

고화질 WMV 파일 포맷 분석을 통한 Key-Frame추출 알고리즘 설계 및 구현

문동규^o, 이좌형, 김병길, 김윤, 정인범
강원대학교 전기전자정보통신공학부 컴퓨터전공

{dkmoon^o, jhlee, bgkim}@snslab.kangwon.ac.kr, {yooni, ibjung}@kangwon.ac.kr

Design and Implementation of the Key-Frame Extraction Algorithm

through Analysis of the High-Definition WMV File Format

Dong-Kyu Moon^o, Joa-Hyoung Lee, Byoung-Gil Kim, Yoon Kim, In-Bum Jung
Dept. Computer Engineering

요 약

WMV(Window Media Video)는 ASF(Advanced Streaming Format)파일 포맷을 기반으로 하는 마이크로소프트사의 대표적인 스트리밍 파일 포맷이다. WMV 파일 포맷은 MPEG-4를 기반으로 내용 기반 부호화를 통해 영상을 객체 단위로 나누어 처리하므로 사용자의 의도에 따라 다양한 조작과 디스플레이가 가능하다. 또한 영상 데이터를 패킷에 붙여 네트워크를 통해 스트리밍 서비스도 가능하다. 따라서 본 논문에서는 WMV파일 포맷 분석을 통해 영상의 가장 중요한 Key-Frame을 추출과 데이터 헤더 부분을 제외한 실제 영상 스트림의 시작 위치를 찾는 알고리즘을 설계 및 구현하였다. 이로써 실제 스트리밍 서비스외에도 다양한 멀티미디어 서비스에 적용 가능한 방법을 마련하였다.

1. 서 론

초고속 인터넷 서비스의 보급이 확산되면서 스트리밍 콘텐츠는 엄청난 변화를 거듭해왔다. 브로드밴드 환경 속에서 대역폭의 제약을 만족시키기 위해 저용량 저화질의 미디어 스트리밍 서비스를 하던 시대를 지나 기가바이트 대역폭의 네트워크 환경이 도래하면서 DVD급 이상의 고화질 미디어 콘텐츠가 새로운 사용자의 요구로 부상하고 있다. 따라서 이러한 사용자의 요구를 만족시키기 위해 새로운 환경이 조성되고 활발한 연구가 진행되고 있다.

WMV파일 포맷은 MPEG-4를 기반으로 내용 기반 부호화를 통해 영상을 객체 단위로 나누어 처리하기 때문에 사용자의 의도에 따라 다양한 조작과 디스플레이가 가능하다. 또한 그림 1과 같이 멀티캐스트 IP, UDP, RTP, TCP, HTTP와 효율이 최상이면서 이용 가능한 프로토콜을 자유롭게 선택하여 패킷단위로 네트워크를 통해 스트리밍 서비스가 가능하다. 즉 WMV는 유무선 네트워크를 통해 멀티미디어 스트리밍 서비스에 적합한 파일 포맷이며 무선 인터넷의 보급이 확산되는 지금 노트북, PDA, 비디오 전화, 원격 영상회의, 인터넷, 양방향 방송, DVD, 캠코더, 디지털 카메라 등 광범위한 부분에 적용 가능하다. 따라서 본 연구는 마이크로소프트사가 제공하는 고화질의 WMV파일 포맷 분석을 통해 필수적인 Key-Frame추출과 데이터의 헤더부분을 제외한 실제 영상 스트림의 시작 위치를 찾는 알고리즘을 설계 구현함으로써 다양한 고화질 미디어 스트리밍 서비스에 적합한 기반 기술 연구를 목적으로 한다.

논문의 구성은 다음과 같다. 2장에서는 논문에 대한 이해를 돕기 위하여 파일 포맷의 구성과 WMV 패킷 구조 등 전반적인 WMV 파일 포맷의 설명을 다룬다. 3장에서는 Key-Frame 추출과 데이터 시작위치를 찾기 위한 알고리즘을 소개하고 코드에 대한 설명을 다룬다. 마지막으로 4장에서는 결론 및 향후 연구 방향과 과제에 대해 기술한다.

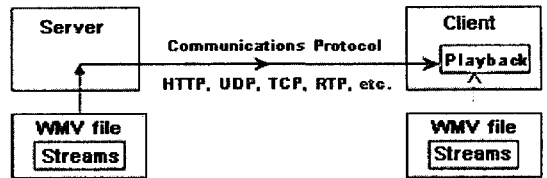


그림 1. WMV 파일 포맷

2. 관련 연구

2.1 WMV 파일 포맷

WMV(Window Media Video)는 ASF(Advanced Streaming Format) 파일 포맷을 기반으로 하는 마이크로소프트사의 대표적인 스트리밍 파일 포맷이다. WMV파일 포맷의 형식과 구조는 ASF와 동일하다. 현재 마이크로소프트사에서는 WMV를 파일 포맷으로 하는 고화질 콘텐츠인 WMVHD를 DVD로 판매하고 있다. 본 연구에서는 WMV파일 포맷 분석을 위해 마이크로소프트사의 WMVHD를 이용하였다[1][2].

2.2 WMV 파일 포맷 구성

WMV 파일은 그림 2와 같이 헤더 오브젝트, 데이터 오브젝트, 인덱스 오브젝트 등 3개의 top-level 파일 오브젝트로 구성된다. 헤더 오브젝트에는 WMV 파일의 멀티미디어 스트림에 대한 전반적인 정보가 포함되고, 프로토콜에 별도로 전송된다. 데이터 오브젝트는 스트림이 가능한 실제 미디어 데이터가 포함되어 있다. 마지막으로 인덱스 오브젝트에는 데이터 오브젝트가 포함하고 있는 데이터 유닛들의 인덱스 엔트리가 포함되어 미디어 데이터의 검색이나 유지보수에 사용된다.

* 본 연구는 한국과학재단 목적기초연구(R05-2003-000-12146-0) 지원으로 수행되었음.
* 본 연구는 강원대학교 ITRC 지원을 받아 수행하였음.

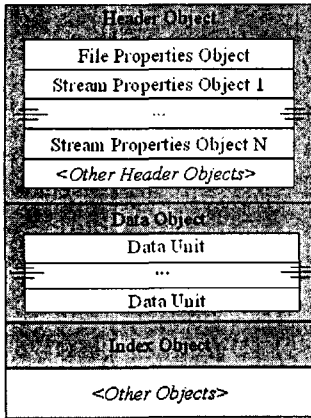


그림 2. top-level 파일 오브젝트

여기서 헤더 오브젝트와 데이터 오브젝트는 WMV파일 포맷에 반드시 필요한 부분이며 나머지 인덱스 오브젝트는 선택적이지만 많이 사용되기 때문에 포함하는 것이 좋다.

2.3 WMV 오브젝트

WMV 파일을 구성하는 기본 유닛은 WMV 오브젝트이다. WMV 오브젝트는 여러 오브젝트들을 식별할 수 있도록 128bit의 GUID로 구성된 오브젝트 아이디와 오브젝트 사이즈 정보를 가지는 64bit Integer, 그리고 실제 데이터를 포함하고 있는 가변길이의 오브젝트 데이터가 있다. WMV 오브젝트는 AVI 파일포맷의 chunk와 비슷하다. 그림 3은 128bit의 Object ID, 64bit의 Object size, 나머지 가변길이의 Object Data의 구조를 보여주고 있다.

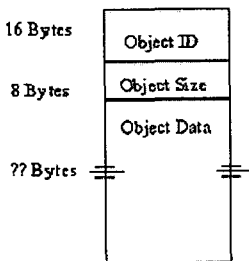


그림 3. WMV 오브젝트

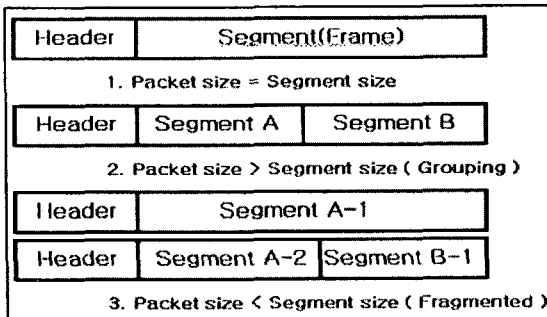


그림 4. WMV 패킷 구조

2.4 WMV 패킷

WMV 파일 포맷의 중요한 구성단위는 패킷이다. WMV 파일 포맷은 스트리밍을 목적으로 한 포맷이기 때문에 데이터 전송을 위한 패킷 구조가 존재한다. 패킷은 세그먼트라고 불리는 비디오/오디오 프레임으로 구성되어 있고 고정된 사이즈를 갖는다. 그림 4는 고정된 패킷에 데이터를 담아 전송하기 위한 3가지 가능성을 나타낸다.

1. 세그먼트사이즈가 패킷 사이즈와 같다면 하나의 패킷에 하나의 세그먼트가 포함되어 전송된다.
2. 세그먼트사이즈가 패킷 사이즈보다 작다면 하나의 패킷 사이즈에 맞도록 여러개의 세그먼트가 포함되어 전송된다. 따라서 세그먼트들은 'grouping' 된다.
3. 세그먼트 사이즈가 패킷 사이즈보다 크다면 하나의 세그먼트가 다른 패킷들로 나누어 진다. 따라서 세그먼트들은 'fragmented' 된다.

2.5 Key-Frame

MPEG-4는 I 프레임, B 프레임, P 프레임이 혼합된 형태의 구조를 지니고 있다. 이때 B 프레임과 P 프레임은 다른 프레임을 참조해야 복호화가 가능한 반면 I 프레임은 자신만으로 복호화가 가능한 프레임이다. 또한 B,P 프레임의 중요한 참조 프레임이다. 따라서 I 프레임은 MPEG-4 파일에서 필수적으로 필요한 프레임이다. 이와 마찬가지로 WMV의 Key-Frame은 MPEG-4의 I 프레임과 비슷하다. 즉 Key-Frame은 WMV파일에서 필수적으로 필요한 프레임이다. Key-Frame을 추출하여 적용 가능한 기능은 주로 고속 재생 기능에서 사용될 수 있다. 고속 재생시 대역폭의 문제가 되는데 Key-Frame만을 추출하여 전송할 경우 일정한 대역폭을 유지시킬 수 있다. 이밖에도 다양한 멀티미디어 서비스에 적용가능하다.

3. Key-Frame 추출 알고리즘

```

while(1)
{
    // 파일에서 한 바이트씩 읽어 배열에 삽입
    ch=fgetc(in);

    if(ch == -1) break;

    for(i=0; i<15; i++)
    {
        temp[i]=temp[i+1];
    }

    temp[i]=ch;

    // 데이터 오브젝트 시작위치 검색
    // 데이터 오브젝트 ID : 0x75b22636
    if(*(unsigned int *)temp == 0x75b22636)
    {
        pos=ftell(in)+34; // pos:시작위치
        printf(" start position : %u \n",pos);
        break;
    }
}
    
```

그림 5. 데이터 오브젝트의 시작위치를 찾는 루틴

3.1 데이터 오브젝트 시작위치 추출

그림 5는 데이터를 16바이트 단위로 읽어와 데이터 오브젝트의 시작 위치를 찾는 루틴이다. WMV 파일 포맷의 경우 위에서 언급한대로 헤더 오브젝트, 데이터 오브젝트, 인덱스 오브젝트로 나누어지는데 실제 영상데이터가 존재하는 부분은 데이터 오브젝트이기 때문에 헤더 오브젝트 부분은 제외되어야 한다. 따라서 영상데이터를 16바이트의 GUID로 읽어와 데이터 오브젝트의 ID와 비교를 통해 데이터 오브젝트의 시작위치를 알아낸다. WMV에서 데이터 오브젝트 ID의 첫 4바이트는 0x75b22636이다.

3.2 패킷 분석

WMV 파일 포맷의 중요한 파일 구조가 패킷이다. WMV는 패킷안에 비디오와 오디오를 압축한 실제 데이터를 포함하여 스트리밍에 적합하도록 구성되어 있기 때문에 패킷에 대한 분석은 WMV 파일 포맷을 이해하는데 중요한 부분이다. 패킷 분석을 위해 패킷단위로 데이터 파일을 읽어온다.

그림 6의 switch문은 패킷 단위로 읽어와 각각 패킷 사이즈, 시퀀스, 패딩사이즈를 분석하는 부분이다. 패킷사이즈는 데이터 파일로부터 읽어오는 패킷의 사이즈를 나타낸다. 보통 패킷 사이즈는 고정된 크기를 갖는다. 시퀀스는 각 패킷당 부여되는 시퀀스 번호를 나타낸다. 이 번호로 패킷 순서와 구별이 가능하다. 패딩사이즈는 패킷 데이터가 들어간 후 남은 여분의 사이즈를 나타낸다. 데이터 삽입 후 패딩 사이즈만큼 무의미한 값으로 채워 하나의 패킷을 완성하게 된다.

```

switch((flags>>5)&3){
case 3: plen=LOAD_LE32(p);p+=4;break;
case 2: plen=LOAD_LE16(p);p+=2;break;
case 1: plen=p[0];p++;break;
default: plen=0;
}

switch((flags>>1)&3){
case 3: sequence=LOAD_LE32(p);p+=4;break;
case 2: sequence=LOAD_LE16(p);p+=2;break;
case 1: sequence=p[0];p++;break;
default: sequence=0;
}

switch((flags>>3)&3){
case 3: padding=LOAD_LE32(p);p+=4;break;
case 2: padding=LOAD_LE16(p);p+=2;break;
case 1: padding=p[0];p++;break;
default: padding=0;
}

* plen : 패킷 사이즈
* sequence : 패킷 시퀀스
* padding : 패딩 사이즈
* 각 switch문의 case 1,2,3은 byte, word, dword의 데이터 타입을 나타냄
    
```

그림 6. 패킷 분석 코드

```

if(p[0]&0x80)
{
keyframe=1;
id = streamno;
}
    
```

그림 7. Key-Frame 추출을 위한 비교

```

switch((segtype>>4)&3){
case 3: seq=LOAD_LE32(p);p+=4;break;
case 2: seq=LOAD_LE16(p);p+=2;break;
case 1: seq=p[0];p++;break;
default: seq=0;
}

* seq : 프레임의 시퀀스 번호
* 각 switch문의 case 1,2,3은 byte, word, dword의 데이터 타입을 나타냄
    
```

그림 8. 프레임 Sequence number의 정보 수집

3.3 Key-Frame 추출

그림 7의 코드는 Key-Frame 추출을 위해 패킷의 프레임 중 Key-Frame을 나타내는 특정비트인 0x80이 존재하는지를 검사한다. 패킷에서 0x80이 존재하게 되면 keyframe 변수를 세팅하여 Key-Frame 추출을 위해 사용한다. 하지만 그림 4의 (2)의 패킷 구조처럼 패킷 안에 여러 개의 프레임이 존재할 수 있기 때문에 Key-Frame만을 구별해 낼 수 있는 방법이 필요하다. 구별 방법으로 프레임 헤더부분에 존재하는 독립적인 Sequence number를 이용한다. 각 프레임은 서로 다른 Sequence number를 지니고 있다. 따라서 Sequence number의 차이를 이용하면 Key-Frame과 일반 프레임을 구별해 낼 수 있다. 그림 8의 코드는 Key-Frame과 일반 프레임을 구별할 수 있도록 패킷 내부의 프레임 Sequence number를 분석하는 역할을 한다.

4. 결론 및 향후 연구방향

최근 초고속 인터넷의 급속한 성장으로 인해 멀티미디어 스트리밍 서비스의 요구가 크게 증가하고 있다. 이러한 시대적 요구에 WMV 파일 포맷은 오디오, 비디오, 슬라이드 쇼, 그리고 동기화된 이벤트 등을 지원하는 등 스트리밍 서비스에 적합한 구조와 기능을 가지고 있다. 본 논문에서는 WMV파일 포맷 분석을 통해 영상의 가장 중요한 Key-Frame 추출과 영상의 헤더부분을 제외한 실제 영상 스트림의 시작위치를 찾는 알고리즘을 설계 및 구현하여 실제 스트리밍 서비스에 적용하는 등 다양한 멀티미디어 서비스에 적용 가능한 방법을 제시하였다.

앞서 WMV의 Key-Frame이외에도 다양한 분석을 통해 네트워크 대역폭 문제, HDTV급의 고화질 영상 서비스의 QoS문제등 다양한 멀티미디어 서비스에 적합하도록 연구할 필요성이 있다.

5. 참고문헌

[1] Microsoft, www.microsoft.com.
 [2] Microsoft WMV 고화질 미디어, <http://www.wmvhd.com/>.
 [3] Microsoft ASF Specification.