 개발의 활성화가 요구되는 시점이다.

7. 참고문헌

- [1] ISO/IEC FDIS 19775-1.2:2008, X3D Architecture and base components Edition 2, Dec 2007.
- [2] ISO/IEC 14772:1997, Virtual Reality Modeling Language(VRML97), IS, Dec 2003.
- [3] ISO/IEC 19776:2005, X3D Encodings: XML and Classic VRML, IS, Nov 2005.
- [4] ISO/IEC 19776-3, X3D encodings: Binary encoding, IS, Sep 2007.
- [5] ISO/IEC 19777-1:2005, X3D language bindings: ECMAScript, IS, May 2006.
- [6] ISO/IEC 19777-2:2005, X3D language bindings: Java, IS, May 2006.
- [7] ISO/IEC CD 19775-2, X3D Scene application interface Edition 2, Dec 2007.
- [8] ISO/IEC 19774 Humanoid Animation, IS, May 2006.
- [9] “H-Anim 기반의 3차원 캐릭터 애니메이션 제어”, 김영신, 이민근, 이명원, 한국컴퓨터그래픽스학회 논문지, 제13권, 제4호, pp.1-6, 2007년 12월.
- [10] “3차원 스캔 얼굴 모델의 텍스처매핑”, 정철희, 조선영, 이명원, 한국정보과학회 학술대회 논문집, 2007년 10월.
- [11] “가상환경에서의 실세계 물리적 길이 비교”, 정철희, 임창혁, 이민근, 이명원, 한국컴퓨터그래픽스학회 논문지, 제13권, 제3호, pp. 19-24, 2007년 9월.
- [12] “가상 객체의 실제 길이 표현을 위한 다중 레벨”, 이명원, 임창혁, 이용덕, 한국컴퓨터그래픽스학회 논문지, 제13권, 제3호, pp. 25-31, 2007년 9월.
- [13] “가상공간에서의 실시간 날씨정보 표현”, 이종협, 이민근, 이명원, 한국컴퓨터그래픽스학회 학술대회 논문집, 2007년 7월.
- [14] “가상환경에서의 실세계 물리적 길이 비교”, 정철희, 임창혁, 이민근, 이명원, 한국컴퓨터종합학술대회 논문집, 한국정보과학회, pp. 71-77, 2007.
- [15] “H-Anim 기반의 3차원 캐릭터 애니메이션 제어”, 김영신, 이민근, 이명원, 한국컴퓨터종합학술대회 논문집, 한국정보과학회, pp. 208-209, 2007.
- [16] “가상물체의 물리적 속성 구현 방법”, 임창혁, 이민근, 이명원, 한국컴퓨터그래픽스학회 논문지, Vol. 12, No. 1, 2007.
- [17] “Level of Details for Representing Virtual Objects’ Real Length”, 이명원, 임창혁, 이용덕, 한국컴퓨터그래픽스학회 학술대회 논문집, pp37-43, 2006년 11월.
- [18] “가상환경의 물리적 속성 정의 및 구현”, 임창혁, 이민근, 박종인, 이종협, 이명원, 한국컴퓨터그래픽스학회 하계학술대회 논문집, pp.137-144, 2006년 7월.



이명원

1981 서울대학교 졸업(학사)
 1984 서울대학교 대학원 계산통계학과 전산전공 졸업(석사)
 1990 일본 동경대학(The U. of Tokyo) 대학원 정보과학과 졸업(박사)
 1984~1986 데이콤 연구소 연구원
 1990~1993 일본 Kubota Corp. 및 동경대학(The U. of Tokyo) 연구원
 1993~1996 KT 멀티미디어연구소 선임연구원
 1996~2001 수원대학교 컴퓨터학과 전임강사/조교수
 2002~현재 수원대학교 인터넷정보공학과 부교수
 관심분야: 컴퓨터그래픽스, 가상현실, 멀티미디어
 E-mail : mwlee@suwon.ac.kr