

특집논문-01-6-3-02

MPEG-4 표준을 이용한 멀티미디어 스트리밍 시스템 구현

임 동근*, 이 정 우*, 김 선 태**, 마 평 수**, 호 요 성*

Implementation of A Multimedia Streaming System using MPEG-4

Dong-Keun Lim*, Jeong-Woo Lee*, Sun-Tae Kim**, Pyeong-Soo Mah** and Yo-Sung Ho*

요 약

최근 멀티미디어 분야의 연구개발 동향은 주로 비디오와 오디오 신호의 개별적인 부호화에 치중하던 이전의 기술적 동향에서 벗어나, 비디오, 오디오, 텍스트 등과 같은 여러 종류의 신호에 대하여 시간상 동기일치를 고려하여 다중화하는 시스템 기술의 중요성이 더욱 증가하고 있다. MPEG-4 영상압축 표준은 영상의 내용에 대한 이해를 바탕으로 내용기반 부호화에 중점을 두고 있다. 이 내용기반 부호화에서는 각 미디어 데이터 자체를 객체 단위로 나누어 처리할 수 있기 때문에 사용자의 의도에 따라 다양한 형태의 조작과 디스플레이가 가능하며 그 응용 분야가 매우 넓다. 본 논문에서는 MPEG-4 표준의 시스템 규격과 파일 포맷에 따라 구현된 멀티미디어 서비스 시스템을 기술하였다. 구현된 기술은 멀티미디어 데이터의 스트리밍, DVD, 영상전자메일, 디지털 방송과 같은 다양한 멀티미디어 서비스에 응용될 수 있는 기반 기술로서 그 중요성을 가진다.

Abstract

In recent days, research activities on multimedia services mainly focus on the multiplexing system with timing synchronization for media components, such as video, audio and text. The MPEG-4 standard emphasizes object-based coding which includes analysis and understanding of the image content. Since in MPEG-4 we can define objects and encode them independently, we can manipulate and display each object for different applications. This feature of MPEG-4 is also very useful for multimedia services, such as video streaming via different network channels, digital versatile disc, internet TV, video E-mail, and so on. In this paper, we implement a multimedia streaming system which is compliant with the MPEG-4 system and the MP4 file format.

I. 서 론

최근 멀티미디어 서비스에 대한 요구가 크게 늘어남에 따라 멀티미디어 통신의 핵심 기술인 초저속 영상/음성 신호 부호화 방식에 많은 관심이 모아지고 있다. 기존의

음성 신호 뿐만 아니라 영상 신호까지 포함한 방대한 정보량을 가지는 멀티미디어 신호를 유선 통신망이나 무선 채널을 통해 효율적으로 전송하기 위해서는 디지털 신호의 압축 처리가 절대적으로 필요하다.

기존의 영상 부호화 표준들은 영상 데이터를 블록 단위로 처리하기 때문에, 초저속 영상통신을 위해 데이터 압축 비율을 크게 높일 경우에 블록화 현상이나 경계흐림 현상 등의 문제가 발생한다. 따라서 MPEG-4 표준에서는 기존의 영상 부호화 방식과는 다르게 객체 기반(Object-based) 부호화 기법을 도입하였다. 아울러, 기존의 부호화 방식에서 부호화 압축율과 재생 화질을 동시

* 광주과학기술원 정보통신공학과
Kwangju Institute of Science and Technology (K-JIST)

** 한국전자통신연구원 정보가전응용연구팀
Electronics and Telecommunications Research Institute (ETRI)

※ 본 연구는 광주과학기술원(K-JIST) 초고속광네트워크연구센터(UFON)를 통한 한국과학재단 우수연구센터(ERC)와 교육부 두뇌한국21(BK21) 정보기술사업단의 지원에 의한 것입니다.

에 높이기 위한 여러 가지 연구들도 진행되고 있다. 이와 같은 초저속 영상부호화 방식에 대한 연구 개발은 현재 전개되고 있는 멀티미디어 통신 서비스의 기술 기반이 될 것이다.

현재 국내외의 여러 연구 기관들과 회사들은 MPEG-1, MPEG-2 또는 MPEG-4 기술을 이용한 멀티미디어 서비스 개발에 많은 노력을 기울이고 있지만^{17,5)}, 이러한 기술들은 아직 그렇게 활발히 이용되지 못하고 있는 상태이다. 국내 실정은 기술적으로는 약간 부족한 상태이지만, 외국 기술을 이용한 다양한 형태의 멀티미디어 서비스들이 이미 소개되고 있다. 특히, 현재 그 사용이 날로 급증하고 있는 인터넷이나 이동통신과 같은 제한된 용량의 전송 채널을 통한 멀티미디어 통신 서비스들이 점차 늘어나고 있다.

현재 표준화 작업이 마무리되어 가고 있는 MPEG-4 표준은 인터넷이나 이동통신, 또는 초고속 정보통신망을 통한 다양한 멀티미디어 통신 서비스에 사용될 핵심적인 기술이다. 본 논문에서는 멀티미디어 서비스 시스템을 구축하기 위한 MPEG-4 표준 시스템 부분과 MP4 파일 포맷의 구현 방법을 설명한다. 또한 MPEG-4 영상/음성 부호화 방식과 MPEG-4 시스템 규격에 따라 구현된 멀티미디어 스트리밍 시스템을 기술한다^{17,3)}.

본 논문의 구성은 다음과 같다. 먼저 II장과 III장에서는 구현된 시스템을 이해하기 위한 MPEG-4 시스템 표준과 이를 구현하는 방법을 살펴보고, IV장에서는 MP4 파일 포맷에 대한 이해와 구현 방법을 설명한다. V장에서는 구현된 시스템에서 얻은 결과를 보여준다. VI장의 결론을 통해 구현된 시스템의 응용 분야를 고려한다.

II. MPEG-4 시스템 표준

1. MPEG-4 표준에서 객체의 의미

MPEG-4 표준³⁾에서 사용하는 오디오-비주얼 장면은 몇 개의 객체(object)로 구성되며 계층적 구조를 갖는다. 그림 1의 예에서 살펴보면, 화면에 보이는 장면은 2차원의 고정된 배경 화면, 배경 화면을 제외한 말하는 사람 모습, 말하고 있는 사람의 목소리, 컴퓨터 그래픽으로 그려진 책상과 지구본, 그리고 칠판에 쓰여지는 문자 등으로 개별 객체를 구분할 수 있다.

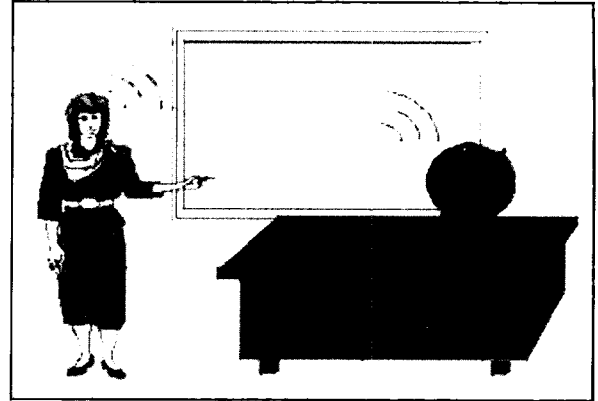


그림 1. MPEG-4 오디오 비주얼 장면
Fig. 1. MPEG-4 Audio-Visual Scene

MPEG-4 표준에서는 자연적인 혹은 인위적인 합성을 통해 생성되어 2차원이나 3차원으로 표현될 수 있는 여러 가지 객체를 정의한다. 앞에서 언급된 것처럼, 직접 보이거나 들리는 것 외에도 문자열과 그래픽, 말하는 사람의 머리 부분, 움직이는 사람의 몸, 그리고 움직이는 사람의 입 부분과 말소리를 동기시키기 위한 데이터들도 객체라고 정의할 수 있다.

부호화 효율을 높이고 원하는 기능들을 지원하기 위해 필요한 비트수만을 객체에 할당하여 부호화하며, 부호화된 객체들은 그들의 주위 환경이나 배경에 독립적이기 때문에 사용자에 의해 임의로 편집될 수 있다.

2. MPEG-4 표준의 내용

MPEG-4 표준에서는 대화형(interactive) 오디오-비주얼 화면으로 구성된 멀티미디어 정보를 처리하기 위한 목표 시스템을 설정하는데, 자연영상 혹은 합성영상, 오디오 혹은 비주얼, 2차원 혹은 3차원 객체의 부호화된 표현, 오디오-비주얼 객체의 상호관계와 시공간적인 위치의 부호화된 표현, 동기일치(synchronization), 인식(identification), 설명(description)과 비트열의 조합(association)과 같은 데이터 비트열의 관리와 관련된 정보의 부호화된 표현, 데이터 비트열 전송계층(delivery layer) 함수들과의 일반적인 인터페이스를 제공한다.

오디오-비주얼 화면들로 구성된 멀티미디어 정보의 통신을 위한 시스템의 전체 동작은 다음과 같다.

송신측에서는 오디오-비주얼 화면 정보를 각각의 부호화기를 이용하여 압축하고, 동기일치를 위해 시간 정보와 결합한다. 이와 같이 결합된 하나 이상의 부호화된 이진

비트열은 전송되거나 저장된다.

수신측에서는 수신된 비트열을 역다중화하여 복원한다. 오디오-비주얼 객체들은 화면구성정보(scene description, SO)와 동기일치 정보에 따라 재구성되어 사용자에게 제시된다. 사용자는 이 표현된 것에 따라 대화형 처리를 위한 상호작용을 할 수 있다. 상호작용 정보는 수신측에서만 처리될 수도 있고, 송신측으로 전송될 수도 있다. MPEG-4 시스템 부분은 자세한 복호화 처리 방법 뿐만 아니라, 이러한 화면 정보를 운반하는 비트열들의 구문구조(syntax)와 의미(semantics)를 정의하고 있다.

3. MPEG-4 시스템 구조

MPEG-4 시스템 부분에 규정된 표현 정보는 부호화된 오디오-비주얼 정보와 해당되는 화면구성정보에 대한 상호 연관된 오디오-비주얼 화면을 생성하는 방법을 설명한다.

초기설정 정보는 부호화된 내용(content)의 표현 형태인 하나 이상의 기초비트열(elementary stream)에 적용되도록 허용되어야 한다. 이러한 기초비트열의 일부는 MPEG-4 시스템 표준^[3]에 설명된 다중화 도구를 사용하여 묶인다.

기초비트열은 오디오 데이터 혹은 비주얼 데이터, 그리고 화면 구성 정보 중의 하나에 대한 부호화된 표현물을 포함한다. 기초비트열은 비트열을 인식하고, 비트열 사이의 상호 의존성을 설명하며, 비트열의 내용과 관련된 정보를 설명하는 정보 자체일 수도 있다. 각 기초비트열은 하나의 데이터 형태만을 포함한다. 기초비트열들은 그들의 비트열과 관련된 복호기를 사용하여 복원된다.

오디오-비주얼 객체들은 화면구성정보에 따라 편집되며, 터미널의 표현 장치에 출력된다. 이러한 동작들은 동기일치 계층(synchronization layer, sync layer)에서 제공되는 동기일치 정보를 이용하여 시스템 복호기 모델(systems decoder model, SDM)에 따라 동기가 일치된다.

시스템 부분은 비트열의 다중화를 위한 전송 계층(delivery layer), 비트열의 동기일치를 위한 동기일치 계층(sync layer), 그리고 압축 계층(compression layer)으로 이루어진다.

시스템 부분은 전송 계층인 DMIF^[3]로부터 FlexMux 패킷을 입력받은 후, 동기일치 계층인 SL 패킷들을 패킷 헤더(header)와 패킷 페이로드(payload)로 분리하고, 패킷 헤더에 있는 타이밍과 패킷화 정보를 이용하여 기초비트열을 분리한다. 압축 계층은 압축된 기초비트열 데이

터의 복호 과정을 위해 필요한 동작을 수행한다. 복원된 정보는 터미널상에 결합, 렌더링(rendering), 출력할 때에 사용된다.

4. 설명어 (Descriptor)

설명어들은 MPEG-4 시스템 부분에 규정된 각 기초비트열들 사이의 인식, 조합, 동기일치 등의 동작을 하기 위해 사용되는 메타 데이터(meta data)이다. 그림 2는 설명어들의 계층적인 구조를 보여준다.

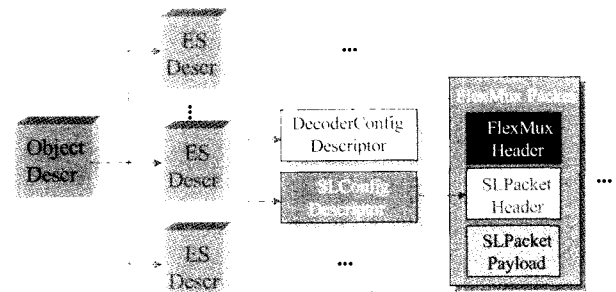


그림 2. 설명어의 계층적인 구조
Fig. 2. Descriptor Hierarchy

객체설명어 OD(object descriptor)는 개별적인 기초비트열과 그것의 성질을 설명하기 위해 그들 항목에 대한 다양한 추가적인 설명어들을 포함한다.

기초비트열 설명어 ESD(ES_Descriptor)는 특정한 기초비트열과 관련된 정보를 운반하며, 세 부분으로 이루어져 있다. 첫번째 부분은 ES_ID와 같이 ESD를 식별하기 위한 표시이며, 부모 OD와의 의존성을 설명하는 메커니즘이다. 두 번째 부분은 기초비트열이 요구하는 특성을 기술하는 해당 항목 설명어로 구성된다. 세 번째 부분은 미래의 확장이나 역방향 호환성(backward compatible)을 제공하기 위한 선택적인 설명어이다.

복호기 조건설정 설명어 DecoderConfigDescriptor는 복호기 형태와 해당 기초비트열이 요구하는 복호기 자원에 관한 정보를 제공한다. 이것은 수신단에서 해당 기초비트열을 복호할 수 있는지 여부를 결정하는데 필요하다.

SL 조건설정 설명어 SLConfigDescriptor는 동기일치를 위한 설정값으로서 SL 패킷 헤더(packet header)에서 참조하는 값을 전송하는데, 해당 변수(parameter)의 부호화 여부를 나타내는 플래그(flag)와 몇 비트 길이로 부호화할 것인지의 정보를 가지며, 그림 3과 같은 구조를 가진다.

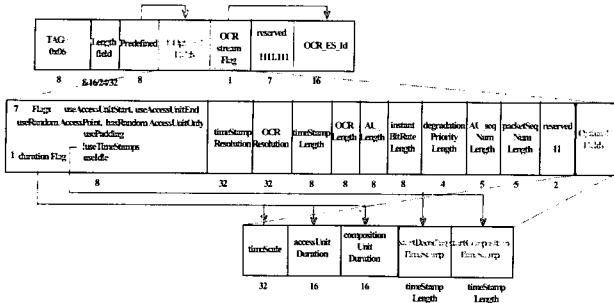


그림 3. SLConfigDescriptor의 구조
Fig. 3. SLConfigDescriptor Structure

SL 패킷은 헤더와 페이로드 부분으로 구성되는데, 헤더에서는 페이로드로 전송되는 데이터의 시간적인 관계와 연결에 관련한 정보를 포함한다. SL 패킷헤더는 각 개별적인 기초비트열의 요구사항에 따라서 설정된다. 변수들은 타임스탬프들과 클럭(clock) 참조의 존재 유무, 해상도, 시간의 정확도에 관한 사항을 포함한다. 그림 4는 SL 패킷의 구조를 설명하고 있다.

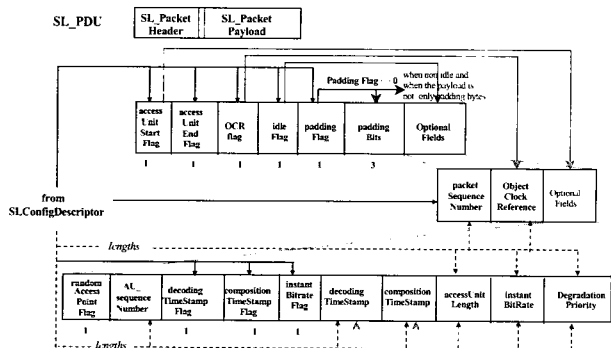


그림 4. SL 패킷의 구조
Fig. 4. SL Packet Structure

III. MPEG-4 시스템의 구현

1. 전체 멀티미디어 시스템의 구조

구현된 멀티미디어 서비스 시스템은 여러 참여 기관들이 각각 구현한 부분들을 결합시켜 그림 5와 같이 전체 시스템을 구성하였다. 본 논문에서는 이 중 멀티미디어 데이터의 압축과 전송에 중요한 역할을 하는 MPEG-4 시스템 부분의 구현 방법을 주로 설명한다.

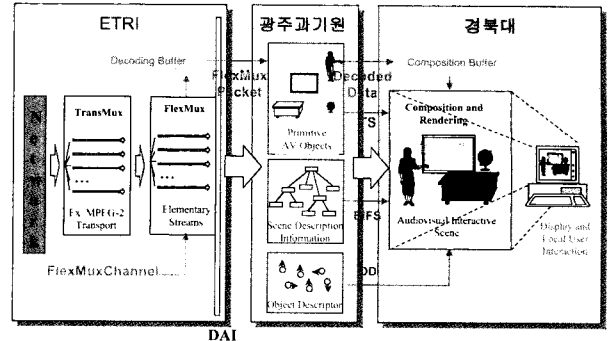


그림 5. 전체 멀티미디어 시스템
Fig. 5. Multimedia System

2. MPEG-4 시스템 구현 구조

그림 6은 복호기로 전송할 비트열을 만드는데 사용되는 FlexMux 비트열을 만드는 MPEG-4 부호화기의 동작을 개략적으로 보여주고 있다.

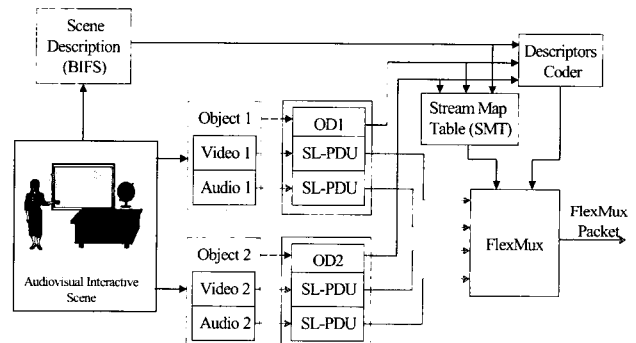


그림 6. MPEG-4 부호화기의 개략적 모델
Fig. 6. Simplified MPEG-4 Coder

그림 6에서 MPEG-4 시스템 부분은 각 객체를 구성하는 기초비트열들을 SL 패킷(SL-PDU)으로 만든다. 이 SL 패킷에는 AU(access unit)단위로 구성된 기초비트열 데이터, 타이밍 정보 DTS(decoding time stamp)와 CTS(composition time stamp), 그리고 각 객체를 복호화하는 시점을 나타내는 기준시간 OCR(object clock reference) 정보가 포함되어 있다. 이 값들은 부호화기에서 필요로 하는 개별 부호화기 블록에 공급된다. OCR 정보는 어느 객체가 복호화를 시작할 것인지를 나타내는 참조 시점으로, 해당 패킷에서 추출된 DTS나 CTS는 이 값과 비교하여 그 사용 시점을 결정한다. 기초비트열들은 비디오, 오디오, 텍스트, 그리고 화면구성정보가 될 수 있다.

MPEG-4 부호화기는 그림 7과 같이 구현되었다. 부호화하려는 비디오, 오디오, 텍스트 등의 여러 종류의 기초 비트열을 입력받아 조합할 때, 설명어 정보는 파일의 헤더 부분에 위치하며, 서로간의 동기를 맞추기 위한 타이밍 정보는 SL-PDU에 부가정보로서 추가된다.

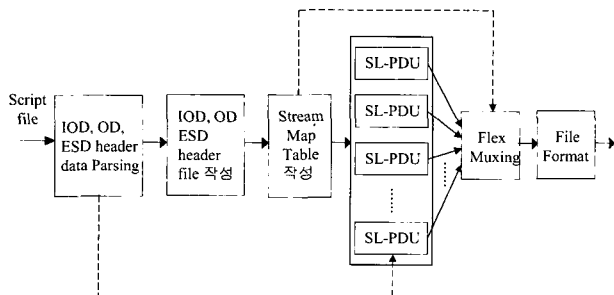


그림 7. MPEG-4 부호화의 구현
Fig. 7. MPEG-4 Encoder Implementation

이때 해당되는 부가정보를 조합하기 위한 기본 입력 파일이 필요하며, 이것은 스크립트 파일 형태를 가진다. 그 내용은 설명어에 삽입되는 내용에 대한 것과 SL 패킷을 만들 때 필요한 타이밍, 타임스탬프(DTS, CTS), 클록 참조(OCR) 등에 관한 사항을 가지고 있다.

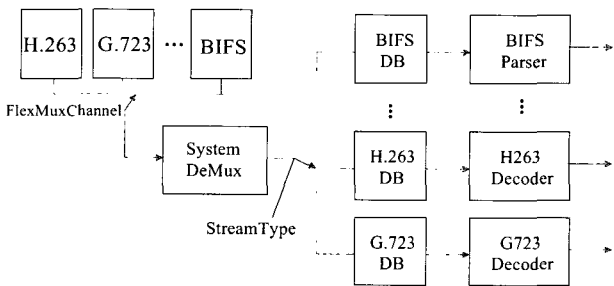


그림 8. MPEG-4 복호화의 구현
Fig. 8. MPEG-4 Decoder Implementation

MPEG-4 복호화기는 그림 8과 같이 구현되는데, 내부에 비디오, 오디오, 텍스트 등을 복호하는 개별 복호화기를 내장하고 있다. 네트워크를 통해 MPEG-4 복호화기, 즉 멀티미디어 시스템으로 전송되어온 비트열 속에 들어있는 다양한 형태의 비디오, 오디오, 텍스트와 같은 멀티미디어 자료를 기초비트열들로 구분하여 각각에 적합한 복호화기로 분배하여 원래의 비디오, 오디오, 텍스트 등으로 복원하는 역할을 한다.

그런데, H.263 혹은 G.723 등의 복호화기에서 필요로 하는 타이밍 정보와 AU를 임시로 저장하기 위해서는 시

스템 역다중화 블록과 복호화기 블록 사이에 디코딩 버퍼(decoding buffer, DB)가 필요하다. 이 버퍼에 들어있는 내용을 입력 받아 복호화기는 데이터 복원을 시작하고, 그 결과를 결합 버퍼(composition buffer, CB)에 저장하고, 화면구성정보 SD를 사용하여 화면의 해당 위치에 출력한다.

IV. MP4 파일 포맷의 구현

1. 스트리밍을 위한 파일 포맷의 필요성

일반적으로 멀티미디어 데이터를 스트리밍하기 위해서는 여러 개의 비트열을 공통 시간대로 동기화해야 한다. 그런데 WAV와 AVI 등과 같은 기존의 파일 포맷에는 멀티미디어 데이터에 스탬프가 없어 자체적으로 재생하는 데는 무리가 없지만, 스트리밍용으로는 적합하지 않다.

WAV 포맷 파일은 오디오 콘텐츠에만 사용할 수 있고, AVI 포맷 파일은 오디오와 비디오 콘텐츠에만 사용할 수 있다. 따라서 캡션과 정지영상, 스크립트 명령, 슬라이드 쇼까지 다양한 유형의 멀티미디어를 지원하는 차세대 디지털 미디어에는 새로운 포맷이 필요하다. Apple사의 QuickTime에 기능을 추가시킨 MP4 파일 포맷은 특히 확장성에 중점을 두고 설계한 것이기 때문에 앞으로 개발되는 새로운 유형의 멀티미디어 데이터도 지원할 수 있다⁴⁾.

2. 파일 포맷의 목적

MP4 파일 포맷은 미디어의 상호 교환, 관리, 편집, 화면 재생을 수월하게 하도록 유연하고 확장 가능한 형식으로 MPEG-4 미디어 데이터를 포함시킨 것이다. 화면재생은 화면 재생 데이터가 포함되어 있는 시스템 상에서 수행될 수도 있고, 네트워크나 TransMux와 같은 다른 비트열 전달 체계를 통해 수행될 수도 있다. 파일 포맷은 특정 TransMux와는 독립적으로 동작하도록 고안되어야 한다.

MP4 파일 포맷은 TransMux와 무관한 비트열의 전달, 효과적으로 TransMux를 지원할 수 있는 파일 포맷의 구현, 새로운 비트열을 삽입하거나 삭제하는 경우에 전체 화면 재생 데이터를 재다중화하지 않고 쉽게 처리할 수 있는 기능, 확장성, 다양한 형식의 상호 교환, 파일 전송, 화면 재생, 데이터 편집중에도 사용자가 시간에 따라 임의의 지점에 접근할 수 있는 기능을 제공한다.

MP4 파일 포맷은 다양한 분야에서 미디어의 상호 교환사용(interchange), 데이터의 저장, 비트열 전송을 위한 준비, 시스템 내에서의 화면 재생 등 다양한 분야에서 사용 가능한데, 구현된 시스템에서는 비트열 전송을 위한 준비 과정으로 사용된다.

3. 파일 포맷의 개략적인 전체 구조

개략적인 파일 포맷의 구조는 그림 9와 같다.

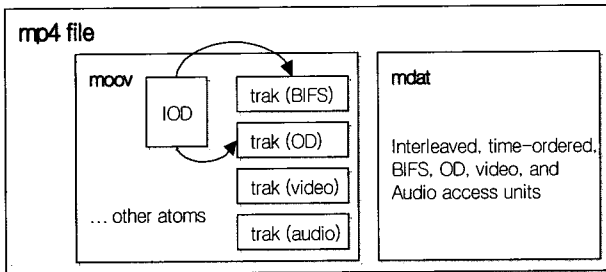


그림 9. 개략적인 파일 포맷 구조
Fig. 9. Simplified File Format

그림 9에서 파일 포맷의 구조는 크게 재생(presentation)과 관련된 메타 데이터 정보를 제공하는 moov와 실제 미디어에 대한 데이터를 제공하는 mdat로 나뉘어진다.

무비 아톰(movie atom)은 실제 재생되는 미디어 데이터에 관한 메타(meta) 데이터인 헤더 정보를 정의한다. 무비 헤더 아톰(movie header atom)은 전체 화면 구성에 관한 미디어의 정보를 정의하는데, 미디어 생성 시간(creation time), 수정 시간(modification time), 길이(duration) 등의 정보가 위치한다.

OD 아톰(object descriptor atom)은 장면구성정보 비트열(scene stream)을 위한 ESD와 OD를 위한 ESD를 포함한다. OD_ID, ESD 등과 같은 실제적인 설명어 정보가 위치한다.

미디어 트랙 아톰(media track atom)은 비디오, 오디오, 텍스트, 정지영상 등 각각의 미디어에 관한 정보를 정의하는 필드이다. 핸들러 참조 아톰(handler reference atom)과 미디어 정보 아톰(media information atom)으로 구성되며, 비디오와 오디오 등을 포함하는 미디어는 하나 이상의 트랙으로 구성된다. 각각의 트랙은 화면 재생 내에서 다른 트랙들과 독립이며, MP4 파일에는 적어도 하나의 미디어 트랙이 존재해야 한다. 또한 한 비트열내에 있는 미디어 데이터에 관한 정보를 선언하는 모든 객체들을 포함한다.

핸들러 참조 아톰은 비트열이 무엇에 의해 처리되는지를 정의한다. 즉, 핸들러의 형태가 비디오, 오디오, 힌트 트랙인지등을 정의한다. 미디어 정보 아톰은 비트열내에 있는 미디어의 특성 정보를 선언하는 필드로서, 미디어 정보 헤더 아톰(media information header atom)과 샘플 테이블 아톰(sample table atom)으로 구성된다.

미디어 정보 헤더 아톰은 미디어 핸들러 형태에 대응하는 각 트랙 형태에 관한 헤더 정보를 정의하는 부분으로 비디오 미디어 헤더 아톰(video media header atom), 사운드 미디어 헤더 아톰(sound media header atom), 힌트 미디어 헤더 아톰(hint media header atom), MPEG-4 미디어 헤더 아톰(MPEG-4 media header atom) 중의 하나이다.

미디어 데이터 아톰(media data atom)에 있는 MPEG-4 ES는 접근 단위(access unit, AU)로 저장된다. AU는 SL 패킷 비트열로 묶여지고, n 바이트 크기를 가진다.

사용자 데이터(user data)는 사용자 데이터에 대한 필드로서 오직 저작권(copyright)만을 표시한다.

4. 화면 재생 데이터 구조

하나의 MP4 파일은 전체 화면 재생을 위한 메타 데이터와 모든 미디어 데이터를 포함할 수도 있다. 하지만 하나의 MP4 파일은 미디어 트랙에 미디어 데이터를 반드시 포함해야 한다. 힌트(hint) 트랙들은 선택적으로 포함될 수 있지만, 힌트 트랙들만 포함되어서는 안 된다.

파일 내에서 미디어 데이터의 배치는 그림 10과 같이 된다. 각 미디어 데이터는 화면에 출력되는 순서로 차례대로 저장된다. 그리고 각 미디어 데이터에는 해당 미디어 부호화기에서 부호화된 비트열이 AU 단위로 저장되어 있다.

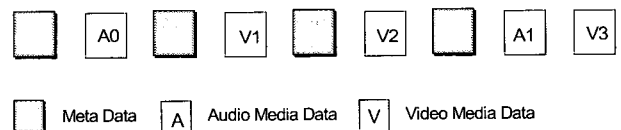


그림 10. 미디어 데이터의 배치
Fig. 10 Media Data Arrangement

트랙과 미디어 데이터 사이의 관계는 타이밍 정보를 포함하는 샘플 테이블에 의해 지정된다.

파일 구조 각각의 형태를 구분하기 위해 사용된 지정자는 표 1과 같다. 모든 지정자는 4바이트의 문자로 구성된다. 이 지정자들 중의 일부는 다른 지정자의 하위 계층을

구성하기도 한다. 또한 일부 지정자들은 병렬적인 구조로서 순서에 상관없이 묶여지게 된다.

표 1. 사용된 지정자와 의미
Table 1. Atoms and Meaning

지정자	의미
moov	Meta Data Atom
mdat	Media Data Atom
mvhd	Movie Header Atom
iods	Object Descriptor Atom
trak	Media Track Atom
mdhd	Media Header Atom
hdlr	Handler Reference Atom
minf	Media Information Atom
vmhd	Video Media Header Atom
smhd	Sound Media Header Atom
hmhd	Hint Media Header Atom
stbl	Sample Table Atom
stts	Composition Time To Sample Atom
stsd	Sample Description Atom
stsz	Sample Size Atoms
stsc	Sample To Chunk Atom
stco	Chunk Offset Atom

파일 포맷은 객체 구조를 가지므로 객체들의 연속으로 구성된다. 객체들의 일부는 다른 객체들을 포함할 수도 있다. 파일에 있는 객체들의 연속은 정확하게 하나의 화면 재생 메타 데이터 아톰(moov)을 포함해야 한다.

메타 데이터는 메타 데이터 아톰에 포함되고, 미디어 데이터는 같은 파일의 미디어 데이터 아톰이나 다른 파일의 미디어 데이터 아톰에 포함된다. AU, 미디어 데이터 객체 혹은 미디어 데이터 파일들로 구성되는 미디어 데이터는 참조되지 않는 다른 정보도 포함할 수 있다.

MP4 파일에서 사용되는 트랙 지정자들은 한 파일 내에서 유일해야 하므로, 임의의 두 개의 트랙이 같은 지정자를 가질 수 없다.

파일에 있는 각각의 기초 비트열은 미디어 트랙으로 저장된다. 이들 트랙을 위한 트랙 지정자는 4바이트로 구성되는데, 상위 2바이트는 0으로 채워지고 하위 2바이트는 ES_ID와 같은 값을 가진다. 힌트 트랙 지정자들은 기초 비트열 지정자로 인식되지 않도록 트랙 지정자의 값들보다 더 큰 값을 사용한다.

무비 헤더에 있는 다음 트랙 지정자의 값은 일반적으로 파일에서 발견된 가장 큰 트랙 지정자의 값보다 1이 크다. 이는 어떠한 상황에서도 트랙 지정자 쉽게 발생하도록 해준다. 하지만 이 값이 65535보다 같거나 큰 상태에서 새로운 미디어 트랙이 더해지면, 현재 파일에서 사용하고 있지 않은 값이 있는지를 검색해야 한다. 만약 기초비트열 지정

자와 같이 이미 값을 알고 있는 트랙 지정자를 트랙에 더 하려면, 먼저 그 파일에서 이미 사용하는 값과 충돌이 발생하지 않는지를 검사해야 한다. 힌트 트랙들은 다소 쉽게 값을 다시 지정할 수 있지만, ES_ID로 트랙 ID를 참조하는 미디어 트랙들은 다른 파일에서 이것을 참조할 수 있으므로 이를 고려해야 한다.

5. 파일 포맷의 동작 및 구현

파일포맷 생성기는 그림 11과 같이 구성된다. 비디오와 오디오 부호화기를 통하여 부호화된 미디어 데이터(AU)들을 결합시켜서 파일포맷으로 만드는데, 이때 각 미디어에 대한 설명어 정보와 시간 정보인 타임 스탬프(time stamp, TS)가 함께 사용된다. 생성된 파일포맷 데이터는 서버의 저장 공간에 저장된다.

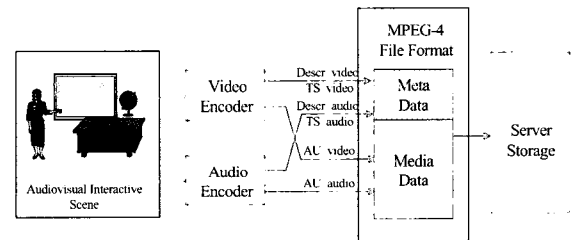


그림 11. 파일포맷 생성기의 구조
Fig. 11. Implementation of File Format Generator

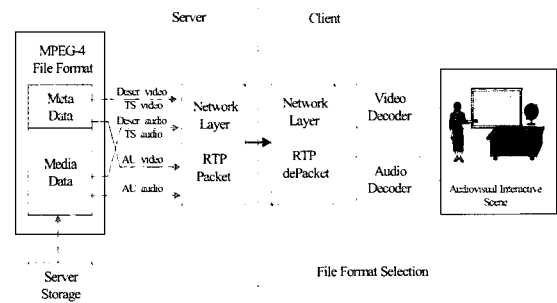


그림 12. 파일포맷 재생기의 구조
Fig. 12. File Format Decoder Implementation

그림 12와 같이 파일포맷 재생기는 클라이언트의 요구를 받아 서버의 저장 공간에 있는 해당 파일 포맷의 미디어 데이터에 있는 압축된 비디오와 오디오 데이터를 메타 데이터에 있는 시간 정보(TS)와 설명어 정보(Descr)를 이용하여 각각의 AU 단위로 분리하고 시간 정보에 따라서 이 데이터를 네트워크 전송단으로 보내어 해당 RTP 패킷으로 만들어서 클라이언트 쪽으로 전송하도록 한다.

V. 구현된 MP4 멀티미디어 스트리밍 시스템

본 작업에서 최종적으로 구현된 시스템의 화면 구조는 그림 13과 같다. 총 6개의 객체가 사용되는데, 비디오와 오디오, 정적 텍스트, 그리고 시간에 따라 변하는 동적 텍스트인 자막 부분으로 구성되어 있다. SD나 OD와 같은 일부 객체는 화면이나 음성으로 출력되지 않고, 각 미디어 데이터들의 연결이나 동기 일치에 사용되었다.

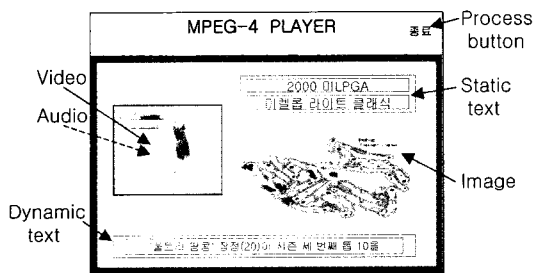


그림 13. MPEG-4 재생기 설계화면
Fig. 13. Design of MPEG-4 Player

그림 14부터 그림 17까지는 MPEG-4 파일 포맷으로 기존의 AVI 데이터를 변환하는 과정을 보여주고 있다. 실제로 구현된 부호화기는 AVI 데이터 뿐만 아니라 미디어 정보와 시간 정보를 가지는 모든 형태의 데이터에 대하여 파일 포맷을 만들 수 있다.

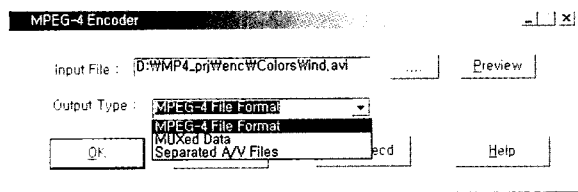


그림 14. MPEG-4 부호화기 입력화면
Fig. 14. Input of MPEG-4 Encoder

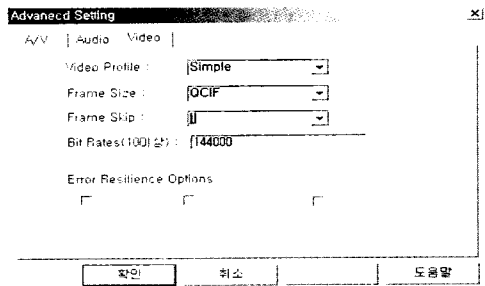


그림 15. MPEG-4 부호화기 입력 설정화면
Fig. 15. Advanced Setting of MPEG-4 Coder

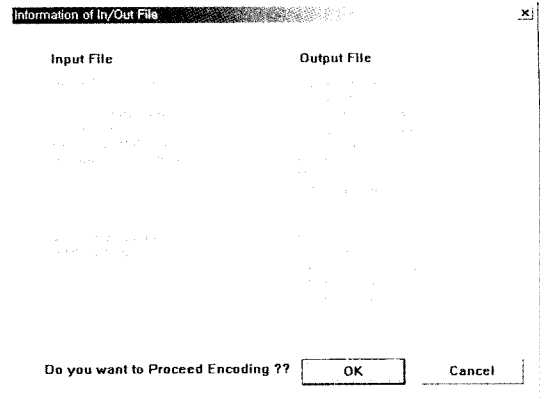


그림 16. MPEG-4 부호화기 설정내용 화면
Fig. 16. Configuration of MPEG-4 Coder

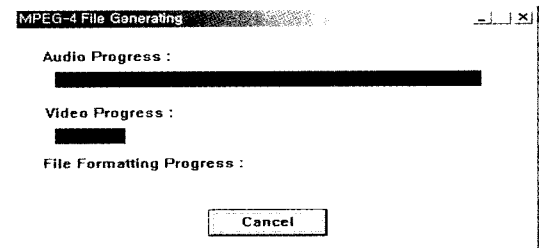


그림 17. MPEG-4 부호화기 실행화면
Fig. 17. Process Status of MPEG-4 Coder

그림 18과 그림 19는 비디오, 오디오, MPEG-4 파일 포맷으로 구성된 데이터를 입력받아 화면에 출력하는 결과를 보여준다. 그림 19에서 번진 듯한 부분은 영상이 잘못 처리되어 번진 것이 아니라, 원래의 화면이 꽃이 피어있는 위치에서 멀리있는 가차에 초점을 조정하였는데, 가까운 곳에 있는 꽃이 흐늘거리면서 카메라 앵글에 들어와서 번진 모습이다. 이 결과를 보여주는 화면은 네트워크로부터의 스트리밍된 데이터를 다룰 수 있을 뿐만 아니라, 로컬 디스크에 저장된 데이터도 출력하는 기능을 가지고 있다.

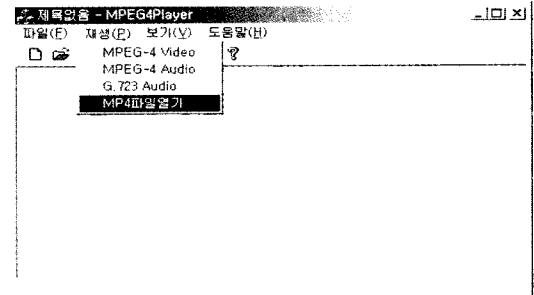


그림 18. MPEG-4 재생기 입력화면
Fig. 18. Input of MPEG-4 Player

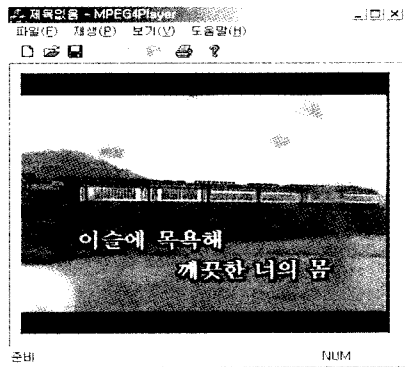


그림 19. MPEG-4 재생기 실행화면
Fig. 19. MPEG-4 Player Display

MPEG-4 시스템 표준을 사용한 멀티미디어 시스템은 MP4 파일포맷 표준을 사용한 더 큰 멀티미디어 시스템의 일부(MPEG-4 Player)로서 사용될 수도 있고, 각 부분이 독자적인 멀티미디어 스트리밍을 구현할 수도 있다. 여러 가지 다양한 형태의 규격을 가지는 멀티미디어 정보를 다루기 위하여 시스템들간의 결합이 자연스럽게 이루어지고 있다.

구현된 시스템은 MPEG-4 표준구현 그룹에서 제공되는 MPEG-4 시스템 표준에 기반한 MoMuSys 시스템이나 MP4 파일 포맷에 기반한 IM-1 2D 시스템과 독립적으로 두 부분을 모두 구현하였으며, 필요에 따라 선택된 도구로 구성할 수 있는 전체 시스템을 구현하였으며, 구현 과정에서 기존의 멀티미디어 포맷과 네트워크 표준과 같은 다양한 레이어 표준들과 호환성과 결합을 추구하였다는 데에 그 의미를 둘 수 있다.

VI. 결 론

본 논문에서 구현된 MPEG-4 표준 방식의 시스템과 파일포맷 코덱은 빠른 속도로 확산되고 있는 저가격이면

서 고성능인 멀티미디어 통신 서비스를 고려하여 유동적으로 기존의 방식 및 새로운 기능들을 모두 지원할 수 있는 부호화 도구를 제공한다. 또한 MPEG-4 표준 자체의 특성인 양방향성, 높은 압축율 및 다채로운 접속을 가능케 하는 AV(Audio-Visual) 표준화 방식 뿐만 아니라, 급격히 발전하는 관련 기술들을 이용할 수 있도록 고도의 융통성과 확장성을 제공할 수 있다. 또한 MPEG-4 표준 방식을 이용하여 기존의 단방향 통신의 일방적인 정보 전달의 차원을 벗어나, 수신자의 요구와 선택에 따라 필요한 정보를 원하는 시간에 제공할 수 있는 양방향 서비스가 실현될 수 있을 것이다.

최근에는 사용자 및 콘텐츠 제작자의 요구와 관심이 멀티미디어 데이터 서비스에 집중되고 있는데, 멀티미디어 데이터를 여러 종류의 전송 채널을 통하여 서비스하는 것은 여러 가지의 품질을 요구할 뿐만 아니라 시간적인 동기일치와 다중화의 측면에서 필수적인 기술이다. 본 논문에서 개발한 기술들은 MoMuSys나 IM-1과는 독립적인 구현의 예로서 멀티미디어 분야의 기술적인 독자성을 이루는데 의미를 가진다.

참 고 문 헌

- [1] ITU-T Recommendation H.263, Video Coding for Low Bit Rate Communications, 1998.
- [2] ITU-T Recommendation G.723, Dual Rate Speech Coder For Multimedia Communications Transmitting At 5.3 & 6.3 kbits/s, 1995.
- [3] ISO/IEC 14496: Information Technology Coding of audio-visual objects, 1999.
- [4] ISO/IEC JTC1/SC29/WG11, MPEG98/N2612p4, Intermedia Format(MP4) VM Text, 1998.
- [5] 호요성 외, "인터넷 정보가전을 위한 멀티미디어 서비스", *정보과학회지* 제19권 제4호, pp.26-38, 2001년 4월.

저 자 소 개

임 등 근



- 1994년 2월 : 전북대학교 전자공학과 (학사)
- 1993년 11월~1995년 2월 : 현대전자(주) 반도체 제2연구소 ASIC 분야(연구원)
- 1997년 2월 : 광주과학기술원 정보통신공학과 (석사)
- 1997년 3월~현재 : 광주과학기술원 정보통신공학과 (박사과정)
- 주관심분야 : 영상신호처리, 동영상 부호화, 컴퓨터비전과 HCI, 고속 VLSI 회로설계



이 정 우

1996년 2월 : 전북대학교 정보통신공학과 (학사)
 1998년 2월 : 광주과학기술원 정보통신공학과 (석사)
 1998년 3월~현재: 광주과학기술원 정보통신공학과 (박사과정)
 주관심분야 : 영상부호화, 멀티미디어 영상통신, 영상 비트율 제어, 객체기반 부호화(MPEG-4)



김 선 태

1997년 8월 : 한국과학기술원 전기 및 전자공학과(학사)
 2000년 2월 : 서울대학교 전기공학부(석사)
 2000년 2월~현재 : 한국전자통신연구원 연구원 컴퓨터소프트웨어 연구소
 주관심분야 : 멀티미디어 신호처리 및 압축, 멀티미디어 스트리밍, 데이터 방송



마 평 수

1985년 : 서울대학교 식물병리학과 졸업 (학사)
 1992년 : City University of New York, USA 전산학과 (석사)
 1995년 : Wright State University, USA 전산학과 (박사)
 1985년~1989년 : 시스템공학연구소 연구원
 1989년~1990년 : (주)태양금속 정보산업연구실 대리
 1996년~현재 : 한국전자통신연구원 컴퓨터소프트웨어기술연구소 책임연구원
 주관심분야 : 멀티미디어 저장서버, 스트리밍 기술, 멀티미디어 검색, 재생 기술 등



호 요 성

1981년 2월 : 서울대학교 전자공학과 (학사)
 1983년 2월 : 서울대학교 전자공학과 (석사)
 1983년 3월~1995년 9월: 한국전자통신연구소 선임연구원
 1989년 12월 : University of California, Santa Barbara, Department of Electrical and Computer Engineering (박사)
 1990년 1월~1993년 5월 : 미국 Philips 연구소 Senior Research Member
 1995년 9월~현재 : 광주과학기술원 정보통신공학과 부교수
 주관심분야 : 디지털 신호처리, 영상신호처리 및 압축, 초저속 영상통신, 디지털TV와 고선명 TV 방식, 삼차원 영상 모델링 및 부호화