

TI DSP 64x에서의 인코딩 시스템 및 디코딩 시스템 구현

전형국⁰ 마평수
 한국전자통신연구원 임베디드 S/W 기술센터
 {hkjun⁰, mpah}⁰@etri.re.kr

Implementation of Encoding System Based on the TI DSP 64x and Decoding System

Hyung-Kook Jun⁰ Pyeong-Soo Mah
 Embedded S/W Technology Center, Electronics and Telecommunications Research Institute

요 약

JPEG 2000 표준을 이용하여 계산상 효율적이면서도 DSP에 적합한 영상 압축 알고리즘을 개발한다. 개발된 기법이 갖는 가장 큰 장점은 비교적 복잡한 연산량을 필요로 하여 실시간 동영상 부호화기에 사용되지 못하고 있는 JPEG 2000 알고리즘을 DSP에 맞게 구현함으로써 고품질의 동영상을 실시간으로 처리할 수 있게 한다. 또한 사용자의 요구 및 사양에 맞추어 보다 효과적으로 콘텐츠를 제공하기 위해, 시, 공간적 스케일러비티를 지원하는 코덱을 개발하여 사용자의 요구와 필요 서비스에 따라 적절한 품질의 비디오를 제공함으로써 더 많은 응용에 쉽게 적용시킬 수 있는 방법들을 제시한다.

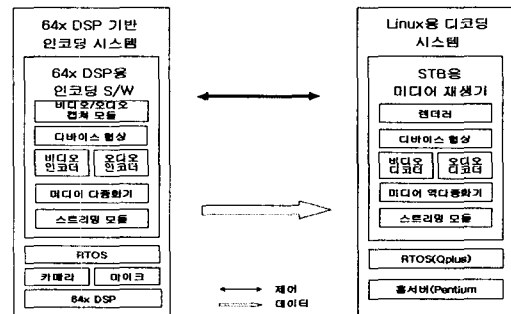
1. 서 론

64x DSP용 비디오 실시간 인코딩 시스템은 고 해상도의 비디오 신호를 실시간으로 JPEG-2000으로 부호화하여 스트리밍 서버와 인터넷을 통해 멀티미디어 신호 수신에 가능한 기기로 전송하는 기능을 수행한다. 64x DSP용 비디오 실시간 인코딩 시스템에는 CIF급 비디오를 기저로 하는 콘텐츠 불변형 실시간 JPEG-2000 비디오 인코더가 요구되며 수신자가 실감성있는 멀티미디어 신호 수신에 가능하게 한다. 디코딩 시스템은(이하 재생기)는 네트워크를 통하여 인코딩 시스템에게 재생기 상대 메시지를 전송하고, 인코딩 시스템으로부터 전송된 미디어 스트림을 처리하여 사용자의 화면에 보여주는 역할을 수행한다. 본 재생기는 JPEG-2000 재생을 주요 기반으로 하고 있지만, 동일한 인터페이스로 다양한 포맷들을 지원할 수 있는 독립적인 어플리케이션으로 동작한다. 본 논문은 2장에서 본 논문에서 제안하는 64x DSP용 비디오 실시간 인코딩 시스템과 Linux 용 디코딩 시스템을 개략적으로 소개하고 3장에서 64x DSP용 비디오 실시간 인코딩 시스템과 디코딩 시스템에서 사용되는 공통 모듈을 4장에서는 실험 환경 5장에서는 결론 및 향후 연구 과제를 설명한다.

2. 인코딩 및 디코딩 시스템

2.1 64x DSP 인코딩 시스템과 디코딩 시스템의 구조

<그림 1>은 본 논문에서 구현하고자 하는 시스템의 전체적인 구성도이다. 64x DSP 기반 인코딩 시스템은



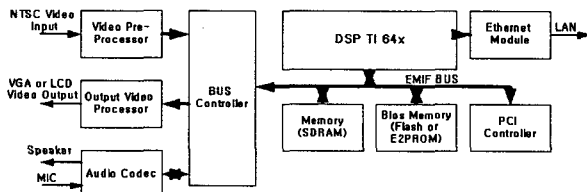
<그림 1> 인코딩 시스템 및 디코딩 시스템

64x DSP를 기반으로 비디오 오디오 캡처 모듈을 통해서 비디오 오디오 데이터를 획득한다. 디바이스 협상기에 의해서 디코딩 시스템의 성능 정보를 획득하고 디바이스의 성능에 맞는 비디오 오디오 데이터를 획득하기 위해 비디오 오디오 인코더 모듈을 통해 JPEG-2000으로 인코딩된다. 인코딩된 데이터는 다중화기를 거쳐 스트리밍 모듈로 전달되어 디코딩 시스템으로 전달된다. Linux 용 디코딩 시스템은 홈서버에서 수행되며 인코딩 시스템으로부터 데이터를 받는 스트리밍 모듈과 다중화된 데이터를 분리하는 역 다중화기 인코딩된 데이터를 처리하는 비디오 오디오 디코더와 렌더러로 구성된다.

2.2 64x DSP 인코더 시스템의 하드웨어 구조

64x DSP용 비디오/오디오 인코딩 시스템의 하드웨어

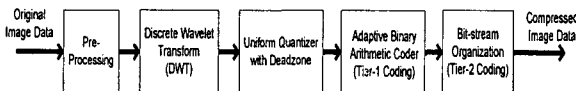
Texas Instrument사의 DSP 64x 시리즈는 고성능의 연산 처리능력과 주변 멀티미디어 관련 외부 디바이스와의 호환성등으로 실시간 멀티미디어 데이터의 인코딩 및 디코딩 솔루션인 고정 소수점 DSP 이다. 64x DSP는 하나의 칩에 모든 외부 디바이스가 집적되어 있는 SoC (System On Chip) 솔루션이 아니므로 TI DSP와 호환 가능한 Device들을 중심으로 하드웨어를 설계해야 한다. 64x 소프트웨어 모듈들이 효율적으로 동작하기 위해서 비디오/오디오 실시간 인코딩을 위한 비디오 오디오 입력부가 있으며 인터넷과의 연결을 위해 Ethernet 인터페이스가 존재해야 한다. 향후, 시스템의 독자적인 동작을 지원하기 위하여 PCI 인터페이스가 존재해야 하며 메모리 모듈과 관련된 소프트웨어의 효율적인 동작을 위한 버스 컨트롤러 등이 구현된다. 구현하고자 하는 시스템 하드웨어는 다음의 모듈을 가지게 되며 구조도는 <그림 2>와 같다.



<그림 2> 시스템 하드웨어의 구조도

2.3 인코더 시스템 소프트웨어 구조

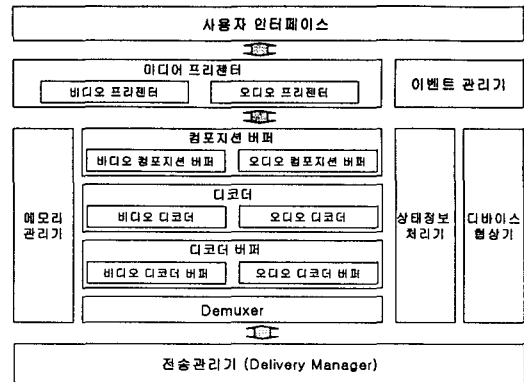
64x DSP 환경에서 비디오 부호기는 Temporal 및 Scalability 특성과 실시간 비디오 스트리밍 성질을 모두 만족하여야 한다. 이를 위해 MJPEG 인코더는 Multiband 입력을 사용하여야 하며 실시간 비디오를 위해 Wavelet 변환을 사용하는 MJPEG 방식으로 구성된다. 일반적인 JPEG 2000 인코더의 기본 구조 블록은 <그림 3>과 같다. JPEG 2000 인코더를 이루고 있는 구성요소는 전처리, DWT, 양자화, 산술부호화(Tier-1), 그리고 비트열 생성(Tier-2)등이 있다.



<그림 3> JPEG 2000 기본 구조 블록

2.4 미디어 재생기 구조

재생기의 구성은 스트리밍 서버와 메시지 및 스트리밍 데이터 송수신을 위한 전송관리기, 전송관리기로부터 받은 스트리밍 데이터를 오디오 및 비디오로 분리하기 위한 Demuxer, Demuxer로부터 분리된 데이터를 저장하기 위한 디코더 버퍼, 디코더 버퍼의 압축된 미디어 스트림을 복호화하는 디코더, 복호화된 디코더의 출력을 저장될 컴포지션 버퍼, 디코더 버퍼 및 컴포지션 버퍼를 관리 하는 메모리 관리기, 복호화된 오디오 및 비디오 데이터를 스피커 및 화면으로 출력하기 위한



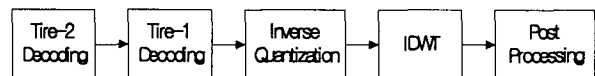
<그림 4> 재생기 모듈 구성도

미디어 프리젠타, 사용자 및 서버로부터 발생한 이벤트를 처리하기 위한 이벤트 관리자, 단말기의 기능 및 성능에 대한 정보 요구에 대한 처리를 위한 디바이스 협상기와 미디어 플레이시 제공되는 서비스에 대한 정보를 관리하여 재 접속할 수 있는 정보를 제공하기 위한 상태정보 처리기, 최종적으로 화면에 나타내고 사용자와 직접 인터페이스를 담당하는 사용자 인터페이스 모듈로 구성 된다.

2.5 홈 서버급 클라이언트 환경에서 비디오 복호기 구조

64x DSP 보드에서 부호화된 MJPEG-2000 Video는 홈 서버급 클라이언트에서 복호되어 렌더러 혹은 비디오 플레이어에 전달된다. 홈 서버 급 클라이언트는 일반적으로 INTEL Pentium급 CPU를 사용하며 Linux와 같은 OS를 탑재 하므로 비디오 복호기는 하드웨어적인 추가 고려 없이 OS 환경과 디스플레이 프로그램과의 인터페이스만을 고려하여 제작 되어진다.

일반적인 MPEG-4 복호기와는 달리 JPEG-2000을 기반으로 하는 Video는 Tire-1, Tire-2 부호화 과정의 역 부호화 과정을 통해 가변장 복호과정을 대체한다. 이 과정을 거친 후 역 양자화 과정을 거친 다음 역 이산시간 웨이블릿 변환을 통해 최종적인 1 프레임당 비디오 신호를 복호 하고 이를 디스플레이 장치에 전달하기 위한 후 처리를 거쳐 최종적으로 복호과정이 완료된다.



<그림 5> MJPEG-2000 복호과정

3. 인코딩 디코딩 시스템의 공통 모듈

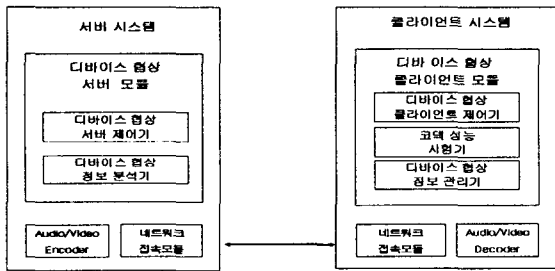
3.1 전송 관리자와의 인터페이스

MJPEG 표준에 맞추어 스트리밍 데이터를 생성하여 전송관리자에게 데이터를 넘기게 될 때 전송 관리자는 멀티 미디어 데이터의 통신망 전송을 위한 가장 기본적인 데이터 포맷을 미리 알고 있어야 효율적인 전송이 가능해진다. 이를 위해서 부호기는 다음의 데이터를 부호화 과정 중에 미리 추출하고 있어야 한다.

- 패킷 데이터의 크기
- 패킷의 인덱스 정보
- 시간 정보 (향후 오디오와의 결합을 위함)

3.2 디바이스 협상 제어 모듈

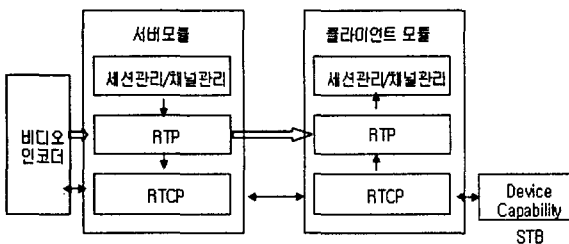
사용자 단말인 클라이언트에서 가장 최적의 멀티미디어 스트림 서비스를 받기 위해서는 클라이언트의 성능이나 기능인 단말기의 capability 특성에 가장 적합한 인코딩 데이터를 전송 함으로써 가장 양질의 서비스를 제공받을 수 있다. 그러나 단말기의 특성을 사용자가 사용할 때마다 입력한다는 것은 번거롭고 어려운 일이므로 서버와 단말기간에 직접 협상하여 단말기의 정보를 얻음으로써 일반 사용자 하에 제공 용이한 사용과 최상의 서비스를 제공 받을 수 있도록 한다. 본 기능은 서버에서 인코딩 하기 전 즉, 인코더의 초기화 시에 필요한 정보로 사용된다. 디바이스 협상 서버 모듈은 디바이스 협상 제어기와 디바이스 협상 정보 분석기로 구성되며, 디바이스 협상 클라이언트 모듈과 상호 협력하여 정보처리가 이루어 진다. 서버와 클라이언트 간의 디바이스 협상 기본 구성도를 <그림 6>에 나타냈다.



<그림 6> 디바이스 협상 기본 구성도

3.3 스트리밍 전송관리자 모듈

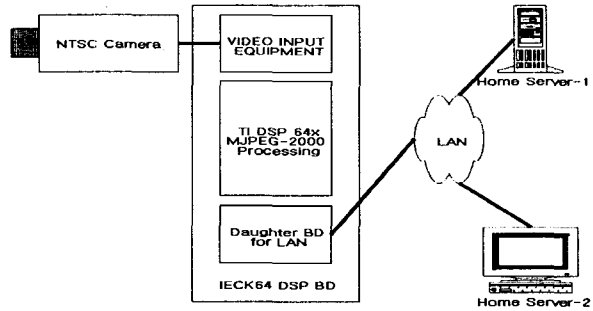
스트리밍 전송관리자는 세션 및 채널을 유지하고 관리하는 세션관리/채널관리모듈과 압축된 데이터를 네트워크에 알맞은 크기로 전송하는 RTP프로토콜 모듈과 데이터의 흐름을 제어하는 RTCP 모듈로 구성되어 있다. 스트리밍 전송관리자는 64x를 이용한 콘텐츠 불변성을 지원하는 connected multimedia service에 사용된다.



<그림 7> 스트리밍 전송관리자의 구조도

4. 실험환경

비디오 부호화 부분은 NTSC 카메라와 TI 64x DSP를 사용하는 IECK64 보드와 LAN 환경, 그리고 홈 서버 클라이언트가 필요하다. <그림 8>은 전체 실험환경을 보여준다.



<그림 8> 64x DSP BD에 기반한 Multimedia 실험환경

클라이언트인 홈 서버 1과 홈 서버 2에서 DSP 보드가 가지고 있는 IP를 직접 입력하거나 혹은 참조된 하이퍼 텍스트를 통해 입력하게 되면 클라이언트측으로 64x에서 복호된 비디오 신호가 스트림 서버를 통해 클라이언트측으로 전송된다.

5. 결론 및 향후연구과제

본 논문에서는 TI DSP 64x에서 구현된 MJPEG 2000 인코더 시스템과 디코더 시스템을 구현하였다. 본 시스템은 향후에 블루투스 지원 PDA를 통하여 미디어 재생기를 제어하고 인코더 시스템과 디코더 시스템 사이에 미디어 서버를 두어 보다 효율적인 Connected 멀티미디어 시스템을 보이려 한다.

6. 참고문헌

[1]M.D. Adams and F. Kossentini, " JasPer: a software-based JPEG-2000 codec implementation," in: Proceedings of the IEEE International Conference on Image Processing, Vancouver, CA, September 2000.
 [2]M. Antonini, M.Barloud, P. Mathieu, and I. Daubechies, " Image coding using wavelet transform," IEEE Trans. Image Process. 1 (2) (April 1992) 205.220.
 [3]R.C. Calderbank, I. Daubechise, and W. Sweldens, B.-L. Yeo, " Wavelet transforms that map integers to integers," App. Cpmput. Harmon. Anal. 5 (3) (1998) 332.369.
 [4]C. Chrstopoulos, J. Askelof, M.Larsson, and E.cient, " methods for encodings regions of interest in the upcoming JPEG2000 still image coding standard," IEEE Signal Process.Lett. 7 (9) (Sep. 2000) 247.249.
 [5]I. Moccagata, S. Sodagar, J. Liang, and H. Chen, " Error Resilient Coding in JPEG-2000 and MPEG-4," IEEE J. Select. Areas Commun. 18 (6) (june 2000) 899.914.
 [6]" RTP Payload Format for MPEG-4 Audio/Visual Stream" , rfc3016, Internet Engineering Task Force, November 2000