

인트라넷 및 인터넷 멀티미디어 서비스를 위한 소프트웨어 비디오 코덱

김용한, 조남익, 김기철

서울시립대학교 전자전기공학부

Email : yhkim@tina.scu.ac.kr, nicho@tina.scu.ac.kr, kkim@scucc.scu.ac.kr

S/W-Based Video Codec Systems for Intranet and Internet Multimedia Services

Yong Han Kim, Nam Ik Cho, Kichul Kim

School of Electrical Engineering, Seoul City University

요약 : 본 논문에서는 소프트웨어 기반의 두가지 다른 코덱 시스템을 연구한다. 하나는 ITU H.263에 기반한 고정된 구조를 갖는 프레임 단위의 비디오 코덱이고, 다른 하나는 현재 진행 중이고 1998년에 완성될 ISO MPEG-4 표준안에 기초하여 재배열이 가능한 유연한 구조를 갖는 오브젝트 단위의 비디오 코덱이다. 이러한 비디오 코덱들의 인트라, 인터넷 환경에서 소프트웨어에 기반한 실시간 및 유연한 구조의 구현 가능성을 알아보기 위하여 이들의 실험적인 구현을 시도한다.

1 서론

카드를 사용하였다.

본 논문의 목적은 인트라넷 및 인터넷 멀티미디어 응용을 위한 소프트웨어 코덱의 가능성을 알아보기 위한 것으로서 두 가지 다른 소프트웨어 코덱을 구현해 보았다.

첫번째로 우선 H.263[1]의 구현을 위한 시스템의 사양을 예측하기 위하여 이의 복잡도를 계산하였다. 그 결과 3-step 탐색 알고리즘을 이용하고, 영상 입력부터 메모리까지 병목 현상이 없다는 가정하에 90MHz 펜티엄 컴퓨터를 이용하면 최소한 네 장의 QCIF 영상을 부호/복호화할 수 있다는 결론을 얻었다. 이러한 분석을 바탕으로 본 논문에서는 실제로 90MHz 펜티엄 노트북 컴퓨터에 H.263에 기반한 소프트웨어 코덱을 구현하였다. PCMCIA 영상 입력 보드와 사운드 카드를 입력으로 하였고 데이터 통신을 위해서는 노트북 컴퓨터의 장점인 이동성을 살리기 위하여 무선 LAN

두번째로는 특정 플랫폼에 관계 없이 동작하는 유연한 구조의 비디오 코덱의 가능성을 보이기 위하여 ISO의 MPEG-4[2] 구현을 시도하였다. 본 논문에서는 새로운 복호화 툴의 일부를 네트워크를 통하여 전송 받아 전체적인 복호화를 재배치 하는 것이 가능하다는 것을 보이기 위하여 MPEG에서 제공하는 MPEG-4의 C 프로그램[4]을 Java encapsulation[3] 방법으로 수행하였다. 실제로 MPEG-4 시스템은 완전한 양방향 통신을 목표로 하고 있지만 본 논문에서는 우선 재배치 구조의 가능성을 연구하기 위하여 단방향의 서비스를 가정하였고 특정한 넷스케이프 웹 브라우저를 사용하는 것으로 하였다.

이 연구의 일부는 한국전자통신연구소의 용역에 의한 것입니다 (계약 번호 : ETRI 96-51).

2 노트북 컴퓨터를 이용한 비디오 전화기

일반적인 데스크탑 컴퓨터에서 수행되는 프로그램은 대부분의 경우 수정 없이 그대로 노트북 컴퓨터에서도 수행이 된다. 그러나 비디오 전화기 프로그램의 경우 노트북 컴퓨터용 영상 보드가 흔하지 않고 공간과 속도의 문제도 있으므로 많은 수정과 새로운 프로그래밍이 필요하다. 또한 데스크탑 컴퓨터의 경우 PSTN이나 유선 LAN을 이용하여 통신을 수행하지만 노트북 컴퓨터의 경우 장점인 이동성을 살리자면 무선 통신이 되도록 하여야 한다.

따라서 비디오 코덱을 노트북 컴퓨터에 구현하기 위해서는 효율적인 비디오 부호/복호화 알고리즘은 물론이고, 효율적인 영상 입출력 및 무선 LAN을 위한 PCMCIA 카드와 이들을 구동하고 부호/복호화 프로그램과 묶어주는 인터페이스 프로그램 모듈들이 필요하다. 그림 1이 본 논문에서 구현한 비디오 전화기의 전체적인 프로그램 구성도이다. 그림에서 "image grabber" 모듈은 NTSC 아날로그 영상 입력을 받아 YUV 디지털 포맷으로 변환하고 이를 메모리에 저장하는 프로그램 모듈이다. 이 부분을 위한 하드웨어로는 NogaTech사의 "PCMCIA Capture Vision"이라는 그래픽 오버레이 카드를 이용하였고 인터페이스 프로그램으로는 여기서 제공하는 16 비트 API(application program interface)를 사용하였다. "Image coder/decoder" 부분은 앞에서 언급한 바와 같이 호환성을 위하여 H.263 표준안에 기반한 것이고 "audio coder/decoder"는 여러 가지 음성 압축 방식을 모두 수용할 수 있는 것이다. "Data multiplexer/demultiplexer"는 음성 및 영상 부호화/복호화기에서 나오는 비트 스트림을 묶어 주거나 분리하기 위한 것이다. 멀티플렉서를 통하여 합해진 최종 비트 스트림은 무선 LAN 카드를 통하여 전송되도록 하였다. 사용된 LAN 카드는 AT&T사의 WaveLAN 카드로서 2.4 GHz 주파수 대역에서의 통신이 가능하다. 네트워크 프로토콜은 TCP/IP를 사용하였다.

본 논문에서 구현된 노트북용 영상 전화기에서는 영상 입력의 제어를 위하여 16 비트 API를 사용한 관계로 영상 입력부분의 프로그

램이 32 비트 모드로 짜여진 나머지 프로그램들과 동기화된 인터페이스가 될 수 없었다. 그 결과로 부호화의 효율이 추정된 것보다는 좋지 않았고 초당 두 프레임의 sub-QCIF 영상 통신이 가능하였다. 이를 개선하기 위하여 현재 32 비트 영상 입력 프로그램과 MFC(Microsoft Foundation Class) 라이브러리를 이용한 프로그램을 개발 중이다.

3 플랫폼 독립 코덱 구조

본 논문에서는 특정 플랫폼에 관계 없이 거의 모든 컴퓨터에서 수행될 수 있는 소프트웨어 코덱에 대한 연구도 수행하였다. 기본적인 복호기의 구조는 그림 2에 나타나 있는 바와 같이 기본적으로 상주 도구("resident tools"), 사용자가 설치한 도구("user-installed tools"), 전송 받은 도구("downloaded tools")의 세 가지 다른 형태로 나눌 수 있다. MPEG-4 표준안을 따르는 모든 복호기는 MPEG-4에서 정의한 기본적인 "resident tools"를 갖고 있어야 한다. 따라서 이 경우는 특정 플랫폼에만 동작하는 프로그램으로 짜여져 있어도 상관없다. "User-installed tools"는 복호기 사용자가 설치할 수 있는 프로그램으로서 상주 도구가 연장된 것이라 할 수 있다. 이는 반드시 플랫폼에 독립적인 필요는 없지만 그렇지 않으면 여러 제약이 따를 수 있다. 즉, 사용자 설치가 없을 경우 이러한 도구가 있다고 가정하고 부호화된 영상은 당연히 복호화할 수가 없다. "Downloaded tools"는 네트워크를 통하여 전송 받는 것으로서 비트 스트림을 서비스하는 컴퓨터에 모든 플랫폼에서 수행이 가능한 실행 파일들이 준비되어 있지 않는 이상 플랫폼 독립적인 프로그램을 이용하여 부호/복호화가 수행되도록 하여야 한다. 이러한 전송 가능한 도구가 있어야 복호화 할 때 원하는 기능을 마음대로 추가할 수 있으며 이러한 유연성이 있는 기능이 바로 MPEG-4가 기존의 H.261, H.263, MPEG-2와 다른 점이라 할 수 있다.

MPEG-4 드래프트 사양에 따르면 어떤 특정한 복호화 모드 또는 알고리즘은 위에서 언급한 복호화 도구를 기본으로 구성된다. 즉, 복호화기가 주어진 몇개의 동작 모드 중에 하

나를 선택하는 형식이라면 상주 도구만을 이용해도 가능하다. MPEG-4에서는 이 모드를 "Flex-0"라고 부르는데, 이는 MPEG-2의 프로파일 구성과 같은 형태이다. 부호화기에서 현재 상주 프로그램을 재배열하여야 복호화가 되도록 하는 모드를 정의한다면 비트 스트림에 상주 도구 재배치에 대한 정보도 함께 전송되어야 한다. 이 모드를 "Flex-1"이라 한다. 마지막으로 "Flex-2"는 새로운 도구를 전송하는 것까지를 포함한다. 전송 가능한 도구는 특정 플랫폼에 종속되어서는 안되므로 본 논문에서는 자바 (Java) 프로그램을 이용하였다. 따라서 자바 프로그램 번역이 가능한 플랫폼에서는 새로운 복호화 도구를 전송하여 전체적인 복호화를 재구성하는 것이 가능하다.

그림 3은 위에서 언급한 모든 "Flex" 모드를 수용할 수 있는 시스템의 구조를 나타낸다. 전송된 프로그램들의 재배열이 가능할 수 있도록 복호화기의 메인 프로그램은 자바로 작성하였다. 전체적으로는 C 프로그램을 encapsulation한 형태이므로 "Java applet"이 아니라 "Java application"이다. 그리고, 압축된 MPEG-4 비트 스트림 파일을 갖고 있는 서버의 HTML 파일을 보기 위해서 넷스케이프와 같은 웹 브라우저를 사용한다. 클라이언트 컴퓨터의 사용자가 비트 스트림이 있는 링크를 클릭하면 웹 브라우저가 넷스케이프인 경우 "Plug-in" 프로그램을 구동시키도록 하였고 이는 다시 자바를 이용한 복호기를 동작시키도록 하였다. 이러한 plug-in을 사용함으로써 웹 브라우저로부터 자바 복호화기까지의 데이터 전송이 가능하며 넷스케이프 창 내에서의 복호화 결과 디스플레이가 가능하다. 이러한 데이터 전송은 "pipe"나 임시 파일을 통하여 가능하다. 이렇게 plug-in을 통하는 이유는 보안 문제 때문에 클라이언트 컴퓨터의 하드 디스크에 "Java applet"을 쓰는 것이 불가능하기 때문이다.

MPEG-4의 출력으로 나오는 비트 스트림에는 영상 정보뿐만 아니라 필요하다면 전송하는 도구 및 이의 배치에 대한 정보도 포함된다. 모든 전송되는 도구는 자바 바이트 코드로 프로그램되어 자바 수행이 가능한 플랫폼에서는 즉시 전송 받은 모듈과 갖고 있던 모듈을 재배치하여 사용할 수 있도록 한다. MPEG-

4 복호기의 메인 프로그램에서는 이를 위하여 전송된 배치 정보를 해석하고 적절한 도구를 선택하여 사용하여야 한다. 만약 새로운 도구가 전송되었다면 복호기의 메인 프로그램에서는 이를 지정된 장소에 놓아서 복호화가 진행되는 동안 적절하게 사용될 수 있도록 한다. 압축된 영상을 풀기 직전에는 복호화된 영상이 만들어지는 즉시 화면에 디스플레이를 해 줄 수 있는 프로세스가 수행되도록 한다.

설계된 소프트웨어 기반 MPEG-4 복호기는 네트워크 클라이언트로 동작하는 IBM 호환 PC에 설치되었다. 서버는 MPEG-4 비데오 Verification Model[5]에 따른 영상의 비트 스트림을 갖고 있도록 하였다. 이러한 서버와 클라이언트 환경에서 "Flex-0"와 "Flex-1"의 성능을 시험하기 위하여 부호화기에서는 두가지 다른 양자화 행렬을 선택하여 사용할 수 있도록 하였다. 그리고 "Flex-2"의 시험을 위해서는 매우 간단한 도구를 자바로 프로그램하여 네트워크로 전송하여 클라이언트에서 사용할 수 있도록 하였다. 그 결과 플랫폼 독립적인 유연성이 있는 복호기의 구조가 가능하다는 것을 볼 수 있었다.

4 결론

첫번째로, 이동 비디오 전화 및 기타 멀티미디어 서비스에 사용할 수 있는 H.263에 기반한 소프트웨어 코덱을 PCMCIA 비디오 카드, 무선 LAN 카드를 갖는 90MHz 펜티엄 노트북 컴퓨터에 설치하였다. 설치된 코덱 시스템을 이용하여 1초에 2 프레임 정도의 sub-QCIF 영상을 부호 및 복호화할 수 있음을 확인하였다. 현재의 시스템에서는 영상 입력부와 나머지 프로그램이 서로 다른 프로세스로 분리되어 있어서 예상한 것 만큼의 프레임 윌이 나오지 않았다. 따라서 이의 성능을 향상시키기 위하여 현재 32 비트 영상 인터페이스 프로그램 및 MFC (Microsoft Foundation Class) 라이브러리를 이용한 소프트웨어 코덱을 프로그램하고 있는 중이다.

두번째로, 특정 플랫폼에 독립적인 복호기의 구조가 가능함을 보이기 위하여 클라이언트가 갖고 있는 복호 도구에 서버에서 전송된 자바 바이트 코드로 프로그램된 도구를 결합

하여 복호기를 재구성함으로써 함께 전송된 압축 영상을 복호화하는 방법을 시험하였다. 앞으로는 이러한 기본적인 구조를 이용하여 완전한 MPEG-4의 구현을 위한 연구를 수행할 예정이다.

참고 서적

- [1] International Telecommunication Union, Line Transmission of Non-telephone Signals, Draft ITU-T Recommendation H.263, April 1995.
- [2] ISO/IEC JTC1/SC29/WG11 N1401, "MPEG-4 MSDL Specification Version 1.3," Sep., 1996.
- [3] Ken Arnold and James Gosling, *The Java Programming Language*, Addison-Wesley Publishing Co., 1996.
- [4] ISO/IEC JTC1/SC29/WG11 MPEG96/M1089 "MoMuSys C implementation of the VM 2.2," July. 1996.
- [5] ISO/IEC JTC1/SC29/WG11 N1277, "MPEG-4 Video Verification Model Version 3.0," July 1996.

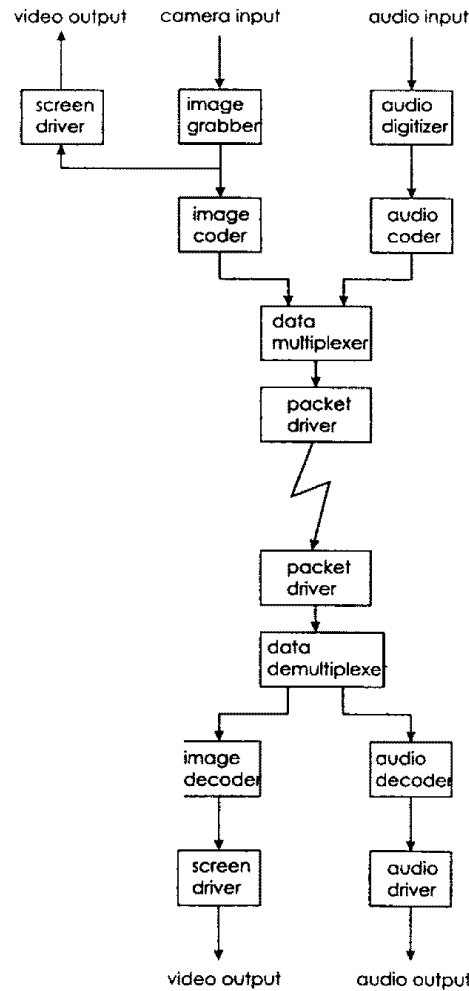


그림 1: 소프트웨어 비디오 전화기의 구조.

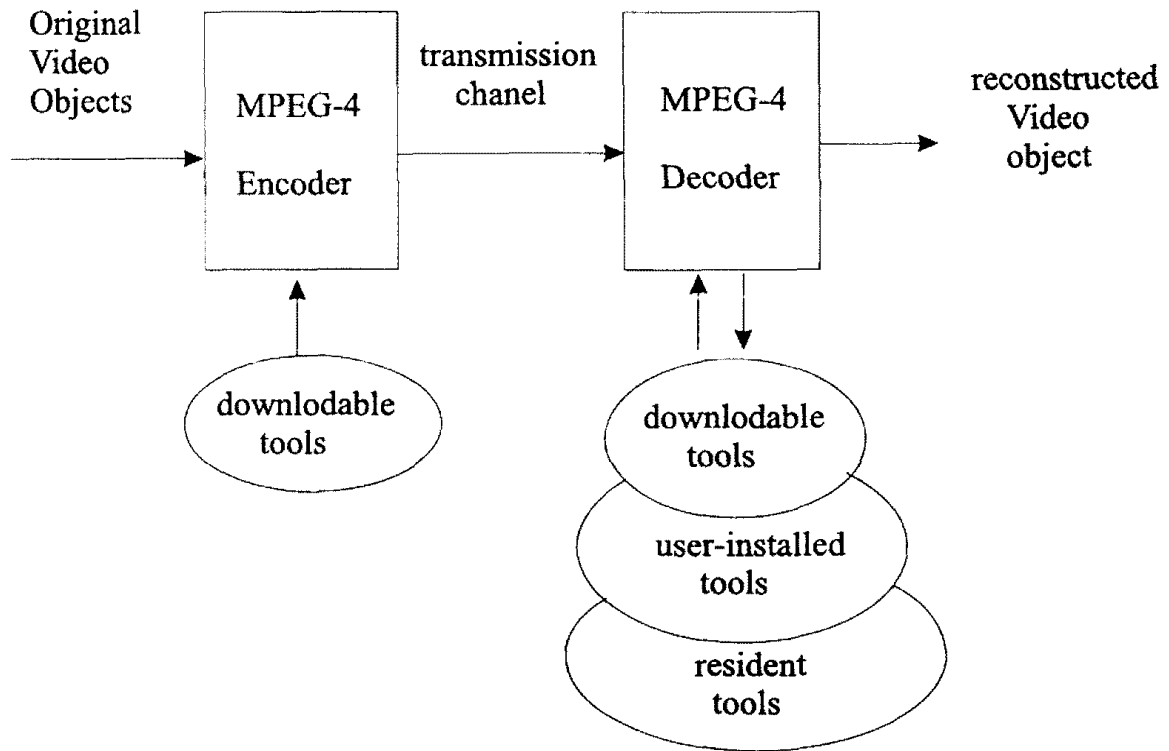


그림 2: 플랫폼 독립적인 비디오 복호기의 기본 구조.

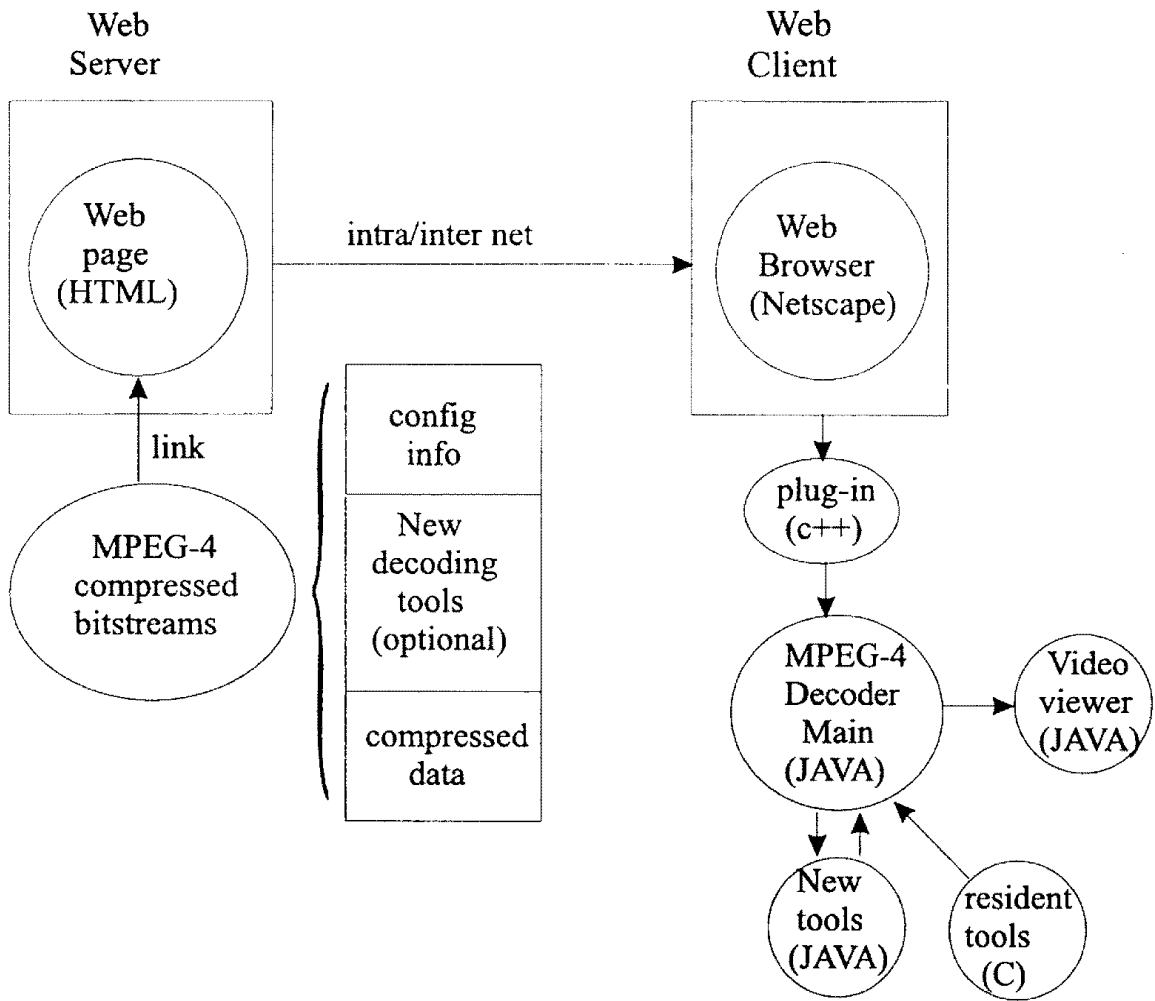


그림 3: 소프트웨어 기반 MPEG-4 복호기의 구조.