

# MPEG-4 저작물의 설계와 구현

조상범<sup>0</sup> 이남열 전영승 진광범 최민서

csangbum@gce.sejong.ac.kr

## Design and implementation of MPEG-4 Authoring tool

Sang-Bum Cho<sup>0</sup> Nam-Yeol Lee Young-Seung Jun Kwang-Bum Jin Min-Seo Choi  
Department of Computer Engineering, Sejong University

### 요 약

MPEG그룹은 미디어 객체를 시간과 3차원 공간상에 배치하여, 멀티미디어 콘텐츠를 표현하고, 사용자의 상호작용을 가능하게 하는 MPEG-4 System을 표준화하였으며, 다양한 전송계층 (File, TCP/IP, Broadcasting등)을 통합하여 처리할 수 있는 DMIF (Delivery Multimedia Interface Framework)[3] 전송규약을 표준화하였다. 이 논문에서는 MPEG-4 System의 BIFS와 SCENE을 제작할 수 있는 저작물의 설계와 구현을 함으로써 MPEG-4 System을 이용하여 비디오, 오디오, 그래픽과 같은 다양한 미디어들을 결합해 주고 사용자와의 상호작용에 의한 실시간 표현을 가능하게 해주며 사용자가 MPEG-4 System에 대해서 잘 알고 있지 않더라도 손쉽게 MPEG-4 미디어 파일을 만들 수 있도록 해주는 저작물을 제안한다 [1,3,5].

### 1. 서 론

정보사회가 발전함에 따라 새로운 멀티미디어 응용 기술의 개발 필요성이 더욱 증대되고 있다. 요구되는 영상의 질을 보장하면서 낮은 전송률을 가지는 데이터를 생성하기 위하여 block-based hybrid coding 기법을 사용하는 기존의 MPEG-1, MPEG-2, H.263 등의 표준과는 달리, 현재 MPEG-4에서는 기본적으로 content-based coding 또는 Object-based coding 방식을 바탕으로 한다. 즉 영상 내에서 의미를 가지는 객체(object)를 바탕으로 coding하는 개념으로서, 다음과 같은 여러 가지 기능을 가능하게 한다. 우선, 부호기 측에서는 객체를 바탕으로 선택적인 부호화할 수 있게 함으로써 계산량을 줄이는 동시에 주어진 대역폭(bandwidth)에 적절하게 대처할 수 있으며 수신측에서는 복호기의 사양에 맞게 객체를 바탕으로 한 선택적인 복호화를 함으로써 최소한의 데이터로 실시간 디스플레이를 가능하게 한다. 또한 사용자의 입력에 의해 얻어진 각 객체간 재구성(recomposition)을 가능하게 함으로써 사용자가 직접 원하는 동영상을 구현할 수 있는 user interaction을 가능하게 한다. MPEG-4는 기존의 디지털 TV, 동적 그래픽스, 홈페이지에서 지원하는 콘텐츠보다 훨씬 풍부한 재사용성과 유연성을 가지는 콘텐츠이며, 또한, 콘텐츠 소유자의 권리를 더욱 효과적으로 관리하고 보호하는 기능도 지원한다. [1,6]

MPEG-4 콘텐츠는 BIFS(Binary Format for Scene Description)들로 표현한다. BIFS는 스크린에 표현되는 장면을 압축하는 형식으로 기존 도구가 생성하는 콘텐츠보다 효율적으로 압축, 저장, 재생하는 기능을 지원한다. 또한, 장면의 표현 중에도 실시간 변경을 지원하고 있다. 이러한 BIFS는 VRML 2.0[7]의 콘텐츠의 구성방법과 유사하다. MPEG-4와 VRML의 콘텐츠 차이점으로는 장면을 표현하기 위한 객체의 비트 표현, 원격지로부터 데이터 스트리밍 방법, 콘텐츠 재생 및 압축 기법등이 있다. MPEG-4의

BIFS를 작성하기 위하여 일반적으로 텍스트 형식이 사용되고 있으나, 장면 구성이 복잡해짐에 장면을 저장하기 위한 도구의 개발이 필요하다. 따라서 본 논문에서는 구현된 MPEG-4 전용 저작도구를 이용하여 BIFS를 알지 못하는 일반 사용자가 쉽고, 간단하게 멀티미디어 정보를 생성하고, 생성된 멀티미디어 정보들을 이용한 웹서비스의 활용으로 사용자에게 동기화 된 다양한 멀티미디어 서비스를 가능하게 해 줄 것이다.

### 2. 본론

#### 2.1 MPEG-4 System의 구성

MPEG-4 콘텐츠는 크게 IOD (Initial Object Descriptor)와 ES(Elementary Stream)으로 이루어진다. 요소 스트림에는 오디오, 정지영상, 동영상 등의 미디어 스트림과 함께, 멀티미디어 객체의 가상 시공간 배치와 사용자 상호작용을 기술하고, 이를 변경하는 BIFS에 기술된 객체중 요소 스트림에 대한 관련 정보와 그 위치를 나타내는 OD(Object Descriptor)들을 기술하고 이를 변경하는 OD 스트림, 콘텐츠의 메타데이터 정보를 나타내는 MPEG-7 스트림과 OCI(Object Content Information)스트림, 콘텐츠의 저작권보호 정보를 포함하는 IPMP(Intellectual Property Management and Protection) 스트림, 미디어 객체의 속성을 시간에 따라서 변경하는 BIFS 애니메이션 스트림, MPEG 터미널과 미디어 객체를 제어하는 Java 애플릿을 포함하는 MPEG-J 스트림등 멀티미디어 콘텐츠에 관련된 정보를 포함하는 다양한 스트림들을 모두 포함한다.

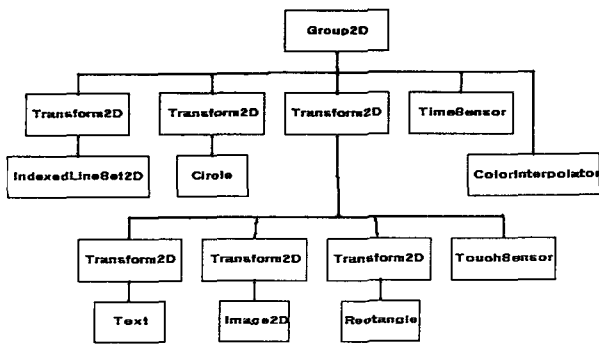
#### 2.2 BIFS 노드

Mpeg-4 콘텐츠로부터 표현되는 장면은 객체를 의미하는 노드들로 구성되며, 장면 트리로 표현한다. 이러한 장면 트리에서 사용하는 노드로는 그룹 노드, 자식 노드, 부착 노드, 센서 노드가 있다. 그룹 노드는 장면의 구조를 구축하는 노드, 자식 노드는 그룹노드의 자식들로서 장면 속의

멀티미디어 객체를 표현하는 노드, 부착 노드는 특별한 속성을 가지는 자식 노드로서 장면의 특정 시점에서 하나만 활성화 될 수 있는 노드이다. 즉, 3차원에서 뷰의 측면을 나타내는 관측 점은 여러 개 가질 수 있지만, 한 시점에 하나씩만 활성화 된다. 보간자 노드는 자식 노드의 서브 타입으로서, 키 프레임 애니메이션을 수행하기 위한 보간 데이터를 표현하는 노드이다. 이 노드는 시간함수 및 다른 입력 인자로서 연속된 일련의 값을 생성한다. 센서 노드는 상호작용을 위한 장면에서 사용자와 환경의 변화를 감지하는 노드이다.

장면 트리에서 각 노드는 장면 속의 그룹이나 객체들의 변환 정보를 가지며, 특별한 행위를 정의하기 위한 필드들로 구성된다. BIFS는 약 100개의 노드를 사용할 수 있고, 노드가 사용하는 필드의 타입으로는 약 20개가 있다. 장면 트리는 이러한 노드들을 조합함으로써 다양한 장면들을 표현한다.

BIFS를 표현하는 장면 트리는 노드 코딩 테이블(NCT: Node Coding Table)과 노드 데이터 타입 테이블(NDT: Node Data type Table)을 사용하며, 노드와 각 노드 필드의 구분은 장면 트리의 문맥에 의존한다. NDT는 노드들의 집합들로 구성되는 테이블이며, NCT는 각 노드의 필드를 기술하는 테이블이다. 필드는 필드 구분자, 이벤트 입력력 관련 필드들로 구성된다. 장면 트리는 NCT와 NDT를 참조하여 MPEG-4장면을 표현하며, 장면을 표현하는 관계 함수  $F$ 는  $F : NCT^+ * NDT^+ \rightarrow Scene$ 이다. NCT<sup>+</sup>는 NCT의 각 노드의 집합, NDT<sup>+</sup>는 각 노드의 필드 값에 따라 참조되는 노드들의 집합, Scene은 장면들의 집합이다.[1,3] 그림 1은 저작된 콘텐츠로부터 생성되는 장면 트리의 한 예이다.

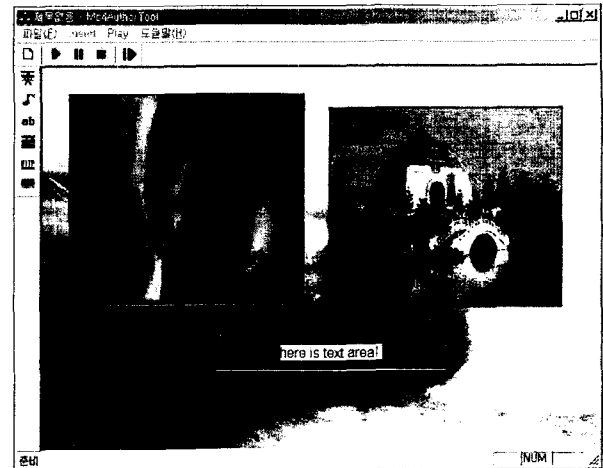


<그림 1. 장면 트리>

### 3. Mpeg-4 저작도구의 설계 및 구현

본 논문에서 구현된 MPEG-4 저작도구는 Visual C++ 기반으로 한 사용자 편의적인 인터페이스로 구현되었고, 일반 사용자가 다양한 특장의 미디어들을 공간적으로 동기화 시킨 멀티미디어 표현물, MPEG-4파일을 저작하기 위한 솔루션을 제공한다. 콘텐츠가 제작되는 순서는 (1) 장면 트리들을 구성하게 되고, (2) 구성된 장면 트리의 각 노드에 속성을 할당하게 되고, (3) 생성된 장면 트리를 전위탐색으로 각 노드를 방문하면서 텍스트 스크립트를 생성하게

되고, 마지막으로 생성된 텍스트코드는 비트맵 맵핑 테이블에 따라 BIFS로 인코딩 하게 된다. 저작도구의 주요 모듈들은 이러한 절차에 따라서 다음과 같은 특징을 갖도록 구현하였다.



<그림 2. MPEG-4 저작물의 인터페이스>

#### 3.1 객체 생성 모듈

객체생성 모듈에서는 저작물에 필요한 동영상, 이미지, 사운드 등을 추가할 때 사용된다. 이렇게 객체를 생성한 후 같은 이미지 객체라 할지라도 서로 다른 이벤트를 만들어 주어서 서로 다른 행동을 하도록 다른 이벤트를 줄 수도 있다.

#### 3.2 공간 모듈

이 공간에서는 객체 생성모듈에서 생성된 객체들을 공간적으로 동기화 시킬 수 있는 영역을 설정한다. 사용자는 생성된 객체를 원하는 곳에 드래그하여 위치시키고 해당 미디어들이 재생할 수 있는 기능을 제공하게 된다. 이곳에 생성된 객체는 사용자가 임의의 데로 위치를 변경하거나 객체의 크기를 자유롭게 변경시킬 수 있도록 구현되었다.

#### 3.3 Timer 모듈

이 모듈에서는 시간을 진행시켜 각기 시간에 따른 미디어들의 행동을 동기화 할 수 있게 해준다. 동영상이 있는 경우 동영상을 진행시키고 동영상의 진행에 따라 이미지나 텍스트들의 정보를 변경시키거나 특정시간에서 각 객체들의 행동을 정의할 때 사용된다.

#### 3.4 Property 모듈

MPEG-4파일이 생성되기 전에 각 객체는 SCENE과 MUX 파일 각 객체의 정보를 저장하게 된다. SCENE파일에는 각 객체가 플레이어에서 보여질 위치나 행동 등을 정의하게 되고 MUX 파일에는 각 객체들의 미디어에 맞는 속성을 가지고 있다. Property 모듈에서는 MUX파일을 구성하고 있는 각각의 객체들에 대한 속성을 설정하게 된다.

### 3.5 Preview 모듈

각각의 객체들의 속성과 시간대별 행동을 설정한후 사용자가 확인해볼 수 있는 기능을 제공한다. 따라서 이를 이용해서 사용자가 일일이 MP4파일을 만들어서 MPEG-4 플레이어에서 실행해 확인하지 않고도 확인할수 있도록 하였다.

그림 3.은 저작물에 의해서 MP4 파일이 만들어지기 전에 생성된 SCENE 파일의 일부이다.

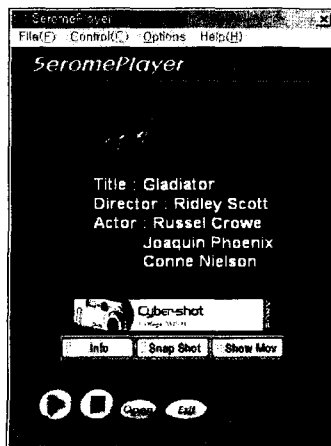
```

1 | DEF MAIN Group {
2 |   children {
3 |     DEF MOV Transforms2D {
4 |       translation 0 30
5 |       children {
6 |         Shape {
7 |           appearance Appearance {
8 |             texture MovieTexture {
9 |               url 2
10 |              loop TRUE
11 |              speed 1
12 |              startTime 4
13 |              stopTime 0
14 |            }
15 |           geometry Rectangle {
16 |             size 200 140
17 |           }
18 |         }
19 |       }
20 |     }
21 |   }
22 |   DEF INTRO Transforms2D {
23 |     translation 0 30
24 |     scale 0.1 0.1
25 |     children {
26 |       Shape {
27 |         appearance Appearance {
28 |           texture ImageTexture {
29 |             url 5
30 |           }
31 |         }
32 |         geometry Rectangle {
33 |           size 220 180
34 |         }
35 |       }
36 |     }
37 |   }
38 | }
39 |
40 | Sound2D {
41 |   intensity 1.0
42 |   spatialize FALSE

```

<그림 3. MPEG-4 저작물에 의해 생성된 SCENE 파일>

그림 4 는 MPEG-4 저작물에 의해 생성된 MP4파일을 세콤 MPEG-4에 의해서 실행되고 있는 장면을 나타낸다. 본 저작물은 세콤의 Decoder를 사용하고 있다.



<그림 4. 표현되고 있는 MPEG-4 장면>

### 4. 결론 및 향후 개발방향

본 논문에서는 다양한 목적에 사용할수 있는 MPEG-4 기반 콘텐츠 저작 도구를 제안하였다. 기존의 콘텐츠 저작 방법들은 대부분 텍스트 기반, 동형의 미디어 콘텐츠 표현, 클라이언트에 미디어 수신 후 재생이 가능하였다. 그러나 MPEG-4 스트림은 다양한 미디어의 실시간 수신 및 재생, 장면의 객체 기반 구성, 스트림의 높은 압축율, 상호작용 등을 특징으로 가짐으로써 기존 콘텐츠보다 효율적인 장면을 재생한다. 이러한 MPEG-4의 속성은 더욱 다양해지는 미디어들의 결합 및 표현으로 다양한 인터넷 쇼핑물의 콘텐츠의 효율적인 저작을 지원한다.

앞으로의 연구 방향은 3차원 저작, XMT[1]를 이용한 기반 정보를 사용하여 XML 기반으로 각 객체를 표현한 후 MPEG-4 콘텐츠만이 아닌 SMIL[2]과 같은 다양한 콘텐츠를 제작할수 있는 방법을 연구중에 있다.

### 5. 참고 문헌

- [1] ISO/IEC JTC1/SC29/WG11, "ISO/IEC 14496-1:2000(E) - MPEG-4 System", MPEG Document No. N3850, 2000년 10월.
- [2] Synchronized Multimedia Integration Language(SMIL2.0)specification, <http://www.w3.org/TR/2001/REC-smil20-20010807/>.
- [3] ISO/IEC JTC1/SC29/WG11, "ISO/IEC 14496-1 CD - MPEG-4 DMIF", MPEG Document No. N2206, 1998년 5월.
- [4] P. Marchisio, "Outlook of Standards for Multimedia Applications", W3C Workshop on "Television and the Web", <http://w3c.bilkent.edu.tr/Architecture/1998/06/Workshop/paper10>, June 29-30, 1998.
- [5] 김해광, "MPEG-4 시스템 참조 소프트웨어의 구성 및 동작분석", 한국방송공학회 논문지, 7권, 2호, 266-271.
- [6] A. Puri, and A. Eleftheriadis, "MPEG-4: An Object-based Multimedia Coding Standard supporting Mobile Applications", Mobile Networks & Applications, Vol.3, No.1, pp.5-32, 1998.
- [7] Web3D Consortium, VRML 200x work in progress, <http://www.web3d.org/technicalinfo/specifications/vrml97/index.htm>