MPEG-4 기반의 멀티미디어 동기화 모델과 응용

성 승 규[†] · 이 명 원[††]

요 약

본 논문에서는 인터넷 멀티미디어 응용 개발에서 필요한 미디어 객체간 시간 관계를 기술하는 동기화 모델을 정의한다. 이 모델에서는 오디오, 비디오, 이미지 등 다양한 멀티미디어 객체들을 통합하여 관리하고 전송하는 표준인 MPEG-4 시스템의 BIFS 구조에 새로운 시간관계 기술 노드를 추가하고 이들간의 관계를 정의한다. 시간관리 노드는 객체의 시작시간, 재생시간, 지연시간을 바탕으로 한 객체들의 시간적 위치관계를 표현한 멀티미디어 시간 모델을 이용하여 생성된다. 또한, 본 논문에서는 이러한 시간 관리 노드를 이용하여 사용자가 직접 멀티미디어 객체에 시간적 동기화를 지정하고 재생시키도록 하는 MPEG4 저작 시스템에 대해서도 기술한다. 기존의 저작도구들이 BIFS 노드를 직접 편집할 수 있는 전문가를 대상으로 하는 것에 비해, 본 시스템은 멀티미디어 객체들간의 시간 관계를 일반사용자가 쉽게 정의할 수 있는 인터페이스를 제공하는 장점을 갖는다.

MPEG-4 Based Multimedia Synchronization Model and Application

Seung-Kyu Sung[†] · Myeong Won Lee[††]

ABSTRACT

This paper describes a multimedia synchronization model based on the MPEG-4(Moving Picture Expert Group) system. It defines and modifies new nodes for representing temporal relationships between media objects in the BIFS(Binary Format for Scene) of MPEG-4 system which integrates, manages and transfers multimedia objects such as audio, video, image, etc. The relationships are represented by using a multimedia temporal model during the start, play and delay time interval. In addition, we illustrate a multimedia authoring system that includes the interface used for defining the temporal relationships. Differently from several contentional tools generally appropriate for professional users who can edit the BIFS nodes of themselves, the system provides end-users with the function that can define the temporal relationships of multimedia objects directly in the interface.

키워드 : 멀티미디어 동기화(Multimedia Synchronization), 시간 동기화(Temporal Synchronization), MPEG-4, BIFS, 멀티미디어 저작(Multimedia Authoring)

1. 서 론

MPEG-4는 대역폭이 적은 통신매체에서도 전송이 가능하고 양방향 멀티미디어를 구현할 수 있는 오디오/비디오 표준 부호화 방식이다[7, 11]. 또한 오디오, 비디오, 텍스트, 정지영상, 2D 및 3D 그래픽 등 다양한 객체들을 컨텐츠로 구성할 수 있다. MPEG-4에서 정의하는 BIFS(Binary Format for Scene)라는 기술언어는 각 객체들의 특성과 표현 방법 등을 지정해 장면을 구성할 수 있도록 해준다. BIFS는 오디오, 비디오 데이터와 2차원 및 3차원 그래픽을 조작하기 위한 정보를 다루며 이들을 노드 단위로 묶어 트리 형태로 계층화시킨다. 또한 멀티미디어 데이터를 위해 시작시간, 종료시간 등을 저장할 수 있는 기능을 제공하는데 이 정보들은 단순히 한 객체에 대한 시간 정보이며 각 객체들간 시간관계에 대한 정보저장 기능은 제공하지 않는다.

본 논문에서는 멀티미디어 객체들의 시간관계 정보를 저장하여 객체간의 시간적 동기화가 가능하도록 확장된 BIFS를 설계하고 이에 따라 MPEG-4 컨텐츠를 제작할 수 있는 저작시스템을 구현한다. BIFS에 객체 간 시간 동기화 기능을 추가함으로서 MPEG-4 시스템 자체에서 동기화 문제를 해결할 수 있어 별도의 동기화 모듈이 추가되는 부담을 줄일 수 있다. 객체 간 동기화를 위한 시간관계 모델은 n개의 객체에 대한 시간관계 모델[13]에 기본을 두고 있으며 객체들 간의 관계를 시간간격(temporal interval)에 따른 모델들로 설정하고 있다. 이들 관계를 기존 BIFS노드에 추가하고 각 노드는 관계에 따라 필요한 정보를 저장한다.

본 논문의 2장에서는 MPEG4 시스템 표준과 BIFS 기술언어에 대한 설명과 기존 MPEG4 저작시스템 및 관련 연구에 대해 분석한다. 3장은 본 논문에서 정의하는 동기화 모

† 정 회 원 : 수원대학교 대학원 컴퓨터과학과
†† 정 회 원 : 수원대학교 인터넷정보공학과 교수
논문접수 : 2003년 10월 21일, 심사완료 : 2004년 7월 13일

델의 개념인 시간관리를 위한 노드와 관계에 대해 설명한다. 4장에서는 시간정보 관리 노드를 정의하고, 장면트리의 생성 방법을 기술한다. 5장과 6장에서는 동기화 정보를 포함하는 BIFS 노드의 생성 모듈과 장면을 구성하는 객체의 시간 관리 방법을 각각 설명하며, 7장에서는 구현에 대해 기술한다.

2. 관련 연구

MPEG-4의 가장 큰 특징은 장면을 구성하는 비디오나 오디오의 객체들을 개별적으로 다룰 수 있도록 인코딩 하는 것이다. 디스플레이 화면에 여러가지의 객체를 표현할 때 종래의 MPEG-1/2에서는 화면이나 배경 음악 전체를 하나의 인코딩 대상으로 간주해서 실행한다. 여기에서는 화면의 각각의 객체들의 화상을 구별하지는 않으며 객체별 음원을 구별하지도 않는다. 그러나 MPEG-4에서는 장면을 구성하는 AV(Audio/Video)객체에 대응하는 정보들을 생성한다. 화면에 디스플레이 되는 오디오, 비디오, 이미지 및 텍스트 등의 객체마다 독립된 정보를 저장하고 표현하는 것이 가능하다. 따라서 각 AV객체 마다 독립적인 조작이 가능하게 되는 것이다. 이러한 AV객체의 개념을 도입하여 새로운 서비스의 창출이 가능하도록 하는 기술언어가 바로 BIFS이다. 이것은 기본적으로 VRML(Virtual Reality Modelling Language)에 상당부분 그 기반을 두고 있으며 VRML보다 더 많은 노드들을 정의하고 있다. BIFS는 바이너리 형태로 압축 코딩되므로 전송측면에서 VRML보다 효율적이다. BIFS는 화면에 구성되는 객체들의 위치, 속성 및 모양 등의 여러 정보들을 각각의 노드로 구성하여 트리형태로 저장한다.

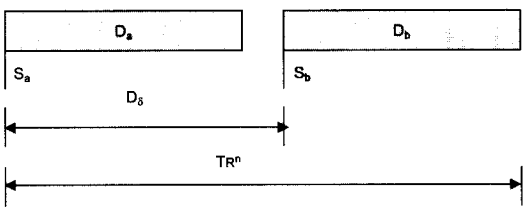
MPEG-4 저작 도구로는 프랑스의 ENST에서 개발한 MPEG-Pro[1, 16], 그리스의 ITI에서 개발한 3D MPEG-4 저작도구[14, 15], envivio의 EBS[4] 및 2D MPEG-4 저작도구[2] 등이 있다. MPEG-Pro와 ITI의 3D MPEG-4 저작도구는 각각 2차원과 3차원 객체를 중심으로 한 저작도구로서 각 객체를 위한 BIFS노드의 속성값을 수정한다거나 객체 기술자(Object Descriptor)와 BIFS노드간의 작업을 저 수준에서 편집이 가능하다. 또한 이들은 저작 환경의 장면 트리 뷰를 통해 사용자에게 BIFS의 구조를 직접 보여주고 사용자는 이벤트에 따른 액션 저작을 장면 트리에 바로 삽입할 수 있는 BIFS 전문가용 시스템들로서, 일반 사용자가 사용하기 위해서는 BIFS 구조에 관한 연구가 필요하다. Harmonia는 MPEG-4 컨텐츠 저작의 전문성을 탈피하고 초보자를 중심으로 한 저작도구로 사용자에게 미리 정의된 템플릿을 제공하고 저작자는 이를 자신에게 맞게 변경하도록 하는 편리한 구조로 되어 있으나 템플릿 형식으로 제공되는 화면이 미리 정해져 있어 사용자에게 극히 제한적이다. envivio의 EBS는 사용자에게 BIFS 구조에 대한 선지식은 요구하지 않으나 컨텐츠를 구성하는데 필요한 모든 객체를 다른 도구에서 생성한 후, 이를 하나의 MPEG-4 컨텐츠로 조합하기 위한 환경만을 제공하므로 저작자는 컨텐츠 화면구성을 위해 2단계를 거쳐야 한다.

본 논문에서 제안하는 방법은 사용자가 BIFS에 대한 전문적 지식이 없어도 도구에서 미디어 동기를 사용자가 직접 정의해서 멀티미디어 제작을 쉽게 할 수 있도록 하는 것을 목표로 한다. 동기화에 필요한 노드를 BIFS에 추가하고 저작도구에서 미디어간 시간 동기화 인터페이스를 제공하며, 시간별로 지정된 미디어로 통합된 멀티미디어를 BIFS 구조로 자동 생성시켜 전송하도록 구성한다.

3. 멀티미디어 동기화 모델

멀티미디어에서의 동기화란 미디어 객체들의 상영에 있어서 미디어들의 시간적 동기화와 공간적 동기화를 포함한다. 이 동기화를 통해서 각 객체의 시간적 위치와 공간상에서의 서로간의 관계를 파악할 수 있기 때문에, 저작자에게는 다양한 멀티미디어 문서를 제작하여 제공할 수 있는 환경을 제공하고 사용자에게는 프리젠테이션 되는 멀티미디어 문서를 사용자 의사대로 조작이 가능하도록 하는 환경을 제공할 수 있다.

멀티미디어의 시간 관계를 표현하는데 있어서 자주 사용하는 개념은 시간 간격(temporal interval)이다. 이것은 두 개의 종단점(endpoint) 혹은 인스턴스에 의해 특정 지어진 시간 구간으로 구성된다. 시간 인스턴스는 시간의 한 순간을 의미하며 시간 간격(temporal interval)은 두개의 시간 인스턴스에 의해 정의되어지는 시간상의 기간(duration)을 나타낸다.



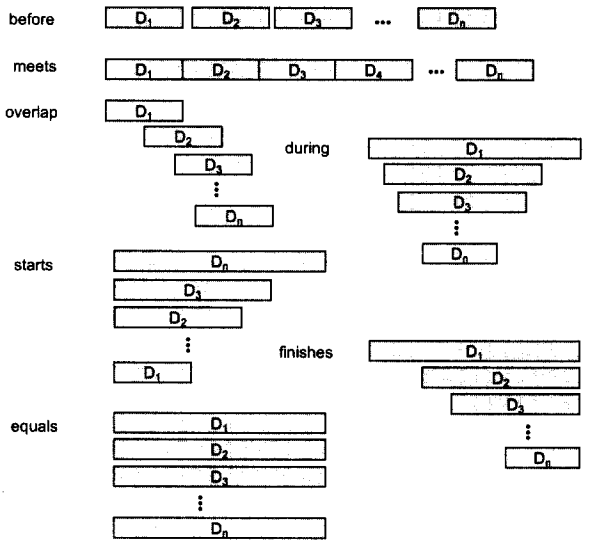
(그림 1) 객체 간 시간관계

두개의 인스턴스 객체 a와 b가 있을 때 얻게 되는 정보는 시작시간(S_a, S_b), 재생시간(D_a, D_b)과 종료시간이다. 이들 객체간의 상대적인 위치는 전체 재생시간(T_r)에서 객체간의 간격(D_δ)으로 얻을 수 있다. (그림 1)은 두 객체 간 시간 관계를 나타낸다. 멀티미디어 컨텐츠는 여러 종류의 복합적인 객체 집합으로 표현된다. 따라서 다중 객체에 대한 시간관계 모델이 필요하다. n개의 시간간격의 순서화된 집합을 P라 하고 $P = \{P_1, P_2, P_3, \cdots, P_n\}$ 으로 나타낸다. R은 시간관계를 나타내며 n차의 시간관계는 R^n으로 표기하고

다음의 관계를 만족해야 한다.

$$P_i \; R \; P_{i+1} \;\; \forall_i \; (1 \leq i < n)$$

만일 n=2라 하면 n차 시간관계 모델은 $P_1 \; R \; P_2$가 되며 각각의 객체는 시작시간(S_i), 기간(D_i), 그리고 종료시간을 표시한다. 객체 간 상대적 위치와 시간의 종속성은 전체기간($\hat{T_R}$)에서 지연시간(D_δ)에 의해 얻어진다. 이들 정보를 기반으로 여러 멀티미디어 객체 간 위치와 재생되는 시간에 따라 그 특징에 맞는 관계모델을 설정할 수 있다. (그림 2)는 시간 관계에 따른 모델을 보여주고 있다. before는 여러 객체가 일정 간격을 두고 순차적 순서를 갖는 경우이고 meets는 간격이 없는 순차적 모델이다. overlap은 하나의 객체가 재생될 때 간격을 두고 다음 객체가 재생되는 경우이며 during은 첫 번째 객체가 재생되는 동안 나머지 객체들의 재생과 종료를 하는 경우이다. starts와 finishes는 각각 객체들의 시작시간과 종료시간이 같은 경우이며 마지막으로 equals는 시작시간과 종료시간이 모두 같은 경우이다.



(그림 2) 다중 미디어 시간관계 모델

이 모델의 특징은 미디어간의 시간 관계를 객체 수준에서 기술할 수 있으며 비디오, 오디오, 텍스트 등의 객체를 가진 멀티미디어 응용에서의 동기화 관계를 나타낼 수 있다는 것이다. 본 논문의 목적은 이러한 각각의 시간 관계모델을 BIFS노드 형식으로 정의하고 각 모델에 맞는 시간정보를 저장하여 멀티미디어 객체들 간의 시간을 동기화 시킬 수 있는 BIFS로 확장시키는 것이다.

4. 시간 정보 관리 노드

본 절에서는 객체의 시간 정보를 관리하기위해 앞에서 기술한 시간모델을 구성하는데 필요한 정보를 저장하는 노드를 새로 정의한다. 이 노드들은 BIFS 트리구조의 적절한 위치에서 각 객체들의 시간을 관리할 수 있도록 기능이 확장된다.

BIFS에는 기본적으로 각 멀티미디어 객체를 제어하고 모니터링 하는 노드와 여러 객체들을 관리하고 객체의 시간적인 변화를 적용시키는 노드를 제공하고 있다[7, 11]. MediaContorl 노드는 재생, 반복, 소리 비활성 등의 객체 제어를 담당하는 노드이며 MediaSensor 노드는 재생 중에 있는 객체의 활성 및 비활성을 제어한다. TemporalGroup 노드는 화면을 구성하는 객체들의 동시시작 및 동시종료에 대한 정보를 저장하며 TemporalTransform 노드는 재생 중인 객체의 변화되는 정보들을 저장한다.

이러한 노드들은 모두 객체의 시간 정보를 다루고 있으나 하나의 객체에 대한 시작시간, 종료시간 및 재생시간 정보에 국한된다. 이러한 정보들은 자신 외에 다른 객체들과의 연동기능이 미흡하여 재생되는 화면 구성의 다양성을 제공하지 못한다. 본 연구에서는 화면을 구성하는 객체들간의 시간적인 위치정보를 저장하고 관리하기 위한 노드를 새로 정의하고 시간관리 방법을 제안한다. 본 논문에서 시간관리 노드 정의를 위해 새롭게 추가되는 노드는 전체 객체의 시간 관계를 관리하는 SyncObject 노드, 서브그룹을 관리하는 DelayTime 노드, 그리고 실제 객체의 시간정보를 저장하는 ObjectTime 노드로 나누어진다.

SyncObject 노드는 시간모델을 관리하기 위한 최상위 노드이며 필드의 속성은 다음과 같다.

field	SFString	TimeModel	""
exposedField	MFNode	children	[]
exposedField	SFString	TimeConst	""
exposedField	SFTime	totalDuration	0
exposedField	SFTime	firstDuration	0

〈표 1〉 시간모델 노드 생성조건

Node	생성조건 $D^i (1 \leq i < n)$
MediaBefore	$< D^i_\delta$
MediaMeets	D^i_δ
MediaOverlaps	$< D^{i+1}_\delta + D^i_\delta$
MediaDuring	$> D^{i+1}_\delta + D^i_\delta$
MediaStarts	$< D^{i+1}_\delta$, $(D^i_\delta = 0)$
MediaFinishes	$D^{i+1}_\delta + D^i_\delta$
MediaEquals	D^{i+1}_δ, $(D^i_\delta = 0)$

SyncObject 노드는 3장에서 설명한 시간모델인 MediaBefore, MediaMeets, MediaOverlap, MediaDuring, MediaStarts, MediaFinishes, MediaEquals로 세분화된 정보를 기록한다. 이들 노드의 생성조건은 각 객체별로 지정된 시작시간(S_i)과 재생시간(D_i)의 관계와 객체 사이의 지연시간(D_δ)

에 따라 결정되며 노드별 생성조건은 <표 1>과 같다.

생성조건에 따라 선택된 시간모델명은 TimeModel 필드에 저장되고 해당 모델의 생성조건은 TimeConst에 저장되며 화면이 재생되면서 전체 재생시간이 기록된 totalDuration 필드 정보와 함께 children 필드의 하위 객체들이 지정된 시간 모델을 유지시킨다.

DelayTime 노드는 객체의 속성을 구성하는 서브그룹을 관리하고 객체간 지연시간과 현재 객체의 경과시간을 체크하는 노드로서 이를 구성하는 필드의 속성은 다음과 같다.

exposedField	SFTime	delayTime	0
exposedField	SFTime	offset	0
exposedField	MFNode	children	[]

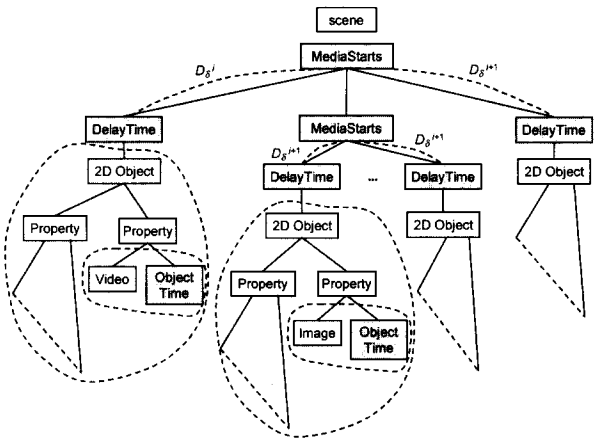
DelayTime 노드는 생성된 시간모델을 기준으로 객체와 객체 사이의 지연시간을 delayTime 필드에 기록하고 객체의 속성을 표현하는 서브그룹을 children 필드에서 관리한다. 여기에 재생 중인 객체가 현재까지 재생된 경과시간을 기록하는 offset 필드를 포함하여 다음에 재생되어야 할 객체의 시간적인 위치를 파악하도록 한다.

ObjectTime 노드는 실제 객체의 시작시간, 재생시간 등을 관리하는 노드로서 필드속성은 다음과 같다.

exposedField	SFTime	startTime	0
exposedField	SFTime	endTime	0
exposedField	SFTime	duration	0
exposedField	SFTime	currentTime	0

ObjectTime 노드는 실제 멀티미디어 객체의 시작시간과 종료시간 및 재생시간 정보를 저장하며 이를 이용하여 중간 노드인 DelayTime 노드의 offset 필드와 delayTime 필드의 정보를 갱신한다.

BIFS 트리에 시간 관계 모델 노드를 추가하면 (그림 3)과 같다. 짙게 표현된 노드가 기존의 BIFS 트리구조에 새롭게 추가된 시간 정보 노드들이다. 시간에 대한 그룹 노드는 현재 해당하는 동기화 모델을 표시하며 전체 노드를 관리한다. 각 객체의 지연시간 정보를 갖는 노드가 위치하여 지연시간과 객체의 재생시간을 관리한다. 최하위 노드의 시간 정보는 해당 객체의 시작시간과 종료시간을 기록한다. 시간정보를 이렇게 그룹노드와 중간노드 그리고 하위 노드로 구분지어 놓은 이유는 구성된 노드들을 객체 단위별로 관리하기 위해서이다. BIFS는 하나의 장면을 구성하기 위해 화면의 구성, 객체의 위치 및 속성 등 여러 가지 정보들을 노드에 포함한다. 그러므로 하나의 객체를 표현하기 위해서는 객체 뿐 아니라 그에 따른 속성노드까지 포함이 되는 하나의 소단위의 하위 트리 구조를 가지게 된다. 그렇기 때문에 하나의 화면에 포함되는 객체들 간의 시간 관계는 객체 노드에만 국한될 수 없고, 객체를 표현하는 노드들을 포함한 소그룹별로 관리가 이루어져야 한다. 또한 이러한 구조를 통해서 사용자에 의한 이벤트가 발생하여 기존의 시간 관계에 변경이 가해지는 경우, 예를 들어 재생시간의 변경이나 순서가 변경되는 경우 BIFS의 노드 구성을 수정하는 것이 용이하다. 즉, 재생시간이나 순서가 변경된 객체의 시간 정보를 변경시키고 그 상위노드의 시간 정보를 변경시켜 시간 순서에 맞게 중간 시간 노드의 위치 수정과 하위 노드의 포인터만 서로 변경시켜주기 때문에 화면의 재생 중에 발생할 수 있는 노드의 수정을 최대로 줄일 수 있다.



(그림 3) 시간관계 노드 추가

5. BIFS노드 생성모듈

본 절에서는 화면에 표현된 객체들의 BIFS 노드를 생성하는 방법과 종속관계에 따른 속성 노드들의 생성규칙에 대하여 기술한다. 각 객체는 위치, 재질 및 모양, 스타일 등의 정보를 필요로 하며 이들을 그룹화하기 위한 그룹노드와 앞에서 정의한 객체간 시간관리 노드가 포함된다.

<표 2> MPEG4 저작에 사용된 노드 리스트

구 분	BIFS 노드 이름
그 룹	Group
위치 및 재질	Transform2D Shape Appearance Material2D
객체 스타일	MovieTexture ImageTexture Text FontStyle
시간 관리	SyncObject DelayTime ObjectTime

<표 2>는 본 연구에서 객체를 구성하기 위해 사용한 BIFS 노드의 목록을 나타낸다. Group 노드는 BIFS 트리 구조의 최상위에 위치하여 나머지 하위 노드를 관리한다. Transform2D 노드는 객체의 위치를 저장하며, Material2D 노드는 객체모양 및 색상정보를 기억하고 Appearance 노드의 하위에 위치해 객체가 화면에 보여지도록 하며 Shape노드는 Appearance 노드와 Meterial2D 노드의 상위노드로 이들을 관리한다. 각 노드들은 정보를 표현하는데 필요한 필드들을 가지고 있으며 이들은 모두 각 필드속성과 함께 클래스로 구성하였다. (그림 4)는 MovieTexture 노드의 자료구조를 나타낸다. 각 노드의 자료구조는 필드 속성에 따른 변수들과 각 노드의 성격에 따라 적절한 수행 함수들을 포함한다.

```
class MovieTexture
{
public :
  MovieTexture (CMovieObject* pMO) ;
  CMovieObject* GetMovieObj() const { return (Movie2D*)
GetObject() ; }
private :
  BOOL m_firstTimeAtStartTime ;
  BOOL m_firstTimeAtStopTime ;
  BOOL m_endOfTimeReached ;
  double m_startTime ;
  int m_timesStarted ;
} ;
```

(그림 4) MovieTexture의 자료구조

```
if(AVO_List → IsObject()) exit(0) ;
else{
    //Root 노드를 생성하고 시간그룹노드와 연결
    Root = new Group() ;
    SyncObj = pTimeNode → GetSyncObj() ;
    Root → AddNode(SyncObj) ;
    while(ObjNum){
        objcnt = 0 ;
        //지연시간 관리노드를 시간그룹노드에 연결
        DelayTime = pTimeNode → GetDelayTime() ;
        SyncObj → AddNode(DelayTime) ;
        //각 객체의 서브그룹을 관리하는 Transform2D 노드 생성
        Transform2D *AVObjectGrp = new Transform2D() ;
        //Transform2D 노드를 지연시간 노드와 연결
        DelayTime → AddNode(AVObjectGrp) ;
        //객체의 타입에 맞는 하위 노드들을 생성
        if(AVO_List → Obj_Table[objcnt].O_Type == TEXT)
          GenTextProc(AVO_List) ;
        else
          GenVisualProc(AVO_List) ;
        objcnt++ ;
    }
}
```

(그림 5) BIFS 노드의 기본트리 생성 알고리즘

화면에 하나의 객체가 생성되면 객체의 위치, 크기, 시간 속성 및 경로 정보를 객체 리스트에 인덱스 값과 함께 저장한다. 화면에 모든 구성이 끝나면 시간모델 생성기에서 생성한 시간관리 노드를 로드하고 장면 객체에 대한 BIFS 기본 트리를 생성한다. BIFS 기본 트리는 Group 노드로 시작하는 Root에 SyncObject 시간 노드와 DelayTime 노드를 생성하고 각 객체별로 서브그룹을 생성하는 Transform2D 노드가 생성된다. 그 후 BIFS 생성규칙에 따라 객체 리스트의 정보와 BIFS 노드 리스트 정보를 토대로 객체를 표현하는 자세한 BIFS 트리를 생성하게 된다(그림 5).

하나의 객체가 표현하는 속성들은 여러 개의 노드들의 집합으로 구성되고 노드간의 종속규칙이 존재한다. 이러한 노드들이 트리구조로 완성되기 위해서는 각 노드들이 지정된 위치에서 종속관계를 유지해야 한다. BIFS노드 생성규칙은 화면에 구성되는 객체들의 특성에 따른 노드를 생성하고 이러한 노드들이 종속관계를 유지할 수 있도록 한 것이다. 규칙은 크게 텍스트와 비주얼 객체로 구분된다(<표 3>).

<표 3> BIFS 노드 생성규칙

화면구성 객체	생 성 규 칙
Text Object	1. Shape, Appearance, Material2D 노드 생성 2. Appearance 노드에 Material2D 노드 연결 3. Shape 노드에 Appearance 노드 연결 4. Text, FontStyle 노드 생성 5. Text 노드에 FontStyle 노드 연결 6. Shape 노드에 Geometry 노드(Text) 연결
Visual Object	1. Shape, Material2D, Appearance 노드 생성 2. Appearance 노드에 Material2D 노드 연결 3. if(Obj_type == Image), ImageTexture 노드 생성 if(Obj_type == Movie), MovieTexture 노드 생성 4. Appearance 노드에 비주얼 노드 연결 5. Shape 노드에 Appearance 노드 연결

Text 노드는 Shape 노드를 기본으로 해서 객체의 외형을 채우는 Appearance 노드와 객체의 고유 형태를 나타내는 노드로 구성된다. Text 노드의 경우는 글꼴의 속성을 지정하는 FontStyle 노드가 고유형태 노드로 사용된다. 비주얼 객체의 경우 Appearance 노드에 ImageTexture 혹은 MovieTexture 노드가 각각 추가됨으로써 어떤 내용이 표현되는지 구별하게 된다. 생성규칙에 따른 노드가 구성되면 이 서브 그룹은 이 그룹을 관리하는 해당 Transform2D 노드와 연결된다.

6. 장면구성 객체의 시간관리

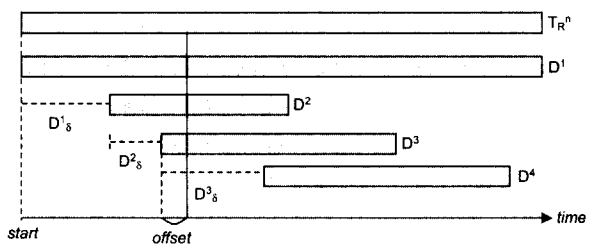
화면에 구성된 객체들이 지정된 시간모델에 의해 재생되기 위해서는 객체들의 시작시간과 재생시간 및 지연시간들을 파악하고 화면에 로딩시키기 위한 관리가 필요하다. 객

체 관리자는 BIFS 트리에 구성된 객체의 노드들에서 시간 정보를 찾아 기록하고 첫 번째 객체를 재생시킨다. 재생과 동시에 타이머를 작동하여 객체의 경과시간을 체크하여 해당 객체들의 서브 그룹을 담당하는 DelayTime 노드의 offset 필드에 정보를 기록한다. 이 정보는 바로 다음에 재생될 객체의 지연시간을 갱신해 주는 자료로 활용된다.

```
(a) If offset < D_i
        If offset < D^i_δ then
            DelayTime = D^i_δ - offset
(b) ElseIf offset == D^i_δ
        load next object
        offset = 0
```

(그림 6) 객체의 지연시간 갱신 알고리즘

(그림 6)는 객체의 지연시간 갱신 알고리즘을 나타낸다. 객체의 경과시간을 갖는 offset값이 해당 객체의 전체 재생시간보다 작을 경우(a) 객체는 현재 재생 중임을 말한다. 이 때 다음에 재생될 객체의 지연시간 내에 offset 값이 포함되면 다음 객체가 재생되기까지 어느 정도의 지연시간이 존재하는 경우이며 기존의 지연시간에서 현재 경과시간 만큼을 제외한 시간 값으로 갱신해 준다. (그림 6)(b)의 경우는 현재 객체의 재생이 완료되었다는 것을 의미하며 객체 관리자는 해당 객체를 화면 구성자에 로딩을 시킨 후 offset 값을 초기화하고 다음에 재생되는 객체에 대한 경과시간을 체크하게 된다. (그림 7)는 객체들의 전체시간 T_R^n과 D^1부터 D^4에 해당하는 각 객체의 재생시간을 나타낸다. offset은 현재 D^3의 경과시간을 기록하면서 다음에 재생될 D^4의 위치를 체크하는데 (그림 6)(a)의 경우를 보여주고 있다.



(그림 7) offset과 지연시간 관계

7. 구 현

본 절에서는 본 연구에서의 시간동기화 모델을 이용하여 MPEG-4 편집기와 재생기로 구성된 MPEG-4 저작시스템 구현에 대해 기술한다. 편집기는 편집창과 타임라인 바로 구분되어 사용자가 시청각 자료를 편집하여 화면을 구성하고 객체의 시간을 설정하도록 하였다. MPEG-4 재생기는 객체에 설정된 시간정보를 기준으로 각 객체를 재생기 화면에 입력된 시간에 맞게 출력시켜 주도록 하였다.

MPEG-4편집기는 비디오, 이미지, 텍스트의 시청각 객체를 편집할 수 있는 편집창을 통해 객체를 편집한다(그림 8). 편집 창에서는 객체의 공간적인 위치와 재생시간을 편집할 수 있다. 편집화면에서 구성된 시간정보와 공간 구성정보는 객체 로더를 통해 각각 시간모델 생성기와 장면 트리 생성기에 전달된다. 시간모델 생성기는 전달받은 객체들의 시간정보를 시간모델 생성조건과 비교하여 해당되는 모델에 따른 시간관리 노드를 생성하며 하나의 그룹노드의 n개 객체에 대한 지연시간 노드와 객체시간정보 노드들로 구성된다. 장면트리 생성기는 객체 로더로부터 전달받은 정보를 객체 리스트에 저장하고 BIFS 노드 생성규칙에 따라 객체에 필요한 노드를 생성하고 시간 노드들과 함께 BIFS 장면트리를 생성하게 된다. 인코더는 장면트리 생성기에서 구성된 BIFS 트리구조를 노드정보와 객체 기술자(Object Descriptor) 정보로 분리한 후 MPEG-4 파일 포맷으로 인코딩 한다. 인코딩 된 파일은 재생기의 디코더에 의해 .bif 확장자를 갖는 스크립트 파일을 생성한다.

(그림 8) MPEG-4 편집기

장면 관리자는 .bif 확장자의 파일을 분석해서 화면 재구성에 필요한 BIFS 트리를 구성한다. 객체 관리자는 장면 관리자에서 시간정보를 추출하여 관리하고 객체의 재생 경과시간에 따라 다음에 재생되는 객체의 지연시간을 갱신하면서 화면에 표현할 객체들을 로딩하는 기능을 담당한다.

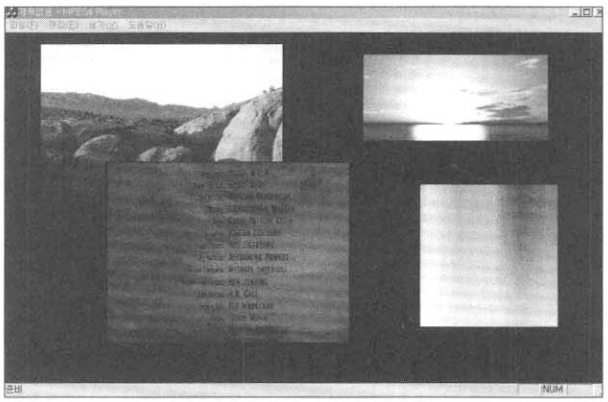
편집기는 동영상, 이미지, 텍스트 등의 데이터를 입력받아서 객체들을 시각적으로 편집할 수 있으며 로딩 된 객체를 자유롭게 이동시킬 수 있다. 이 객체에는 기본적으로 화면상의 위치, 모양, 재질 등의 속성정보들을 저장하게 된다. 편집되는 객체들은 타임라인 바를 통해 시간이 설정되며 사용자는 객체별로 재생시간을 설정해 줄 수 있다. 타임라인 바를 통해 객체들의 실행시간을 확인할 수 있으며 객체의 재생시간을 어떻게 입력하느냐에 따라 장면에서 객체들

간 동기화 되는 시간 관계 모델을 조절할 수 있다. 모든 객체의 시간정보가 입력되면 시간모델별 조건을 검색하여 그에 따른 시간관계 노드를 구성하고 BIFS 노드를 생성하여 BIFS 장면 트리를 구성하는 내부 작업을 수행하게 된다.

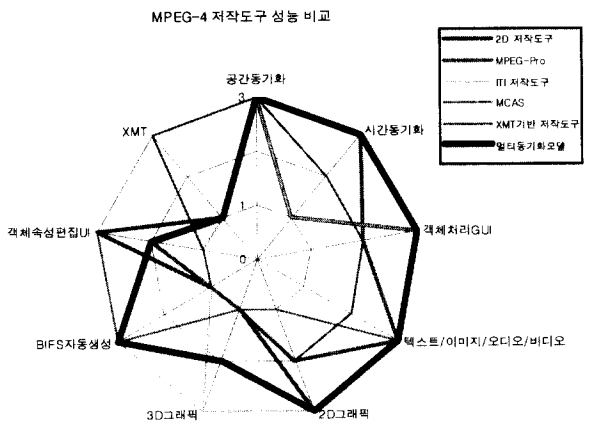
MPEG-4 재생기는 편집된 객체들을 재구성하고 입력된 시간정보에 맞게 객체들을 재생한다(그림 9). 이 재생기는 MPEG-4 표준 제정 그룹의 참조 라이브러리인 IM1-2D 재생기를 이용하였다[11].

〈표 4〉 MPEG-4 저작도구 기능 비교

종류 항목	2D MPEG-4 저작도구[2]	MPEG-Pro[1]	ITI MPEG-4 저작도구[15]	MCAS[19]	XMT기반 MPEG-4 저작도구[9]	멀티 동기화 모델 [본 논문]
멀티미디어 공간 동기화	○	○	○	○	O	O
멀티미디어 시간 동기화	×	○	×	○	△	O
객체처리 GUI	○	△	○	○	△	O
텍스트/이미지 오디오/비디오	○	○	○	○	△	O
2D 그래픽	○	△	○	×	△	O
3D 그래픽	×	×	○	×	X	△
BIFS 자동생성	×	×	×	○	X	O
객체속성편집UI	○	△	×	○	X	△
XMT	×	×	×	×	O	X



(그림 9) MPEG-4 재생기



(그림 10) 저작도구의 성능

<표 4>는 본 시스템과 기존 시스템과의 기능을 비교한 표이다. 해당 기능이 없는 경우엔 X, 불분명하거나 기능이 미약한 경우엔 △, 있는 경우엔 O로 표시하였다. (그림 10)은 각 MPEG-4 저작도구의 성능의 분포를 그래프로 나타낸 것이다. 각 기능의 값은 1~3으로 구분하였고 각 성능은 동일한 가중치를 갖는 것으로 가정하여 표시하였다.

8. 결 론

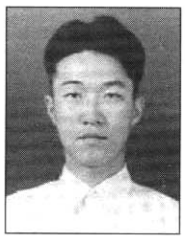
본 논문에서는 인터넷 멀티미디어 응용 개발에서 요구되는 멀티미디어 동기화 모델을 정의하고 이를 이용한 저작 시스템 구현에 대해 기술하였다. 이 모델에서는 MPEG-4 시스템의 BIFS 노드에 추가로 동기화에 필요한 새로운 노드들과 그들간의 관계를 정의하였다. 본 시스템의 개발에는 Windows 2000 환경에 Visual C++ 6.0과 DirectX의 DirectDraw 라이브러리를 이용하였다. 본 시스템은 BIFS 노드와 트리구조를 내부적으로 자동 생성하므로 기존의 편집도구와 같이 BIFS로 표현된 객체 노드를 직접 편집할 필요가 없고 BIFS에 대한 별도의 지식을 필요로 하지 않는다는 장점을 가진다.

향후 연구과제로는 편집도구에 객체 이벤트를 설정할 수 있도록 이벤트 관련노드를 추가하고 정의된 시간관리 노드에 이벤트 속성을 갖는 필드를 추가하여 재생기에서 재생될 수 있도록 연결 기능을 설정하는 것이다. 또한 작성된 MPEG-4 컨텐츠를 네트워크를 통해 효율적으로 전송하기 위한 스트림 프로파일을 추가하는 작업을 할 예정이다.

참 고 문 헌

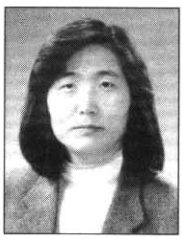
[1] Boughoufalah S., Dufourd J-C., Bouilhaguet F., "MPEG-Pro, an Authoring System for MPEG-4 with Temporal Constraints and Template Guided Editing," Proceedings of

ICME2000, Vol.1, pp.175-178, 2000.
[2] D. W. Viljoen, A. P. Calitz, N. L. O. Cowley, "A 2-D MPEG-4 multimedia authoring tool," Proceedings of the 2nd international conference on Computer graphics, virtual Reality, visualisation and interaction in Africa, pp.151-160, 2003.
[3] Florian Mueller, "mediacaptain - an interface for browsing streaming media," Proceedings of ACM Multimedia Conference2000, pp.419-421, 2000.
[4] http://www.envivio.com.
[5] http://www.w3c.org.
[6] httpL//www.telecomitalialab.com.
[7] Information technology-Coding of audio-visual objects Part 1 : Systems, ISO/IEC 14496-1, Jan., 2001.
[8] K. Asrar Haghighi, Y. Pourmohammadi and H. M. Alnuweiri, "Realizing MPEG-4 Streaming Over the Internet : A Client and Server Architecture using DMIF," Proceedings of International Conference on Information Technology : Coding and Computing (ITCC '01), pp.23-29, 2001.
[9] Kwang-Yong Kim, Hyun-Cheol Kim, Won-Sik Cheong, Kyuheon Kim, "Design and Implementation of MPEG-4 Authoring Tool," Proceedings of the 4th International Conference on Computational Intelligence and Multimedia Applications(ICCIMA'01), 2001.
[10] MPEG-4 Tools by ENST, http://www.comelec.enst.fr.
[11] Overview of the MPEG-4 Standard, http://mpeg.telecom-italiallab.com/standards/mpeg-4/mpeg-4.htm.
[12] The Virtual Reality Modeling Language, ISO/IEC 14772-1 : 14772-1, http://www.vrml.org/technicalinfo/specifications/vrml97, 1997.
[13] Tomas D. C. Little, Arif Ghafoor, "Interval-Based conceptual Models for Time-Dependent Multimedia Data," IEEE Transactions On Knowledg and Data Engineering, Vol.5, No.4, August, 1993.
[14] http://uranus.ee.auth.gr/pened99/Demos/Authoring_Tool/authoring_tool.html.
[15] P. Daras, I. Kompatsiaris, T. Raptis and M. G. Strintzis, "MPEG-4 Authoring Tool for Composition of 3D Audio-visual Scenes," Proceedings of Second International Workshop on Digital and Computational Video (DCV'01), pp. 110-117, 2001.
[16] S. Boughoufalah, J. C. Dufourd and F. Bouilhaguet, "MPEG-Pro, an Authoring System for MPEG-4," ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 2000.
[17] Yasser Pourmohammadi-Fallah, Kambiz Asrar-Haghighi, and Hussein Alnuweiri, "Internet Delivery of MPEG-4 Object-Based Multimedia," IEEE Multimedia, Vol.10, No. 3, pp.68-78, 2003.
[18] 김준기, 홍성수, 이호석, "실시간 고압축 MPEG-4 비디오 코딩을 위한 전처리 시스템", 정보과학회논문지 : 컴퓨팅의 실제, 제9권 제5호, pp.509-520, 2003.
[19] 배수영, 김상욱, 마평수, "MPEG-4 저작 시스템에서 BIFS 생성 모듈", 정보과학회논문지 : 컴퓨팅의 실제, 제8권 제5호, pp.520-529, 2002.
[20] 이수철, 황인준, "동기화된 멀티미디어 프레젠테이션을 위한 XHTML 확장", 정보처리학회논문지B, 제8-B권 제6호, pp. 717-724, 2001.



성 승 규

e-mail : marquez@suwon.ac.kr
2001년 수원대학교 전자계산학과(학사)
2003년 수원대학교 대학원 컴퓨터과학과 (석사)
관심분야 : 데이터베이스, 네트워크, 멀티미디어



이 명 원

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