실시간 멀티미디어 시스템을 위한 G.723.1 음성 코덱의 구현

이 상 광, 호 요 성
광주과학기술원 정보통신공학과

Implementation of A G.723.1 Speech Codec for A Real-Time Multimedia System

Sang-Kwang Lee and Yo-Sung Ho
Kwangju Institute of Science and Technology (K-JIST)

요약

본 논문에서는 실시간 멀티미디어 서비스 시스템을 위해 구현한 G.723.1 음성 코덱에 대해서 기술한다. G.723.1 표준은 멀티미디어 서비스를 위해 낮은 비트율에서의 음성이나 오디오 신호를 압축하는데 이용될 수 있는 부호화 방법을 규정하며, 5.3 kbit/s 와 6.3 kbit/s 의 이중 비트율을 갖는다. 이 부호화기는 시스템의 복잡성을 고려하여 두 비트율에서 최상의 음질을 갖도록 음성 신호를 최적화시킨다. 음악이나 다른 오디오 신호들은 음성 신호만큼 충실히 따르지는 않지만, 이 부호화기를 이용하여 압축하거나 압축을 풀 수 있다. 본 작업에서는 실제적인 멀티미디어 서비스 응용을 위해, 실시간 운영 시스템인 VxWorks 환경에서 음성 코덱의 API 를 구현하였다.

1. 서론

최근 인터넷이 급속하게 보급되어 일상생활에 다양하게 활용되고 있다. 인터넷 채널을 통해 교환되는 정보의 형태는 문자와 정지 영상 뿐만 아니라, 음성, 오디오 및 동영상 등으로 실시간 멀티미디어 부분까지 확대되고 있다.

실시간 멀티미디어란 인터넷 또는 인트라넷에서 오디오나 비디오 등의 멀티미디어 데이터를 다운로드가 아닌 실시간적으로 주고 받는 것을 말한다[1]. 실시간 멀티미디어는 일반적으로 멀티미디어 데이터의 On-Demand 서비스와 생방송(live broadcasting) 서비스에 이용된다. On-Demand 서비스는 멀티미디어 데이터를 사용자(client)가 원할 때 제공해 주는 주문형 서비스로서, 디스크에 대용량으로 저장된 멀티미디어 데이터를 실시간적으로 전송하고자 할 때 실시간 멀티미디어 시스템을 이용한다. 생방송 서비스에서는 비디오 캠코더의 영상 또는 마이크의 음성을 사용자에게 실시간적으로 제공한다. 이러한 생방송 서비스를 이용하여 인터넷 가상 방송국을 차릴 수 있다.

일반적으로 실시간 멀티미디어 시스템은 다음과 같이 구성된다.

1) 캡쳐 보드와 인코더: 비디오 카메라, 마이크, VTR 등의 아날로그 소스를, 사운드나 비디오 캡쳐 카드를 사용해 디지털화하고, 인코더를 사용해 실시간 전송이 가능하도록, 파일 또는 실시간 신호 형태로 압축한다.
2) 서버: 인코더에서 압축된 파일과 신호는 네트워크를 통해 서버로 전송된다. 서버는 클라이언트들의 접속을 받아들이고, 클라이언트가 요구하는 멀티미디어 데이터를 실시간적으로 제공한다.
3) 재생기: 인코더가 압축한 데이터를 서버로부터 전송 받아 실시간으로 재생한다.

그림 1 은 실시간 멀티미디어 시스템 기본 구조를 보여주고 있다.

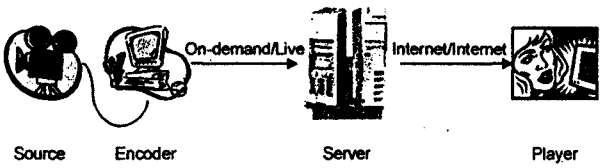


그림 1. 실시간 멀티미디어 시스템의 기본 구조

실시간 멀티미디어는 생방송 서비스를 이용하여 인터넷을 통한 가상의 방송국을 운영할 수 있으며, 가상 교육 및 원격 강의에 최적의 VOD 솔루션으로 적용할 수 있다. 또한, 현재 이미지 배너 형태의 웹페이지 광고를 멀티미디어 광고형태로 전환할 수 있으며, 동영상 및 음성을 이용하여 더욱 원활한 상품 소개를 할 수 있는 전자상거래나 사내 교육 및 안내 시스템, 방송국 등 사내 인트라넷과 연동할 수 있다.

앞으로의 정보가전 산업은 단순한 시스템 조립 업체에 의한 상품의 대량생산이 제품 경쟁력의 제고를 위한 방법이 아니라 얼마나 우수하고 다양한 서비스를 제공하고 소프트웨어를 개발하는데 있다. 그 대표적인 예가 디지털 TV 이며, PC 의 일부 기능을 흡수하면서 얼마나 효율적인 소프트웨어를 지원하느냐가 경쟁의 관건이 될 것이다. 이런 시대를 준비하기 위해서는 소프트웨어 기술, 디지털 TV 나 PDA, 게임기 같은 임베디드 시스템용 실시간 운영체제 (Real-Time Operating System, RTOS) 기술을 확보하고 있어야 한다. 또한, 이와 관련된 미들웨어와 멀티미디어 API (Application Programming Interface)를 개발해 다른 분야로까지 파급 효과를 가질 수 있다. 현재 MPEG-1, MPEG-2, MPEG-4 지원 API 와 대화형 TV 기능을 보유한 ITV API, VBI(Vertical Blanking Interval) API 와 함께 정보 가전에서 기본 제공해야 할 웹브라우저, 멀티미디어 전자메일, 자바 애플리케이션을 위한 퍼스널 자바(personal java) 등을 지원하는 통합 기술이 개발되고 있다.

본 작업에서는 이러한 RTOS 환경에서의 멀티미디어 API 개발의 중요성에 부합하여, 실시간 서비스라는 현실적 문제와 응용 범위를 고려하여 낮은 비트율에서의 멀티미디어 서비스 중 음성이나 다른 오디오 신호 성분을 압축하는데 이용될 수 있는 G.723.1 음성 코덱[2] API 를 RTOS 환경

에서 개발하였다.

G.723.1 은 5.3 kbit/s 와 6.3 kbit/s 의 이중 비트율을 갖는 구조로 현재 별정 통신으로 상용화되어 있는 인터넷폰과 이동 통신용 음성 코덱으로 사용되고 있으며, 낮은 전송률에서도 비교적 우수한 음질을 제공하고 있다. 더불어, 최적의 전송 환경을 위해 두 개의 비트율을 사용하기 때문에 다른 음성 코덱 표준에 비해 더욱 응용성이 높다.

본 논문에서는 G.723.1 음성 코덱에 대해 기술한 후, 기본적인 요구 사항을 만족하는 MPEG-4 시스템 구현과 실시간 처리를 위한 VxWorks 환경에서의 구현에 대해 기술한다.

2. G.723.1 음성 코덱

G.723 은 낮은 비트율에서의 멀티미디어 서비스 중 음성이나 다른 오디오 신호 성분을 압축하는데 이용될 수 있는 부호화 방식이다.

이 부호화기는 5.3 kbit/s 와 6.3 kbit/s 의 이중 비트율을 갖는다. 높은 비트율은 더 나은 음질을 제공한다. 낮은 비트율은 적절한 정도의 음질을 제공하며, 시스템 설계자는 부가적인 유용성을 갖는다. 두 비트율은 30 msec 프레임 경계에서 전환될 수 있다.

이 부호화기는 제한된 복잡성을 기반으로 두 비트율에서 최상의 음질을 갖도록 음성 신호를 최적화 시킨다. 음악이나 다른 오디오 신호들은 음성 신호만큼 충실히 따르지는 않지만, 이 부호화기를 이용하여 압축하거나 압축을 풀 수 있다.

이 부호화기는 30 msec 프레임 단위로 음성과 다른 오디오 신호들을 부호화한다. 더욱이, 예약된 7.5 msec 를 포함하며, 결과적으로 37.5 msec 의 전체적인 알고리즘 지연을 갖는다. 이 부호화기의 구현과 동작에 있어서 부가적인 지연은 부호화기와 복호화기에서 실제적으로 신호가 처리되는 시간, 통신 링크에서의 전송 시간 및 다중화 프로토콜에서의 부가적인 버퍼 지연에 기인한다.

2.1 부호화 원리

이 부호화기의 입력은 아날로그 입력을 전화선 대역폭 필터링(ITU Rec. G.712)을 수행하고, 8000 Hz 로 표본화 한 뒤, 16 bit 선형 PCM 으로 변환되어 얻어진 디지털 신호이다. 복호화기의 출력은 비슷한 방법에 의해 아날로그 신호로 변환되어야 한다. 64 kbit/s PCM 데이터에 대한 ITU Rec. G.711 과 같이 다른 입력/출력 특성을 갖는 신호들은 부호화 전에 16 bit 선형 PCM 신호로 변환되어야 하며, 복호화 후에 16 bit 선형 PCM 신호로부터 적절한 형태의 신호로 변환해야 한다.

부호화기는 선형 예측 분석/합성 부호화의 원리에 기반하고 있으며, 지각적으로 가중치가 적용된 오류 신호를 최소화한다. 부호화기는 각 240 표본들의 블록(프레임), 즉, 8 kHz 표본화율에서 30 msec 로 동작한다. 각 블록은 DC 성분을 제거하기 위해 고주파 통과 필터링되며, 각 60 표본들로 구성된 4 개의 부프레임(subframe)으로 분리된다. 각 부프레임에 대하여, 처리되지 않은 입력 신호를 사용하여 10 차 선형 부호화기(Linear Prediction Coder, LPC)가 계산된다. 마지막 부프레임에 대한 LPC 필터는 예측 분할 벡터 양자화(Predictive Split Vector Quantization, PSVQ)된다. 양자화되지 않은 LPC 계수들은 지각적 가중치 필터을 구성하는데 이용되며, 이는 전체 프레임을 필터링하고 지각적으로 가중치가 적용된 음성 신호를 얻는데 이용된다.

두 부프레임(120 표본들)에 대해, 개방 루프 피치 주기는 가중치가 적용된 음성 신호를 이용하여 계산한다. 이 피치 추정(pitch estimation)은 120 표본들로 구성된 블록에 수행된다. 피치 주기는 18 에서 142 표본까지의 범위 내에서 탐색된다. 지금부터 음성 신호는 부프레임을 기본으로 60 표본을 단위로 처리된다.

위에서 추정된 피치 주기를 이용하여 조화 잡음 형태 필터(harmonic noise shaping filter)가 만들어진다. LPC 합성 필터, 포만트(formant) 지각 가중치 필터, 조화 잡음 형태 필터의 조합이 임펄스 응답을 만들기 위해 이용된다. 그런 후, 임펄스 응답은 앞으로의 계산을 위해 이용된다.

피치 주기 추정, 개방 루프 피치 주기, 임펄스 응답을 이용하여 폐루프 피치 예측기가 계산된다. 5 차 피치 예측기가 이용된다. 피치 주기는 개방 루프 피치 추정 근방의 작은 차이값으로써 계산된다. 그런 후, 초기 목표 벡터에서 피치 예측기에 의한 값을 뺀다. 피치 주기와 차이값이 복호화기에 전송된다.

마지막으로, 여기(excitation)의 비주기적인 요소들이 근사화된다. 높은 비트율에 대해, MP-MLQ (Multi-Pulse Maximum Likelihood Quatization) 여기가 이용되며, 낮은 비트율에 대해, ACELP(Algebraic Codebook Excitation)이 이용된다. 그림 2 는 부호화기의 블록도를 나타내고 있다.

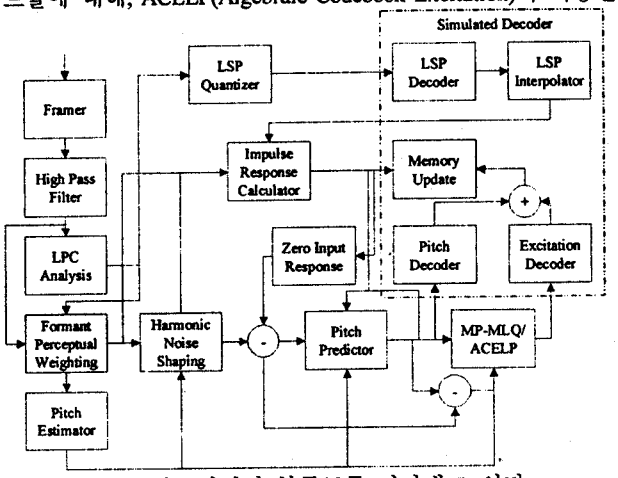


그림 2. G.723.1 부호화기의 블록도

2.2 복호화 원리

마찬가지로, 복호화기 동작은 프레임을 기본으로 한 프레임씩 수행된다. 양자화된 LPC 인덱스들이 복호화된 후, 복호화기는 LPC 합성 필터를 만든다. 매 부프레임에 대해, 적응 코드북 여기(adaptive codebook exitation)와 고정 코드북 여기(fixed codebook excitation)가 복호화되어 합성 필터의 입력이 된다. 적응 후필터(postfilter)는 포만트 후필터와 순반향-역방향 피치 후필터로 구성된다. 여기 신호는 피치 후필터의 입력이 되고, 차례로 합성 필터의 입력이 된다. 그 출력은 포만트 후필터의 입력이 된다. 이득 스케일링 장치는 포만트 후필터의 입력 레벨을 유지시킨다. 그림 3 은 복호화기의 블록도를 나타내고 있다.

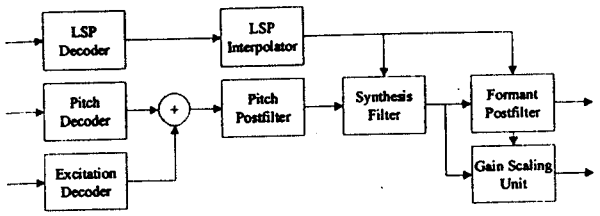


그림 3. G.723.1 복호화기의 블록도

3. MPEG-4 시스템에서의 구현

MPEG-4 표준화 활동에서는 일반 전화망이나 이동 통신망과 같이 전송 주파수 대역이 아주 작은 채널에서도 디지털 비디오/오디오 통신 서비스를 제공할 수 있도록 초당 64 kbps 이하의 매우 낮은 비트율을 갖는 부호화 방식을 개발하고 있다. 그 동안 MPEG-4 표준 방식의 응용 및 접근 방법에 대해 제안된 여러 가지 의견들을 검토하고 수렴하면서, MPEG-4 시스템[3]에서 갖추어야 할 주요 기능들을 정의하였다. 이러한 기능들은 MPEG-4 표준의 근본 취지를 뒷받침하며, 기존의 표준 방식들이 지원할 수 없는 내용들을 포함하고 있다. 이를 위해 본 논문에서는 실시간 멀티미디어의 응용으로 원격 강의를 목표로 기본적인 MPEG-4의 요구 사항을 만족시키는 시스템을 구현하였으며 음성 코덱으로 G.723.1 을 사용하였다.

3.1 MPEG-4 시스템 구조

MPEG-4 복호화기는 내부에 비디오, 오디오, 텍스트 등을 복호하는 개별 복호화기를 내장하고 있다. 네트웍을 통하여 MPEG-4 복호화기, 즉 클라이언트로 전송되어온 TransMux 비트열 속의 다양한 형태의 비디오, 오디오, 텍스트 등의 멀티미디어 자료를 기초비트열(Elementary Stream, ES)들로 구분하여 각각에 적합한 복호화기로 분배하여 원래의 비디오, 오디오, 텍스트 등으로 복원하는 역할을 한다. 그림 4 는 복호기로 전송할 TransMux 비트열을 만드는데 사용되는 FlexMux 비트열을 만드는 MPEG-4 부호화기의 동작을 개략적으로 보여주고 있다.

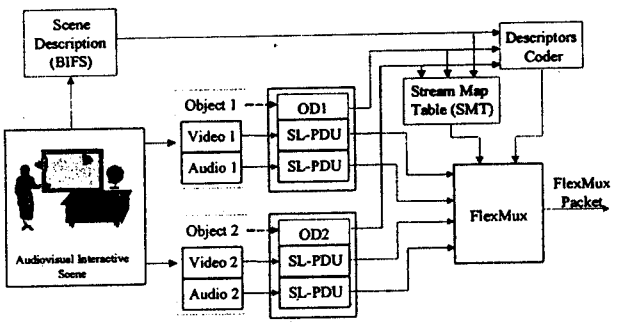


그림 4. MPEG-4 부호화기의 모델

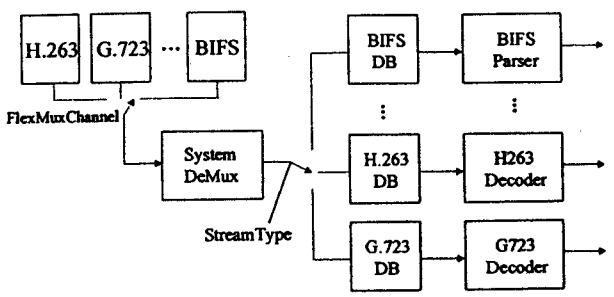


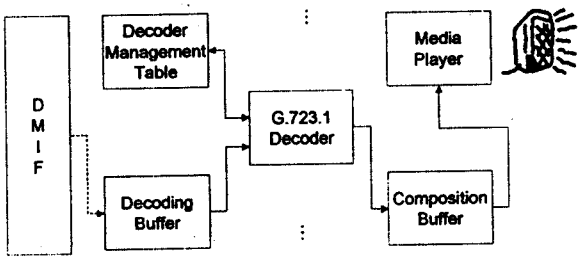
그림 5. MPEG-4 복호화기 구현

MPEG-4 복호화기는 그림 5 와 같이 구현되었다. H.263 혹은 G.723.1 등의 복호화기에서 필요로 하는 타이밍 정보와 AU(Access Unit)를 임시로 저장하기 위해서는 system DeMux 블록과 Decoder 블록사이에 디코딩 버퍼(Decoding Buffer, DB)가 필요하다. 이 버퍼에 들어 있는 내용을 입력받아 복호화기는 복호화를 시작하고, 그 결과를 CB(Composition Buffer)에 저장한다.

3.2 매체 재생기와의 인터페이스

그림 6 은 MPEG-4 복호화기 내부의 G.723.1 음성 코덱과 매체 재생기와의 인터페이스를 보여주고 있다. 복호화기를 통해 복호화된 결과는 CB 에 저장되고 매체 재생기를 통해 스피커로 출력이 된다.

그림 6. 매체 재생기와의 인터페이스



4. VxWorks 환경에서의 구현

4.1 토네이도와 VxWorks

윈도우 98 이나 NT 에서 보장할 수 없는 실시간성을 부여한 RTOS 는 운영체제라고는 하지만 POSIX 레벨의 함수와 gcc 급의 컴파일러, 적당한 소스 레벨의 디버거가 전부일 정도로 간단하다. 현재 국내에서 많이 쓰이고 있는 RTOS 로는 마이크로텍의 VRTX, ISI 의 pSOS, 윈드리버(WindRiver)의 VxWorks 등이 있고, 이밖에 QSSL 의 QNX 와 같은 운영체제가 쓰이고 있다. 화성 착륙선인 패스파이더의 운영체제로 쓰여져 유명해진 VxWorks 의 경우 가장 고가이긴 하지만 다양한 개발 환경을 제공하고 있다[4].

임베디드 시스템의 개발 환경이 일반 데스크탑과 가장 다른 점은 교차 개발 환경(cross development enviroment)이라는 것이다. 즉, 코딩과 컴파일은 개발 호스트에서 이뤄지고, 실제 코드의 실행은 타겟 시스템에서 이뤄진다. 또 디버깅은 호스트와 타겟 환경을 모두 이용해 작업하게 된다. 따라서 이런 환경을 위한 개발 도구는 분명 일반 데스크탑과는 다른 요구 사항이 있게 마련이며, 편리한 디버깅 도구를 얼마나 많이 갖고 있느냐에 따라 개발 환경의 좋고 나쁨이 가려진다.

윈드리버의 토네이도는 이러한 교차 개발 환경을 지원하는 통합 개발 환경이다. 타겟 시스템의 상태를 일목요연하게 보여주는 타겟 브라우저와 다른 툴과 연동되는 디버거, 그리고 타겟 시스템에 대한 명령 해석기인 WindSh 등을 포함하고 있다. 코드 실행이나 디버깅 또는 타겟 시스템의 메모리 등을 C 언어 구문을 이용해 자유롭게 확인할 수 있으며, 필요에 따라 타겟 시스템에 내장시켜 현장 테

스트에 이용할 수도 있다.

VxWorks 는 지원하는 수십 종의 CPU 구조에 대해서 최적화된, 그러면서 사용자에게는 일관된 인터페이스를 제공하는 실시간 운영체제다. 5KB 마이크로 커널을 핵심으로 100 개의 모듈을 이용해 원하는 대로 운영체제를 구성할 수 있으며, 256 단계의 우선 순위를 지원하는 선점형 멀티태스킹과 중첩된 인터럽트 핸들링 환경을 제공한다.

4.2 VxWorks 환경에서의 G.723.1 음성 코덱 구현

G.723.1 음성 코덱 API 를 개발하기 위해 그림 7 과 같은 개발환경을 구성하였다. Tornado 시스템의 OS 는 윈도우 NT 4.0 을 사용하였으며, Target 시스템의 OS 는 VxWorks 를 사용하였다.

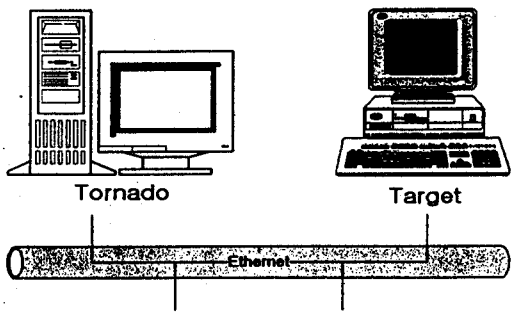


그림 7. G.723.1 음성 코덱 개발 환경

VxWorks 환경에 적절한 API 를 개발하기 이전에 우선 SUN OS 의 환경에서 gcc 를 이용하여 API 형태의 음성 코덱을 개발하였으며, 다음으로 VxWorks OS 로 프로그램 소스를 포팅하였다. 아울러 테스트용 비트열 제작을 위해서 SUN OS 의 플랫폼 위에서 G.723.1 음성 코덱을 이용하여 제작하였다.

개발된 오디오 복호화기가 제대로 동작하는지를 확인하기 위해 G.723.1 비트열을 네트웍 드라이브 방식을 사용하여 타겟 시스템으로 로드 한 후에, 복호화하여 그 결과를 다시 네트웍 드라이브에 저장하였다. 이 후, 복호화된 신호는 PCM 재생기를 이용하여 확인하였다. 최종적으로, 네트웍 드라이브 방식을 사용하지 않고 타겟 시스템의 버퍼를 이용하여 G.723.1 비트열을 로드한 후, 복호화하여 스피커를 통해 그 결과를 확인하였다.

4.3 구현된 APIs

VxWorks 환경에 구현된 G.723.1 음성 코덱의 API 는 다음과 같다.

1) LSP 디코더
   - Lsp_Inq(): LSP 주파수들의 역벡터양자화를 수행한다.
2) LSP 보간
   - Lsp_Int(): 한 프레임에 대해 양자화된 LPC 계수들을 계산한다.
   - LsptoA(): 한 서브프레임에 대해 LSP 주파수를 LSP 계수로 변환한다.
3) 피치 정보의 디코딩
   - Get_Rez(): 이전 여진 벡터들로부터 디스플레이된 부분을 받는다.
   - Decod_Acbk(): 이전 여진 벡터들로부터 적응 코드북을 여진을 계산한다.
4) 여진 디코더
   - Fcbk_Unpk(): 두 비트율에 대해 고정 코드북 여진을 디코딩한다.
5) 피치 후필터
   - Comp_Lpf(): 피치 후필터 파라메터를 계산한다.
   - Find_F(): 순방향 상호상관(crosscorrelation)을 최대화하여 최적화된 피치 후필터 순방향 래그 (lag)를 계산한다.
   - Find_B(): 역방향 상호상관을 최대화하여 최적화된 피치 후필터 역방향 래그를 계산한다.
   - Get_Ind(): 피치 후필터의 이득을 계산한다.
   - Filt_Lpf(): 각 서브프레임에 대해 피치 후필터를 적용한다.
6) LPC 합성 필터
   - Synt(): 한 서브프레임에 대해 디코더 합성 필터를 구현한다.
7) 포만트 후필터
   - Spf(): 한 서브프레임에 대해 포만트 후필터를 구현한다.
   - Comp_En(): 한 서브프레임 벡터의 에너지를 계산한다.
8) 이득 스케일링
   - Scale(): 후필터 출력 신호의 이득을 조절한다.
9) 프레임 보간
   - Comp_Info(): 유성음과 무성음을 분류한다.
   - Regen(): 프레임 분류에 따라 residual 보간을 수행한다.
10) 디코더 초기화
   - Init_Decod(): 디코더에 대해 0 값이 아닌 변수들을 초기화한다.

5. 결론

본 논문에서는 실시간 멀티미디어 시스템을 위한 G.723.1 음성 코덱의 구현에 대해 기술하였다. 개발된 API 는 VxWorks 환경에서 실시간 복호화가 가능했으며, 원격강의를 목표로 한 MPEG-4 시스템에서의 음성 코덱 역할을 하는데도 부족함이 없었다. 현재 Q-PLUS(Q+)라는 이름으로 개발된 한국형 RTOS 환경에 포팅이 진행 중에 있으며, 정보 가전에 초점을 맞춘 제품으로 그 응용이 기대되고 있다.

실시간 멀티미디어 시스템을 위한 멀티미디어 API 개발은 기존의 단방향 통신의 일방적인 정보 전달의 차원을 벗어나, 수신자의 요구와 선택에 따라 필요한 정보를 원하는 시간에 제공할 수 있는 양방향 서비스(interactive service)를 실현시킬 수 있으며 그 파급효과를 노릴 수 있으므로 앞으로 보다 나은 성능의 API 가 개발될 수 있도록 연구가 지속되어야 한다.

감사의 글

본 연구는 한국전자통신연구원(ETRI)의 조립식 실시간 OS 기술개발 과제, 광주과학기술원(K-JIST) 초고속광네트워크연구센터(UFON)를 통한 한국과학재단 우수연구센터(ERC)와 교육부 두뇌한국 21(BK21) 정보기술사업단의 지원에 의한 것입니다.

참고 문헌

[1] http://www.daou.co.kr/product/real
[2] ITU-T Recommendation G.723, Dual Rate Speech Coder For Multimedia Communications Transmitting At 5.3 & 6.3 kbits/s, 1995.
[3] ISO/IEC 14496-1:1999, Information Technology – Coding of audio-visual objects - Part 1: Systems.
[4] http://www.windriver.com