

MPEG-4기반 BIFS 노드를 이용한 3차원 컨텐츠렌더링에 관한 연구

김태우*, 김용호, 박영경, 김중규
성균관대학교 디지털신호처리연구실

A study on 3D contents rendering using BIFS node based MPEG-4

TaeWoo kim, YongHo Kim, YoungKyung Park, JoongKyu Kim
Dept. of Information & Communication Engineering, SungKyunkwan Univ

요 약

본 논문에서는 BIFS노드를 추가한 MPEG-4 기반의 3차원 플레이어를 구현했다. BIFS는 바이너리 포맷으로서 VRML에 비하여 압축률이 높고 객체의 시공간적인 관계를 나타내주기 때문에 여러 개의 3차원 객체를 효과적으로 동시에 렌더링할 수 있는 방법을 제공한다. 이러한 BIFS의 특징은 3차원 객체에 대한 압축만을 제공하는 3D 메쉬코딩의 단점을 효율적으로 해결한다. 구현된 플레이어는 BIFS 노드가 추가된 3차원 컨텐츠를 렌더링함은 물론 3차원 객체들에 대한 이벤트를 지원하여 유저와의 상호작용을 가능하게 하였다.

1. 서론

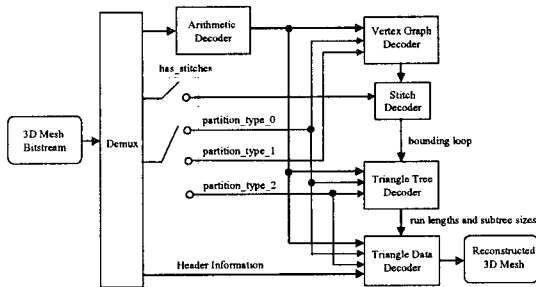
최근 영상, 음성, 텍스트 등의 다양한 데이터를 종합적으로 취급하는 멀티미디어에 대한 연구개발이 다방면에서 활발하게 이루어지고 있다. 그중에서도 특히 3차원 데이터는 인간에게 시각적인 효과를 제공한다. 측면에서 중요한 정보원중의 하나이다. 그러나 3차원데이터의 정보량은 방대해서 효율성 있게 저장, 전송 하기 위해서는 압축이 필수불가결하다. 또한 디지털 미디어의 서비스 발전과 함께 사용자와의 상호작용도 서비스의 필수요건이 되었다. MPEG-4 기반의 3차원 컨텐츠는 높은 압축률로 3차원 데이터가 가지는 고용량의 문제를 해결한다. 이전에는 이

러한 MPEG-4기반의 3차원 컨텐츠를 렌더링하고 사용자에게 컨텐츠의 조작기능을 제공하는 플레이어를 구현했다. 그렇지만 이전에 구현된 플레이어는 단일 3차원 객체만을 지원하기 때문에 컨텐츠의 종류에 제약이 따르고 보다 실감나는 이벤트를 지원할 수 없다는 단점을 가지고 있었다. 본 논문에서는 이러한 단점을 해결하기 위해 플레이어에 BIFS노드를 추가했다. BIFS는 MPEG-4 시스템에서 규격화된 객체의 표시 방법과 특성을 지정하기 위한 장면기술 언어이다. BIFS노드를 추가함으로써 플레이어는 여러 개의 3차원 객체를 동시에 렌더링함은 물론 컨텐츠 제작자는 BIFS정보를 통해 각각의 객체에 이벤트를 부여할 수 있다. 플레이어는 Java 프로그램으로 개

발되었으며 Java3D API를 이용해 3차원 객체를 렌더링한다. 2장에서 디코더의 구조를 간단히 설명한 후 3장에서는 BIFS에 대해 설명할 것이다. 4장에서는 플레이어의 구현 및 실험결과를 보인 후 5장에서 결론을 맺을 것이다.

2. 플레이어 디코더의 구조

3D 메쉬 디코더의 구조는 그림1 에서와 같이 4가지 모듈로 구성된다. Arithmetic 디코더는 입력스트림을 읽어 비트스트림으로 출력해주며 vertex 그래프 디코더는 bounding look-up 테이블을 만들어 준다. Triangle 디코더는 각 run의 길이와 triangle 트리에서 각 서브트리의 크기를 디코딩 해주며 마지막으로 triangle 데이터 디코더는 triangle 데이터 스트림을 출력해준다. Triangle 데이터 스트림은 geometry의 각 삼각형과 관련된 property를 포함하여 renderer가 이 정보를 이용해 렌더링 하게 된다.



[그림 1] MPEG-4 3d mesh decoder 의 구조

3. BIFS(Binary Format for Scene)

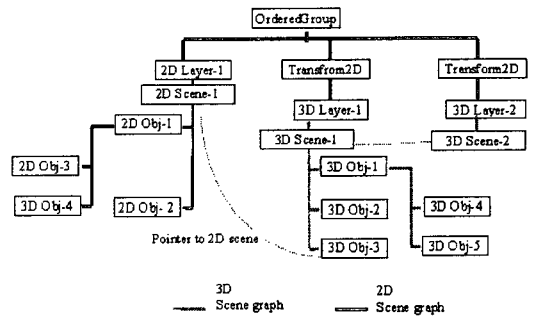
3.1 BIFS

시청자에게 제시되는 화면의 구성을 장면 이라고 한다. MPEG-1, MPEG-2는 장면이라는 개념이 없었지만 MPEG-4 에서는 임의형상의 객체를 부호화 할 수 있다. 또 압축 부호화한 비디오 이외에도 파라미터를 지정하여 생성되는 객체도 취급 할수 있게 되었다. MPEG-4 에서는 객체의 표시 방법과 특성을 지정하기 위한 장면 기술언어로서 BIFS 를 규격하고 있

으며 이것은 오디오 비주얼 객체의 시공간적 관계를 표현하는 바이너리 포맷으로 정의된다. BIFS는 VRML을 바탕으로 하고 있으며 장면은 트리 상태로 배치되어 다양한 노드들의 집합으로 표현된다.

3.2 BIFS 장면 그래프의 구조

BIFS의 장면은 3차원 공간에서 계층적인 구조로 오디오 비디오 객체를 표현하고 각각의 객체에 대한 시공간적인 관계를 표현한다. BIFS는 장면을 묘사하기 위한 방법으로서 그래프로 표현될 수 있고 각각의 객체들은 그래프의 노드로 표현된다. 장면 그래프의 예가 그림2에 나와있다.



[그림2] 장면그래프

3.3 BIFS 의 노드와 필드

BIFS 에서는 VRML에서 정의된 노드에 여러가지 노드를 추가하고 있다. 노드의 종류로는 MPEG-4 장면에서 가장 위쪽에 위치하는 TOP 노드가 있으며, 자식 노드들의 필드를 포함하는 리스트를 가지고 있는 Grouping노드, key frame animation의 선형 보간을 실행하는 Interpolator노드, 이벤트를 감지하는 Sensor노드, 기하학적인 객체를 표현 하는 Geometry노드 및 children노드 등이 있다.

노드는 필드라 불리는 요소들로 이루어진다. 필드는 노드의 성질이며 인터페이스이다. 필드의 특성은 필드의 이름과 필드 값, 필드 값의 형태 및 필드 값의 성질로 구성된다. 표2에서는 노드가 가질 수 있는 필드의 4가지 유형이 나타나 있다.

필드의 유형	
필드는 다음의 4가지 유형으로 구분된다.	
field	변경되지 않는 값. 한번 정의된 값은 변경할 수 없는 필드
exposedField	외부 이벤트에 의해 변경되거나 값을 다른 노드로 전달할 수 있는 필드
eventIn	이벤트를 받기만 하는 필드
eventOut	이벤트를 발생하기만 하는 필드

표2. 필드의 유형

3.4 ROUTE

ROUTE는 노드의 이벤트를 처리하기 위한 구문으로서 특정노드(eventOut)의 값을 다른 노드(eventIn)로 넘겨주는 역할을 하며 이 구문을 통하여 이벤트가 발생하게 된다. 이벤트의 시작은 시간의 흐름이나 마우스를 클릭하는 등의 행동으로 시작 하고 이 이벤트를 인식하는 장치가 Sensor 노드 들이다. 이벤트를 인식한 Sensor 노드는 ROUTE 를 타고 Target 이 되는 노드로 연결되 이벤트가 일어나게 되며 그림 3 은 이벤트가 일어나는 과정을 보여준다.



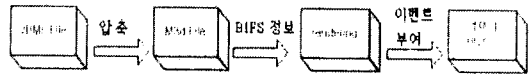
[그림3] ROUTE 를 통한 이벤트 발생

4. 구현 및 고찰

플레이어는 이전에 구현된 플레이어를 기반으로 콘텐츠의 BIFS정보를 분석하고 분석된 정보를 토대로 객체들을 다운받아 렌더링할 수 있는 모듈을 삽입했다. 또한, BIFS의 각 객체에 대한 이벤트 정보를 저장한 후 사용자의 버튼조작으로 각 객체의 이벤트처리가 가능하도록 했다. 실험 과정은 다음과 같다. 먼저 VRML 파일의 각각의 객체들을 압축하여 m3d 파일로 만들고 이 객체들을 구성하는 장면정보

를 BIFS파일로 제작한다. 플레이어는 요청된 콘텐츠에 대한 BIFS파일을 먼저 다운받아 분석한다. 그 다음 분석된 정보를 토대로 콘텐츠를 구성하는 각 객체의 m3d파일을 다운받고 디코딩한 후 화면에 렌더링한다. 이벤트 구현과정은 그림 4와 같다.

본 논문에서는 대상을 3차원 객체로 한정하였기 때문에 표3 과 같은 노드와 필드만을 지원 하도록 하였다. 그림 5, 6, 7 은 몇 가지 콘텐츠에 대한 렌더링 결과를 보여준다.

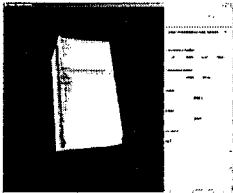


[그림4] 이벤트 구현과정

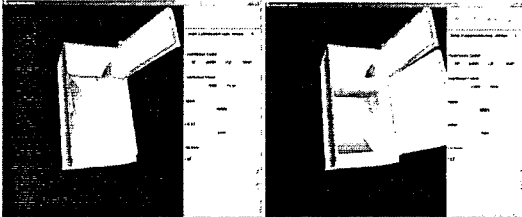
NODE NAME	FIELD NAME	Data Type	Field Type
Transform	Children	MF3DNode	exposedField
	Center	SFVec3f	exposedField
	rotation	SFRotation	exposedField
	scale	SFVec3f	exposedField
	translation	SFVec3f	exposedField
Shape	Appearance	SFAppearanceNode	exposedField
Appearance	Texture	SFTextureNode	exposedField
Image Texture	url	MFURL	exposedField
TimeSensor	cycleinterval	SFTime	exposedField
	loop	SFBool	exposedField
	startTime	SFTime	eventIn
	Fracton_changed	SFFloat	eventOut
Touch Sensor	TouchTime	SFTime	EventOut
Orientation Interpolator	Set_fraction	SFFloat	eventIn
	Key	[MFFloat]	exposedField
	KeyValue	[MFFRotation]	exposedField
	Value_changed	SFVec3f	eventOut

Position	Set_fraction	SFFloat	EventIn
	Key	[MFFloat]	ExposedField
Interpolator	KeyValue	[MFFRotation]	ExposedField
	Value_changed	SFVec3f	eventOut

표3 .플레이어에서 지원하는 노드와 필드



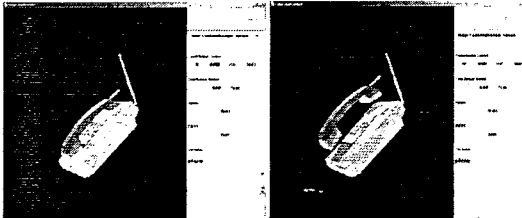
(a) 이벤트 이전



(b) 첫번째 이벤트과정
이벤트과정

(c) 두번째

[그림 5] ref.bif 의 실행 결과

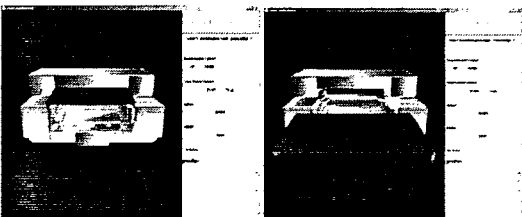


(a) 이벤트

이전

(b) 이벤트 이후

[그림 6] phone.bif 의 실행 결과



(a) 이벤트 이전

(b) 이벤트 이후

[그림 7] printer.bif 의 실행 결과

5. 결론

디지털 방송기술의 발전과 더불어 부가데이터 서비스는 날로 다양해 지고 있다. 이러한 디지털 콘텐츠를 사용자의 요구대로 움직이고 이벤트를 주는 멀티미디어 서비스는 점점 더 중요하게 부각되었고 앞으로 계속 관심을 가져야 하는 연구과제이다.

본 논문에서는 3D 메쉬코딩을 한 3차원 객체에 이벤트를 줄 수 있는 BIFS 의 노드와 필드를 선정, 분석하였으며 이를 토대로 3차원 플레이어를 구현했다. 구현된 플레이어는 기존의 플레이어에 비해 보다 다양한 콘텐츠를 다룰 수 있음은 물론 사용자에게 보다 더 실감나는 서비스를 제공할 수 있다.

본 논문은 디지털 방송의 좀더 다양한 서비스와 구축을 위한 기초 자료로 활용될 수 있을 것으로 사료된다.

[참고문헌]

- [1] 박영경, 김용호, "java 언어를 이용한 MPEG-4기반 3차원 플레이어의 구현 대한전자공학회 제22권 6호, pp817-820
- [2] "Information technology-Coding of audio-visual objects-part 2 :Visual ,ISO/IEC 14496-2, 1999
- [3] "information technology-Coding of audio-visual objects-part 1: System ISO/IEC 1449-1, 1999
- [2] Jean-claude Duford , "BIFS:Scene Description THE MPEG-4 BOOK, pp103~147
- [3] MIKAELBOURGÉS-SEVENIER "2D/3D Scene Composition & BIFS-Update ,MPEG-4 jump-start, pp87~213
- [4] Andre Gueziec, Frank Bossen, Gabriel Taubin, Claudio Silvea, Efficient Compression of Non-manifold Polygonal Meshes ,Visualization 99.Proceedings, pp73-512, 1999
- [6] marrin&Campbell "초보자를 위한 VRML2 21 일 완성 ,인포북