

장 현순, 주 창희, 최 중수  
중앙 대학교, 전자 공학과

CODEC Design of Hybrid Interframe & BTVQ

Chang Hyun Soon, Joo Chang Hee, Choi Jong Soo  
Dept. of Electronic Eng, Chungang University

ABSTRACT

This paper describes CODEC based on hybrid interframe and BTVQ including CRC(Conditional Replenishment Coding) system and RLC(Run Length Coding) method.

Based upon the algorithm described in this paper, a practical CODEC has been designed for video conference and video phone use at 54Kbps using an ADSP-2100 as a CPU.

1. 서론

기존의 디지털 전송로를 통해 영상정보를 전송하고자 할 경우 방대한 정보량으로 인한 대역폭의 증가로 인해 문제가 생기는 바, 이를 해결하기 위한 방안으로서는 기존의 전송로를 광대역 선로인 광섬유로 대체하는가 아니면 기존의 선로를 이용하되 영상정보의 중복성을 충실도를 잃지 않는 범위 내에서 억제시킴으로써 전송률을 낮추는 방법이 있다.

현재로서는 경제성의 측면에서 후자의 방법을 택하는 것이 바람직하다. 1970년대 이후 여러 영상 압축 기술이 개발되었는 바, 크게 네 그룹으로 분류할 수 있다. 첫째는 변환코딩 방법 이고 둘째는 예측 부호화법 셋째는 벡터 양자화법이며 넷째는 제 2세대 코딩법을 들 수 있다.

본 논문에서는 준최적이지만 탐색량이 적은 BTVQ(Binary Tree-searched Vector Quantizer)를 프레임간 코딩기법으로서 움직임이 비교적

적은 영상의 경우에 매우 유용한 조건 보충 코딩 (CRC;Conditional Replenishment Coding)및 런 렉스 부호화 법을 병합함으로써 비디오 전화나 비디오 회의 신호를 54Kbps로 전송할 수 있는 코덱을 설계하였다.

본 시스템에서 한 프레임은 128\*128\*8 비트로 구성되어 있으며 초당 30 프레임을 전송하게 된다. 그리고 처리하고자 하는 영상의 프레임간의 움직임은 전 프레임 대비 30% 이내로 가정하였다.

2. 혼합 부호화 시스템

프레임간 및 양갈래 탐색 벡터 양자화기를 혼합한 코덱의 기본 구성도는 그림 1 에 도시된 바와 같다.

우선 전송단에서 프레임간 차신호는 전 프레임 버퍼 메모리와 비교되어 계산되고, 계산된 결과값이 움직임 검출기(MAD;Moving Area Detector)로 보내진다. 이어 움직임 검출기는 벡터 양자화할 대상인지의 여부를 주어진 문턱치를 이용하여 결정하게 된다. 만일 움직임이 있었다고 판단되면 입력 데이터는 BTVQ로 입력되어 출력으로서 BTVQ의 인덱스가 FIFO로 보내진다. 그리고 만일 입력 벡터가 비 움직임 영역으로 판단되면 인덱스 "0"이 RLC에 보내져 런렉스 된 신호가 FIFO에 입력되어 전송되게 된다.

수신단에서는 위에서 설명한 역순으로

원신호가 복구 되게 된다.

일반적인 VQ는 스칼라 양자화에 비해 훨씬 높은 압축률을 얻을 수 있을 뿐만 아니라 VQ 중에서도 가장 좋은 화질을 얻을 수 있고 복호기의 동작이 간단하다는 장점이 있으나 몇가지 단점이 있다. 첫째 부호책에 저장되어 있는 대표 벡터들 중에서 입력벡터와 왜곡이 가장 작은 벡터를 찾는데 많은 계산량이 필요하다. 둘째 부호책을 저장하는데 많은 기억용량이 필요하다. 셋째 블럭화 현상이 나타난다. 넷째 외부 트레이닝 시퀀스 영상에 대해 화질이 저하될 수 있다.

그러나 본 논문에서 제안한 프레임간/BTVQ는 기억용량이 증가되는 점외에는 상기한 문제점을 해결할 수 있을 뿐만 아니라 비록 준 최적이지만 탐색시간을 훨씬 줄이면서 안정성과 블럭화 현상이 개선 되게 된다.

BTVQ코드북의 구조는 그림 2에 도시된 바와 같다.

본 논문에서 코드어의 수는 256개로 잡았으며, 코드어의 확장은 비트레이트에 크게 영향을 미치지 않는다. 코드북을 설계하기 위해 스플리팅 방법이 사용되었으며 파라메타는 시행 착오를 거쳐 문턱치로서 .001 델타값으로서 1.0을 잡았다.

### 3. ADSP-2100

본 하드웨어 구현에 이용된 ADSP-2100의 중요한 특징은 다음과 같다.

-프로그램 메모리(32k \* 24 bit)와 데이터 메모리 (16k \* 16 bit)를 분리해서 갖으므로 동시에 두 메모리를 액세스 할수 있으며 프로그램 메모리는 데이터와 프로그램저장의 두가지 목적으로 사용될수 있고 데이터 메모리는 데이터만을 저장 한다.

-한 명령 사이클로 연산및 프로그램 메모리와 데이터 메모리를 동시에 액세스 할수있다.

-4개의 외부 인터럽트가 있고 각각 레벨 또는 에지(edge) 인터럽트 모드로 프로그램에 의해 조정할수있다.

### 4. 하드웨어 구현

4-1. 송신단 그림3는 코덱의 전송부를 나타낸다. 전송부는 3부분으로 나누어진다. 제 1부분은 송신부의 연산수행에 있어서 중심이 되는 부분으로서 제 3부분으로부터 영상 데이터를 받아 인덱스 넘버 찾기 및 RLC(run length code) 계산을 수행하는 소오스(source) 코딩을 행한다. 소오스 코딩한 결과는 FIFO (first-in first-out) 메모리에 보내진다.

제 2부분은 FIFO 메모리로부터 소오스 코딩한 데이터를 받아 채널(channel) 코딩을 수행하는 동시에 코덱의 수신단과 데이터 전송을 수행한다. 제 3부분은 카메라로부터 아날로그-디지털 변환된 8 비트 영상 데이터를 영상메모리에 저장 한다. 영상 메모리는 2부분으로 나누어 지는데 한부분이 디지털 변환된 영상 데이터를 카메라로부터 받을때 다른 한부분은 ADSP-1의 메모리 부분으로 되어 연산에 이용되고 한 프레임의 연산이 끝났을때는 그반대가 된다.

다음은 제 1부분과 제 2부분이 수행하는 소오스 코딩 및 채널 코딩에 대한 설명이다.

#### 4-1-1. 소오스 코딩

ADSP-2100의 데이터 메모리 용량은 위에서 언급한 바와같이 데이터 메모리 16k 바이트 프로그램 메모리 32k 바이트 인데 제 1부분에 필요한 데이터 메모리의 용량은 16k\*2(영상 메모리) + 16k(프레임 메모리) + 32k(코드북) = 80k 바이트 인바 ADSP-2100의 메모리 용량문제를 해결 하기 위해 그림3 에서 보는 바와같이 Page\_port 및 SW\_port를 이용해서 현재 연산에 사용되어지는 메모리만을 ADSP-2100의 메모리 영역에 불입으로하여 메모리를 확장 하였다. 프레임 버퍼에는 이전 영상의 데이터 값이 저장되어 있고 book1에는 2-255 까지 book2에는 256-511까지의 인덱스

번호에 해당하는 내용이 저장되어 있다. 인덱스 번호 또는 RLC 를 찾기위한 알고리즘은 다음과 같다. 영상 메모리에서 영상 버퍼를 뎀으로서 여러 프레임을 생성 한다. 여러 프레임은 영상 메모리에 저장된다. 여러 프레임의 해당 블록에 대해서 각각의 요소(element)의 절대값을 더해서 그 결과가 미리정해진 임계치 보다 크면 움직임이 있는 부분으로 간주해 인덱스 번호를 찾고 그 반대면 RLC를 1 증가 시킨다. 만약 RLC 가 15이면 FIFO로 RLC를 보내고 RLC는 0 이된다. 인덱스 번호를 찾을때 RLC가 존재하면(0 이 아니면) FIFO로 RLC를 네 보낸후 인덱스 번호를 찾는다. 인덱스 번호를 찾았으면 인덱스 번호를 FIFO로 보낸다.

한 블록 계산에 할당된 사이클은 ADSP-2100이 초당 12.5M 사이클을 수행할수 있고 초당 30프레임, 한 프레임당 256블록 을 처리해야 함으로  $12.5M / (30 * 256) = 1600$  사이클이 된다. 인덱스 번호를 찾는데 필요한 사이클 수는 약 3776 사이클 RLC 계산에 필요한 사이클 수는 약 256 사이클이다. 따라서 RLC계산 : 인덱스 번호찾기의 비율 약 1.62 : 1로 하면, 즉 움직임이 있는 블록이 38% 미만이라 가정하면 다음 프레임의 영상 데이터가 들어오기전에 현재 프레임의 계산을 완료할수있다.(본 코덱 설계에서는 움직임이 있는 부분이 30% 라고 가정하였다.)

인덱스 번호 전송에 9비트 RLC 전송에 5비트를 할당 하면 한 프레임당 데이터 비트수는 1682 비트가되고 1초당(30 프레임당) 전송 데이터 비트수는 50,460 비트가 된다.이 전송 비트 레이트(rate)를 일정하게 유지하기 위해 임계치를 가변으로 하여 한 프레임의 계산이 끝날 때마다 움직임이 있는 블록이 적으면 임계치를 감소시키고 움직임이 있는 블록이 많으면 임계치를 증가 시킨다.

#### 4-1-2. 채널 코딩

제 3부분은 데이터의 전송및 채널 코딩을 담당 한다. 수신측에서 인덱스 번호와 RLC를 구분하기 위해 인덱스 번호는 9비트로 구성하고 인덱스 번호의 최상위비트(MSB)는 1로 한다. RLC는 5비트로 구성하고 최상위비트는 0로한다. 전송방식은 Sliding window protocol과 HDLC(high-level data link control)을 약간 변형한 방식을 사용하였다. 그림4는 사용된 HDLC의 frame 구조이다.

데이터 전송 frame 내의 데이터는 400 비트씩으로 하고 여러 체크(error check)를 위해 CRC 방식을 사용한다. 데이터 전송 속도는 HDLC frame 구조에 의해 8비트(flag) + 8비트(control code) + 16비트(CRC check) + 8비트(flag) = 40비트가 400비트마다 더해짐으로 약 56k bps(bit per second)가 되고, 데이터 1-비트를 전송하는동안 ADSP2는 약 220 번의 명령을 수행할수 있다. 따라서 220 번의 사이클 마다 타이머에서 인터럽트를 걸고 인터럽트 루틴(interrupt routine)에서 F/F(flip flop)에 전송할 비트를 보낸다. 데이터의 수신 역시 타이머에 의해 인터럽트를 걸어 1-비트당 10번의 샘플을 함으로서 데이터를 받는다. 9비트 마다 최소한 1비트의 변화가 있으므로 비트가 변할때마다 다시 동기를 맞춤으로서 데이터를 정확히 받을수 있다. 코덱의 수신단은 전송된 데이터를 받아 에러가 발생 했으면 에러가 발생한 frame의번호를 송신단에 보내고 이때 송신단은 에러가 발생한 frame을 재송신한다.

#### 4-2. 수신부

그림5는 코덱의 수신단 블록도이다. 송신부와외 통신은 ADSP4가 담당하며 송신단의 ADSP2와 동일한 역할을 수행한다. 역시 수신시 비트당 샘플 레이트(rate)는 8번이 고 전송은 220 사이클 마다 인터럽트 루틴에

의해 수행한다. 데이터의 한 frame을 받았을 때는 일단 램(RAM)에 저장했다가 에러 체크를 하고 에러가 발생했을 경우 에러가 발생한 frame의 넘버를 송신측에 보내고 에러가 발생한 frame이 다시 전송 되기를 기다린다. 에러가 발생하지 않았을 경우에는 송신단에 ACK(acknowledgment) 신호를 보내고 그 frame을 FIFO 메모리에 보낸다.

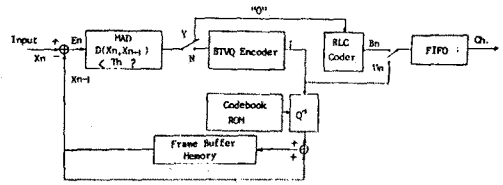
ADSP3는 FIFO메모리로부터 데이터를 받아 영상 버퍼를 리프레쉬(refresh) 하고 주사선의 귀선시간을 이용하여 프레임 버퍼의 내용을 영상 메모리에보냄으로서 모니터상에 수신단의 프레임메모리 즉 송신단의 프레임메모리가 나타난다.

5. 결 론

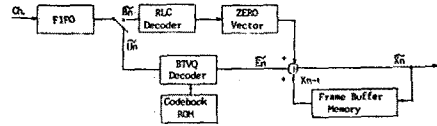
비디오 전화나 영상회의 신호를 54Kbps로 전송할 수 있는 프레임간 및 양갈래 탐색 벡터 양자화기에 바탕을 둔 코덱의 설계에 대해 서술하였다. 프레임간 조건보충 코딩과 양갈래 탐색 벡터 양자화 및 다른 기술들을 병합함으로써 정보량을 실용적인 수준까지 낮출 수 있다. 본 시스템은 비교적 움직임이 적은 영상에 대해 적합하며(30% 이내) 외부 트레이닝 시퀀스에 대해서도 안정하다. 그러나 장면의 변화에 대해서는 몇개의 프레임이 지나야 비로소 영상의 질이 복구 되게 된다. 보다 개선된 시스템의 구축에 대해서는 차후에 발표하기로 한다.

REFERENCE

1. ADSP-2100 User's Manual & Data book.
2. R.M.Gray, "Vector Quantization," IEEE ASSP Mag. Vol. 1, pp. 4-29, Apr. 1984.
3. P.A.Wintz, " Transform Picture Coding," Proc. IEEE, Vol.60, pp.809-819, July 1972.
4. Y.Linde, A.Buzo, R.M.Gray, "An algorithm for Vector Quantizer Design," IEEE Trans. commun., Vol. COM-28, pp.84-95, Jan. 1980.
5. W. H. Chen & W.K. Pratt, "Scene Adaptive Coder," IEEE Trans. Commun. Vol. COM-32, pp.225-232, Mar 1984.
6. 김 낙언, 최 종수, "프레임간 및 양갈래 탐색 벡터 양자화기를 혼합한 영상 부호화 시스템", 한국통신학회 1987년도 춘계 학술 발표논문집, pp.31-34.



(a) Encoder



(b) Decoder

그림 1. 혼합 코딩 시스템의 구성도

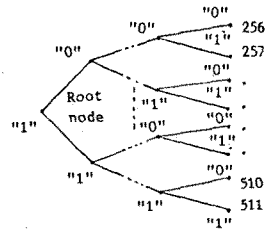


그림 2. BTVQ 코드북의 구조

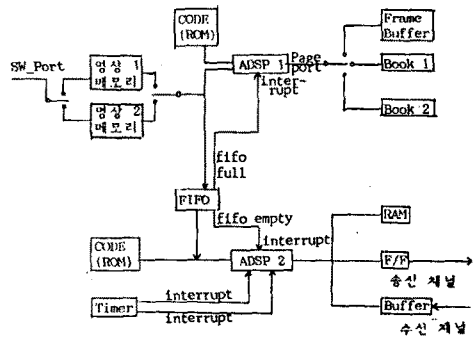


그림 3 Codec 송신부

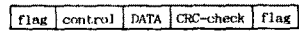


그림 4 HDLC frame 구조

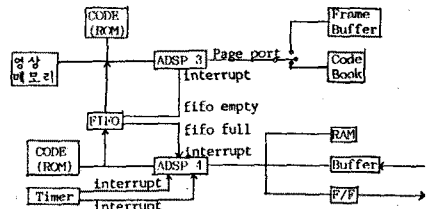


그림 5 Codec 수신부