

DVB-MHP 셋톱박스 미들웨어를 위한 MPEG-2 비디오 드립 디코더의 설계 및 구현

김우종, 이양선
서경대학교 컴퓨터공학과
e-mail : {wjkim^o, yslee} @pl.skuniv.ac.kr

Design and Implementation of MPEG-2 Video Drips Decoder for DVB-MHP Set-top Box Middleware

Woo-Jong Kim, Yang-Sun Lee
*Dept. of Computer Engineering, Seokyeong University

요 약

데이터 방송은 지상파, 위성, 케이블 등의 방송망을 이용하여 하나의 송신자가 다수의 수용자에게 디지털 멀티미디어 콘텐츠를 여러 종류의 단말기에 전송하는 것으로 콘텐츠는 프로그램에 연동되는 서비스 및 비연동형 서비스, 또는 대화형 서비스 형태로 제공되는 다양한 서비스를 포함한다. 방송망을 이용한 데이터방송의 전송 규약은 DSM-CC에서 정의한 캐로셀(carousel) 형태로 전송된다. 캐로셀은 주기적으로 반복 전송되는 데이터 모듈을 가리키며, 캐로셀로 전송되는 데이터 중에는 MPEG-2 비디오로 인코딩되어 광고와 배경 이미지 등으로 사용할 수 있는 영상 데이터가 있다. 이 MPEG-2 비디오 영상 데이터를 TV화면에 출력하기 위해 셋톱박스의 미들웨어는 썬 마이크로시스템즈(Sun Microsystems)에서 만든 자바 기반의 JMF(Java Media Framework) 플레이어를 통해 화면에 출력한다. 이렇게 제공되는 MPEG-2 비디오 영상 데이터를 비디오 드립(drips)이라 하고, 비디오 드립을 JMF 플레이어를 통해 화면에 출력하는 모드를 드립피드(drip-feed) 모드라 한다. 그러나 MHP용 셋톱박스를 위한 JMF 버전 1.0은 그대로 사용할 수 없다. 비디오 드립 모드를 위한 구현이 없기 때문에 별도의 확장을 통해 비디오 드립을 지원하거나, 네이티브(Native) 메소드를 만들어 JMF를 확장하는 방식을 제공해야 한다.

본 논문에서는 데이터 캐로셀로 전송되는 비디오 드립 모드를 구현하기 위해 네이티브 코드로 JMF를 확장해서 비디오 드립을 JMF 플레이어를 통해 TV 화면에 출력하는 MPEG-2 비디오 드립 디코더를 설계하고 구현하였다.

1. 서론

데이터 방송은 디지털 TV방송에서 기본 TV방송 프로그램과는 별도로 문자, 그림, 음향 등으로 구성된 다양한 데이터를 전송신호에 다중화하여 함께 방송함으로써, 시청자가 기본 A/V방송 이외에 부가적인 콘텐츠를 이용할 수 있게 하며, 이에 대한 시청자의 응답을 실시간 온 라인으로 접수 받아 추가 정보를 제공하거나 프로그램에 반영하는 방식으로 상호작용 함으로써 양방향(Interactive) 디지털 TV(iDTV)를 구현하는 서비스이다[6,11].

데이터 방송의 서비스는 기본 A/V(Audio/Video)

프로그램 내용과의 관련 여부에 따라 '프로그램 연동형'과 '독립형' 서비스로 구분하며, 프로그램 연동형 서비스는 기본 TV 프로그램에 관련된 콘텐츠이고, 독립형 서비스는 기본 TV 프로그램과 관계없이 독립적인 내용으로 구성된 콘텐츠이다. 이런 데이터 방송의 기술적인 표준으로 위성파와 지상파, 케이블에 따라 그 표준이 다르게 적용된다[5,6,11].

우리나라 디지털 위성방송의 데이터 방송 기술표준은 유럽의 디지털 방송 표준 기구인 DVB(Digital Video Broadcasting)에서 정한 MHP(Multimedia Home Platform)방식이며, 지상파 디지털 TV의 데

이더 방송 기술표준은 미국의 ATSC(Advanced Television Standard Committee)에서 정한 DASE(DTV Application Software Environment), 디지털 케이블 TV의 기술표준은 미국방식인 OpenCable-OCAP(OpenCable Application Platform)이다. 국내 위성방송 서비스 스카이라이프에서는 유럽 방식인 MHP 방식으로 데이터 방송을 하고 있다. 그리고 데이터 방송을 위한 데이터의 전송 기반은 DSM-CC(Digital Storage Media Command and Control)의 데이터 다운로드 프로토콜을 사용하고 있다. DSM-CC는 섹션(Section)이라는 단위의 가변길이 데이터 구조로 전송하고 있다. 섹션을 이용한 데이터 전송 규약 중에는 캐로셀(Carousel)이 있다. 이 캐로셀은 반복적으로 전송되는 데이터 모듈들을 가리키는 말로 양방향 디지털 TV(iDTV)에서는 이러한 모듈을 계층구조로 만들어 읽기 전용 방송 파일 시스템을 구축해서 사용한다[1,4,5].

응용 프로그램 작성 시 이미지, 클래스 파일 등의 대부분의 파일이 데이터 캐로셀(Data Carousel) 혹은 객체 캐로셀(Object Carousel) 등의 캐로셀 규약을 통해 전송되며, Java.io 패키지 API를 통해 읽을 수 있게 되어 있다. MHP에서는 배경화면이나 광고 화면으로 사용할 수 있는 콘텐츠 형식으로 MPEG-2 비디오로 인코딩된 간단한 영상 이미지를 출력 할 수 있는 모드를 제공하고 있는데 이를 비디오 드립-피드(drip-feed) 모드라 한다[4,5].

비디오 드립은 썬 마이크로시스템즈(Sun Microsystems)의 JMF(Java Media Framework) 플레이어를 통해 TV 화면에 나타나게 된다. 그러나 MHP용 셋톱박스를 위한 JMF 버전 1.0은 그대로 사용할 수 없다. 비디오 드립 모드를 위한 구현이 없기 때문에 별도의 확장을 통해 비디오 드립을 지원하거나, 네이티브(Native) 메소드를 만들어 JMF를 확장하는 방식을 제공해야 한다[5,8].

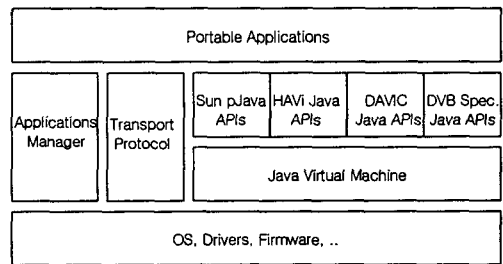
본 연구에서는 비디오 드립 모드를 구현하기 위해 네이티브 코드로 JMF를 확장해서 MPEG-2 비디오 스트림 디코더를 PC에서 설계하고 구현하는 방법과 셋톱박스 응용 프로그램에서 비디오 드립을 서버에 요청하고, JMF와 디코더를 어떻게 연결해서 TV 화면에 출력하는지 PC상에서 구현하고 테스트하였다.

2. DVB-MHP 셋톱박스

2.1 DVB-MHP

우리나라 디지털 위성방송 표준으로 채택된 MHP 1.0.1은 썬의 퍼스널 자바 1.2를 기반으로 자바 TV 1.0과 HAVi(Home Audio /Video Interface) 레벨 2 유저 인터페이스 1.01을 포함하고 있다[4].

[그림1]은 MHP 표준에 따른 셋톱박스 미들웨어의 구조를 나타낸다. 자바 API는 AWT 중에서 버튼, 프레임, 패널 등과 같은 중량 컴포넌트(Heavy Weight Component)는 제외되고, 자바 TV에서 방송파일 시스템과 연관된 캐로셀 패키지, TV화면과 관련된 미디어 패키지, 서비스 정보, 셋톱박스 응용 프로그램을 위한 xlet 패키지 등을 포함하고 있다. 그리고 HAVi UI API는 중량 컴포넌트를 대신 할 각종 사용자 인터페이스 패키지를 제공한다[4,5,6].



[그림 1] DVB-MHP 구조

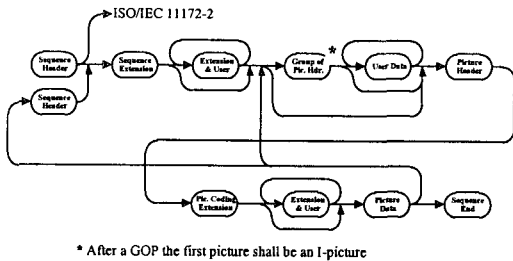
2.2 비디오 드립

셋톱박스는 메모리와 기타 하드웨어 자원이 일반 PC보다 성능이 떨어지기 때문에 메모리를 효율적으로 사용하면서 기존 A/V 화면이나 데이터 방송 화면에 광고 이미지나 배경 이미지 등으로 사용할 수 있는 비디오 드립(Video Drip)이라는 콘텐츠 형식을 MHP에서 정의 하고있다[4,5].

비디오 드립 전송 모드는 MPEG-2 비디오 스트림을 응용 프로그램에서 청크(Chunk) 단위의 바이트 배열로 잘라, MPEG-2 비디오 디코더로 공급하는 것이다. 이 모드에서 MPEG-2 비트 스트림은 B-프레임이 없는 I-프레임과 P-프레임만을 가지고 있고, 필드 구조가 아닌 프레임 구조로 이루어진 비트 스트림이다. MPEG-2 비디오 디코더로 공급되는 청크는 하나의 비디오 프레임에 해당하는 단위이며 최초로 공급되는 청크는 시퀀스 헤더와 GOP(Group of Picture) 헤더, 그리고 I-프레임이 공급되며 그 뒤로 P-프레임이 반복해서 공급된다. 바이트 형태의 청크로 구성된 콘텐츠는 [그림2]와 같은 구문에 따라 공급된다. 이렇게 공급되는 모든 청크를 하나로 연결하면 ISO/IEC 13818-2에서 기술하고 있는 구문을 만족하는 하나의

MPEG-2 비디오 스트림이 된다[2,7,12].

비디오 드립 모드에서 MPEG-2 비디오 스트림은 B-프레임이 없는 'low delay' 모드여야 하고, JMF 플레이어에서 MPEG-2 디코더로 비디오 스트림을 전달하기 위해 사용하는 데이터 소스는 MHP에서 새로 정의한 org.dvb.media.DripFeedDataSource 인스턴스를 사용해서 연결해야 한다. DripFeedDataSource 클래스는 feed(byte[])라는 메소드를 새로 정의하고 있다. 이 메소드에서 디코더로 비디오 스트림의 청크를 바이트 단위로 전달한다. 이렇게 전달되는 비디오 드립은 셋톱박스의 미들웨어에서 JMF 플레이어와 연결된 MPEG-2 비디오 디코더에 의해 디코딩되어 디지털 TV 화면으로 출력된다[4,8,9].

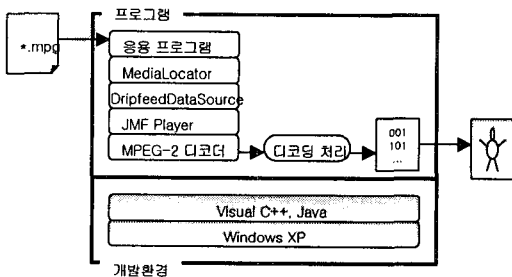


[그림 2] MPEG-2 비디오 데이터의 구조

3. MPEG-2 비디오 드립 디코더

3.1 시스템 구조

[그림3]은 비디오 드립 디코더의 전체 시스템 구조이다.



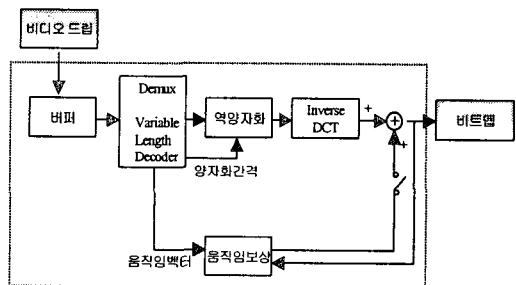
[그림 3] 비디오 드립 디코더 시스템 구조

비디오 드립 디코더의 시스템은 윈도우즈 XP에서 비주얼 C++와 자바를 이용해 구현하였다. 응용 프로그램과 미디어 로케이터 그리고 DripFeedDataSource, JMF 플레이어는 자바 언어로 구현 했으며, MPEG-2 디코더는 비주얼 C++로 구현하였다. JMF 플레이어와 MPEG-2 디코더는 네이티브 코드로 JMF 플레이어에서 MPEG-2 디코더 DLL(Dynamic

Linking Library)을 호출하는 구조로 이루어져 있다.

3.2 MPEG-2 디코더

MHP와 자바 TV에서는 미디어 제어를 위한 표준 API로 JMF 1.0을 사용하고 있다[6,8]. JMF는 비디오 스트림을 프레임 단위로 입력받아 출력하는 기능이 없다. 그래서 비디오 스트림을 프레임 단위로 처리하는 기능을 구현 해야한다. 본 논문에서는 MPEG-2 디코더를 구현하기 위한 방법으로 mpeg.org의 소스코드를 기반으로 윈도우즈용으로 수정해서 자바 코드에서 호출할 수 있도록 네이티브 인터페이스를 갖는 DLL로 작성하였다[7,8]. 그리고 화면 출력은 JMF 플레이어에서 기존 디코더의 네이티브 코드를 호출하는 부분을 MPEG-2 비디오 디립 디코더의 네이티브 코드를 호출하도록 수정하였고, 디코딩된 결과를 비트맵 이미지로 받아 화면에 출력하도록 구현하였다. [그림 4]는 비디오 드립 디코더의 디코딩 과정을 나타낸다.



[그림 4] 비디오 드립 디코딩 과정

3.3 구현 및 결과

기본적으로 JMF에서 미디어 데이터의 출력과정은 데이터 소스와 플레이어가 연결되면, 미디어 데이터는 자동으로 메모리로 적재되어 플레이를 할 수 있는 상태가 된다. 그러나 드립 모드에서는 미디어 데이터가 자동으로 메모리로 적재되지 않고 수동적으로 메모리에 적재되어 화면에 출력되는 특별한 콘텐츠이다. 본 논문에서는 DVB-MHP의 클래스와 JMF 클래스의 일부를 PC 환경에 맞게 수정하고 MPEG-2 비디오 디코더를 구현해서 시뮬레이션하였다.

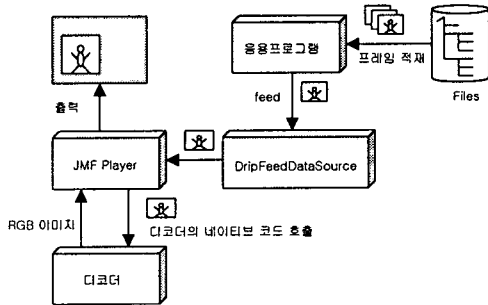
MPEG-2 비디오 드립 파일은 DSM-CC의 캐로셀로 전송되어 파일 시스템 형식으로 저장되어 있다고 가정하고, 자바 응용 프로그램에서 미디어 파일을 하나씩 읽어 들인 다음 디코더로 한 프레임씩 공급하게 된다. MPEG-2 비디오 디코더는 DLL로 작성했으며, JMF Player는 비디오를 디코딩하기 위해 디코더의 네이티브 코드를 호출한다. [예제1]은 응용 프로그램이 MPEG-2 비디오 스트림을 파일로부터 읽어내어 JMF Player와 DripFeedDataSource를 사용해서 비디오

오 프레임을 출력하는 간단한 코드이고, [그림5]는 비디오 드립의 출력 과정을 나타낸다.

```

DripFeedDataSource dripSource;
Player dripPlayer;
MediaLocator dripLocator
InputStream in;
public void create() {
    dripLocator = new MediaLocator("dripfeed://");
    dripSource = Manager.createDataSource(dripLocator);
    dripPlayer = Manager.createPlayer(dripSource);
}
public void feed() {
    in = new FileInputStream(new File(nFrame + ".mpg"));
    int size = in.available();
    byte[] frame = new byte[size];
    in.read(frame);
    in.close();
    dripSource.feed(frame);
}
    
```

[예제 1] 비디오 드립 예제코드



[그림 5] 비디오 드립 출력 과정

MPEG-2 비디오 드립을 출력하기 위한 전체 흐름은 먼저 콘텐츠가 무엇인지를 지정하는 미디어 로케이터를 생성해야한다. 이 미디어 로케이터를 인자로 데이터 소스를 생성하고, 생성된 데이터 소스 객체를 가지고 JMF 플레이어와 연결시킨다. 응용 프로그램은 자바 API로 미디어 파일을 읽고, 데이터 소스 객체의 feed() 메소드를 이용해 플레이어로 영상 데이터의 바이트 배열을 전달한다. 플레이어는 디코더의 네이티브 코드를 호출해서 영상 바이트 배열을 디코더로 넘겨주면 디코더는 영상 데이터를 디코딩해서 RGB 비트맵 형태로 변환해서 플레이어로 넘겨준다. 플레이어는 RGB 비트맵을 윈도우 화면으로 출력한다. [그림6]은 JMF로 비디오 드립을 출력한 결과 화면이고, 키보드를 누르면 다음 이미지를 출력한다.

4. 결론

MHP에서는 비디오 드립 이외에 GIF89a, JPEG, PNG, 자막, 문자열, 음성, 폰트와 같은 다양한 콘텐츠 형식을 정의하고 있다. 이런 콘텐츠 중에서 MPEG-2



[그림 6] 비디오 드립 출력

비디오 드립은 제한된 셋톱박스의 자원을 효율적으로 사용하는 것으로 연속되는 영상 이미지들을 TV 화면에 출력하기 위해 MPEG-2 비디오로 인코딩된 스트림을 JMF로 한 프레임씩 재생하는 콘텐츠이다.

JMF에서는 비디오 드립을 위한 프레임 단위의 디코딩 처리를 지원하지 않는다. MHP 셋톱박스 미들웨어에서 비디오 드립을 구현하려면 JMF를 수정하거나 별도의 디코더를 제공해야 한다. 본 논문에서는 이를 위해 PC상에서 비디오 드립 디코더를 설계하고 구현하였다. 앞으로 MPEG-2 비디오 드립 이외의 콘텐츠 형식을 처리하도록 확장해서 MPEG-2 전송 스트림을 분석하는 분석기에 포함시킬 예정이다.

참고문헌

- [1] ISO/IEC 13818-6, "Information Technology - Generic Coding of Moving Pictures and Associated Audio Information: Part 6: Extension for Storage Media Command and Control", International Standard, Sep 1998.
- [2] ISO/IEC 13818-2, "Information Technology - Generic Coding of Moving Pictures and Associated Audio Information: Video", International Standard, May 1996.
- [3] ISO/IEC 13818-1, "Information Technology - Generic Coding of Moving Pictures and Associated Audio Information: Systems", International Standard, Apr 1996.
- [4] ETSI TS 102 812 V1.1.1 Chap 7.
- [5] <http://www.mhp-interactive.org/tutorial/>
- [6] <http://www.mhp.org>
- [7] <http://www.mpeg.org>
- [8] <http://java.sun.com/products/java-media/jmf/>
- [9] <http://java.sun.com/products/javatv/>
- [10] Ken Arnold, James Gosling & David Holmes, The Java™ Programming Language, 3rd ed., Addison Wesley, 2000.
- [11] <http://tri.kbs.co.kr/publication/>
- [12] Tektronix Document, "A Guide to MPEG Fundamentals and Protocol Analysis", Tektronix, 1997.