

MPEG-4 IM1과 DirectShow의 통합

이광의
동의대학교 멀티미디어공학과

Embedding DirectShow into MPEG-4 IM1

KwangEui Lee
Dept. of Multimedia Engineering, DongEui Univ.
E-mail : kelee@dongeui.ac.kr

요 약

IM1은 MPEG-4 시스템의 레퍼런스 소프트웨어로서, 많은 MPEG-4시스템들이 IM1을 기준으로 구현되고 있으나, 아직 이에 맞는 코덱의 부족으로 MPEG-4의 목표에 부합하는 다양한 스트림의 재생에 문제가 있다. 다이렉트쇼우(DirectShow)는 마이크로소프트(Microsoft)사에서 개발된 멀티미디어 스트림을 재생하기 위한 프레임워크로서 다양한 형태의 코덱을 포함하고 있으며, 필요한 코덱을 개발하기 위한 프레임워크를 제공하여 앞으로도 많은 수의 코덱이 개발될 것으로 예측된다. 본 논문에서는 MPEG-4 콘텐츠가 다양한 스트림을 포함하고, 이를 IM1에서 재생할 수 있도록 IM1과 다이렉트쇼우를 통합하는 방법에 대해서 기술한다.

1. 서론

인터넷이 보편화 되고, 그 용량이 확장됨에 따라 인터넷을 통하여 실시간에 동화상을 전송, 재생하고자 하는 요구가 증대되었으며, 이에 따라 좀더 높은 비율의 멀티미디어압축에 대한 요구가 커졌다. 이러한 요구뿐만 아니라 상호작용 할 수 있는 멀티미디어에 대한 요구를 만족시키기 위하여 만들어진 표준이 MPEG-4이다[1].

MPEG-4에서는 고효율의 압축과 상호작용성을 제공하기 위하여 화면을 구성하는 데이터를 객체단위로 분리하여 각각을 압축한 후 이를 기본스트림(Elementary Stream)으로 전송한다. 클라이언트 시스템은 각각 기본스트림으로 전송된 다양한 스트림을 하나의 동화상으로 묶어 재생한다[2].

IM1은 MPEG-4 표준화 기구에서 MPEG-4 클라이언트 시스템의 레퍼런스 소프트웨어로서 제공되는 소프트웨어이다. 많은 MPEG-4시스템들이 IM1을 기준으로 구현되고 있으나, 아직 IM1에 적용 가능한 코덱의 부족으로 MPEG-4의 목표에 부합하는 다양한 스트림의 재생에 문제가 되고 있다.

다이렉트쇼우는 마이크로소프트사에서 개발된 멀티미디어 스트림을 재생하기 위한 프레임워크로서 다양한 형태의 코덱을 포함하고 있으며, 필요한 코덱을 개발하기 위한 프레임워크를 제공하여 앞으로도 많은 수의 코덱이 개발될 것으로 예측된다[3]. 따라서 다이렉트쇼우의 프레임워크를 IM1의 프레임워크에 통합하게 된다면, IM1을 이용하여 MPEG-4를 활용하면서 코덱 문제를 자연스럽게 해결할 수 있게 된다.

본 연구에서는 다이렉트쇼우에서 사용 가능한

다수의 코덱을 IM1에서 그대로 사용할 수 있도록 IM1과 다이렉트쇼우를 통합하는 방안을 제시하고, 그 방안에 따라서 다이렉트쇼우를 포함하는 IM1 코덱을 구현하였다. 특히 설명은 마이크로소프트사의 ASF를 기준으로 하였으나, 다이렉트쇼우가 지원하는 모든 코덱은 본 논문의 방법으로 이용 가능하다. 본 논문의 2절에서는 IM1과 다이렉트쇼우의 특징에 대하여 기술하고, 3절과 4절에서는 IM1과 다이렉트쇼우에 대한 통합설계 및 구현에 대하여 기술하며, 마지막으로 5절에서 결론을 맺는다.

2. IM1의 구조

IM1 플레이어는 프로그래머로 하여금 새로운 형식의 스트림을 추가할 수 있도록 하기 위하여 디코더 프레임워크를 제공한다[?]. 즉 새로운 타입의 스트림에 대하여 그 스트림을 디코딩하기 위한 디코더를 주어진 프레임워크를 이용하여 구현하고 이를 시스템에 등록하면 IM1 플레이어는 새로 등록된 타입의 스트림을 재생할 수 있게 된다. 여기서 전체 IM1의 프레임워크 중 설명에 필요한 디코드버퍼와 디코더, 그리고 컴포지션버퍼 부분만을 보면 다음의 그림 1과 같다.

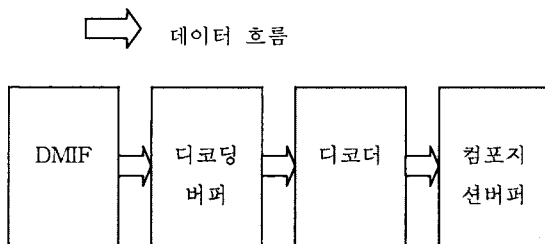


그림 1. IM1 프레임워크의 일부분

MPEG-4 표준의 여섯번째 부분으로 멀티미디어 데이터전송을 위한 프레임워크인 DMIF (Delivery Multimedia Integration Framework)는 서버로부터 전달된 데이터를 디코드 버퍼에 저장하고 디코더는 디코드 버퍼로부터 데이터를 읽어서 이를 디코딩 한 후 디코딩 된 데이터를 컴포지션 버퍼에 저장하여 컴포

지터가 이를 가져가 재생할 수 있도록 한다[4][5].

3. IM1과 DirectShow의 통합설계

IM1과 다이렉트쇼우를 통합하기 위해서는 IM1의 디코더 프레임워크를 이용하여 ASF 스트림을 위한 디코더를 구현하면 된다. 그러나 ASF 형식의 경우 마이크로소프트 자체 소유의 형식으로 현재 그 구조가 외부에 공개되지 않고 있기 때문에 개인이 ASF 디코딩을 위한 디코더를 구현하는 것은 불가능하다. ASF 형식의 데이터를 재생하기 위한 현재 알려진 유일한 방법은 다이렉트쇼우에서 제공하는 소스필터와 디코더 필터들을 조합하여 재생하는 방법이다.

상기의 문제를 해결하기 위하여 본 논문에서는 먼저 새로운 디코더를 IM1에 삽입하기 위하여 AXF(ASF 스트림을 포장한 형식)라는 스트림 포맷을 정의하고 AXF 형식을 위한 디코더를 구현하였다. AXF 파일은 실제로 구현된 디코더를 활성화 시키는 역할만을 하기 위한 파일형식으로 재생하고자 하는 ASF 스트림의 위치만을 저장하고 있다.

전체적인 설계는 다음의 그림 2와 같은 구조를 가진다.

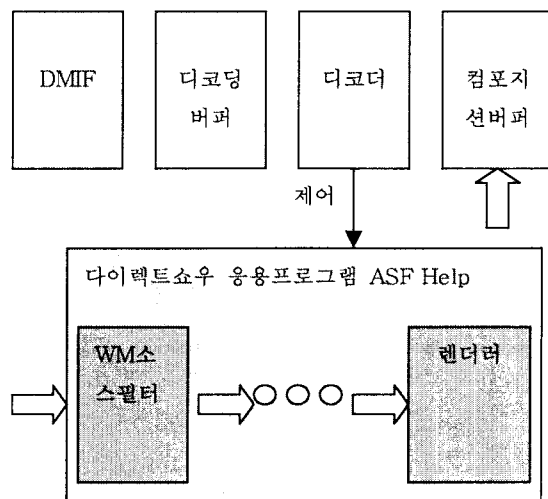


그림 2 설계의 전체구조

이러한 구조에서 원하는 결과를 얻기 위하여 프로그래밍의 동작과정은 다음의 단계들로 구성된다.

1. 콘텐츠 제작자는 재생하고자 하는 ASF 스트림의 위치만을 포함하는 확장명이 "AXF"인 파일을 만들고 제공되는 MUX를 이용하여 MPEG-4 시스템 파일을 생성한다.
2. 클라이언트가 AXF 파일이 포함된 MPEG-4 시스템 파일을 요청하면, IM1은 AXF 파일을 디코딩하기 위한 디코더를 실행시킨다.
3. AXF 디코더는 외부에 다이렉트쇼우 응용프로그램을 생성하여(이를 ASF Helper라 하자) 그 응용프로그램에 AXF 파일로부터 전달 받은 ASF 파일의 URL을 넘겨준다.
4. ASF Helper는 정상적으로 ASF 파일을 디코딩하고 이를 렌더링하게 되는데, 이때 렌더링 필터는 새로 제작된 것으로 화면에 렌더링하지 않고 IM1의 AXF 디코더를 위한 컴포지션 버퍼에 렌더링한다.
5. IM1은 컴포지션 버퍼의 데이터를 이용하여 화면을 재생한다.

다시말하면, AXF 디코더의 기능은 AXF 파일로부터 재생하고자 하는 ASF 파일의 위치를 받아서 외부에 다이렉트쇼우 응용프로그램을 이용하여 주어진 ASF 파일을 디코딩 한 후 그 결과를 IM1 내부에 할당된 AXF 디코더의 출력으로 내 보내는 방법으로 IM1이 ASF 형식의 데이터를 재생할 수 있게 하는 기능을 수행한다. 즉, 새로운 스트림 타입인 AXF의 디코더는 자신의 컴포지션 버퍼를 다이렉트 쇼우 응용프로그램인 ASF Helper로부터 데이터를 받아서 채운다.

4. 구현

전체적인 구현은 AXF 파일형식 처리부분과 디코더 구현부분으로 분리할 수 있다.

4.1 AXF 파일형식 처리부분

AXF 파일형식 처리부분은 주로 새로운 파일 형

식인, AXF 형식을 정의하고, 정의된 AXF를 위한 디코더를 구현하는 일이다. 또한 상기의 AXF 디코더가 존재하는 경우 IM1으로 하여금 AXF 디코더를 활성화시키고 동작시키도록 하기 위하여 AXF를 등록하고 MPEG-4 데이터에 AXF 파일을 포함시키는 작업을 포함한다.

이를 위해 먼저 AXF라는 파일형식을 정의하여야 한다. 이때 IM1에서 파일에 대한 메타데이터로 필요로 하는 모든 정보는 ASF Helper로부터 얻을 수 있으므로 AXF가 실제로 재생될 ASF에 대하여 제공할 내용을 ASF의 위치뿐이다. 따라서 파일은 다음의 예에서와 같이 ASF 파일의 URL만을 포함하면 된다.

AXF 파일의 전형적인 예)

`http://203.245.205.131/stream/sample.asf`

^Z

이제 AXF 파일을 포함하는 MPEG-4 시스템 스트림을 생성하기 위하여 시스템 스트림을 생성해주는 도구인 MUX에 AXF 파일형식을 인식할 수 있도록 하는 코드를 추가 하여야 한다[6]. 참고문헌 [6]에 따라서 Mux에 AXF를 인식할 수 있는 코드를 추가하는 것은 매우 간단하다. 그 코드가 하는 일은 MUX의 요청에 따라 파일의 내용을 8bit 씩 돌려주며, DTS (Decoding Time Stamp)를 돌려주는 역할을 한다.

이때 첫번째 그리고 유일한 AU (Access Unit)의 data는 AXF 파일의 첫 줄이며 DTS (Decoding Time Stamp)를 0으로 한다. 다음 번 AU 요청에 -1 (EOF을 의미한다)을 돌려준다.

4.2 디코더 구현부분

디코더 구현부분은 다이렉트쇼우에서 구현되어야 하는 부분과, IM1에서 구현되어야 하는 부분으로 구분할 수 있다. 먼저 다이렉트쇼우에서 구현되어야 하는 부분은 렌더링 필터의 구현이다. 앞의 절차에서 살펴본 바와 같이 필요한 렌더링 필터는 그 결과를 화면에 렌더링하는 것이 아니라, IM1의 컴포지션 버퍼에 렌더링 할 필요가 있다. 따라서 이러한 기능을

수행하는 다이렉트쇼우 필터를 작성하여야 한다. 이 때 다음과 같은 점들을 고려하여야 한다.

1. IM1 플레이어는 RGB(24bit)만을 인식하므로 제작된 필터가 입력을 RGB(24bit)로 변경하여 주어야 한다.
2. 구성된 렌더링 필터는 디코딩된 데이터를 전달 받기 위하여 메모리 주소를 전달 하고, 현재 필터가 실행중임을 확인할 수 있는 인터페이스를 가져야 한다.
3. ASF를 디코딩하기 위하여 필요한 메타 데이터를 IM1에 제공하여야 한다.

또한 다이렉트쇼우는 ASF 파일에 대하여 기본적으로 사용하는 필터에 대한 정보를 가지고 있으므로 위에서 제작된 필터를 사용하기 위하여 다이렉트쇼우의 필터그래프를 조작하는 응용프로그램을 작성하여야 한다. 이것이 ASF Helper 프로그램이다. 실제로 IM1의 AXF 디코더는 이 응용프로그램을 호출하게 되고, 이 응용프로그램이 구현된 다이렉트쇼우 필터를 호출하여 사용하게 된다.

마지막으로 IM1의 디코더 프레임워크에 따라서 AXF 디코더를 구현 추가하여야 하는데, IM1의 AXF 디코더는 앞에서 언급한 바와 같이 실제로 ASF 파일에 대한 디코딩을 하지 수행하는 않고, 자신이 실행되었을 때, AXF 파일로부터 ASF 파일의 URL을 얻어 이를 앞에서 제작된 다이렉트쇼우 응용프로그램인 ASF Helper에게 전달하는 기능만을 수행한다. 이는 IM1의 디코더 프레임워크를 이용하여 간단히 구현할 수 있다.

5. 결론

본 논문에서는 다이렉트쇼우에서 사용 가능한 다수의 코덱을 IM1에서 그대로 사용할 수 있도록 IM1과 다이렉트쇼우를 통합하는 방안을 제시하고, 그 방안에 따라서 다이렉트쇼우를 포함하는 IM1 코덱을 구현하였다. 모든 코드는 IBM 호환 컴퓨터에서 윈도우 2000 운영체제를 기반으로 구현되고 시험되었다.

[참고문헌]

- [1] MPEG Home Page,
<http://mpeg.telecomitalia.com/>
- [2] ISO/IEC 14496-4, "Information Technology Generic Coding of Audio-visual Objects Part 1: System," May., 1998
- [3] DirectShow Home Page,
<http://www.microsoft.com/directx>
- [4] ISO/IEC 14496-4, "Information Technology Generic Coding of Audio-visual Objects Part 6: DMIF," May., 1998
- [5] ISO/IEC JTC1/SC29/WG11 M3111, "APIs for Systems Software Implementation," May., 1998.
- [6] ISO/IEC JTC1/SC29/WG11 M3548, "MPEG-4 Off-line Multiplex Software Release 1.3," May., 1998.