

論文 90-27-3-4

이동 보상과 분류 벡터 양자화기를 이용한 영상 부호화에 관한 연구(Ⅱ : 하드웨어 실현)

(A Study on the Interframe Image Coding Using Motion Compensated and Classified Vector Quantizer (Ⅱ : Hardware Implementation))

全重滿*, 申泰民*, 崔成男*, 朴圭泰*

(Joong Nam Jeon, Tae Min Shin, Sung Nam Choi, and Kyu Tae Park)

要 約

본 논문은 MC-CVQ(motion compensated and classified vector quantization) 알고리즘의 하드웨어 실현에 관한 것으로, 128×128화소로 구성된 흑백영상을 64Kbps채널로 1초에 약 10장의 화면을 전송할 수 있는 화면간 부호화장치의 제작에 대하여 설명하였다.

위의 조건하에서 부호화를 수행하기 위하여, 시스템을 MC부, CVQ부, 복호화부로 구분하여 마이크로프로그램 제어에 의한 멀티프로세서 구조로 구성하였다. 그리고 MC부와 CVQ부에서 최소 거리 오차를 효율적으로 계산하기 위하여 연산부에는 3-단 파이프라인 구조를 채택하였다.

시스템 제작 후 성능을 평가한 결과, 본 시스템의 화면전송율은 영상신호의 상대적 이동량에 따라 1초에 6-15장 정도임을 확인하였다.

Abstract

This paper describes a hardware implementation of the interframe monochrome video CODEC using a MC-CVQ(Motion Compensated and Classified Vector Quantizer) algorithm. The specifications of this CODEC are (1) the resolution of image is 128x128 pixels, and (2) the transmission rates are about 10 frames/sec at the 64Kbps channel.

In order to design the CODEC under these conditions, it is implemented by a multiprocessor system composed of MC unit, CVQ unit and decoder unit, which are controlled by microprogramming technique. And the 3-stage pipelined ALU (Arithmetic and Logic Unit) is adopted to calculate the minimum error distance in the MC unit and CVQ unit.

The realized system shows that the transmission rates are 6-15 frames/sec according to the relative motion of the video signal.

I. 서 론

영상통신은 디지털 종합정보통신망의 중요한 일부분을 차지하고 있으며, 특히 64kbps 채널을 사용하

*正會員, 延世大學校 電子工學科
(Dept. of Elec. Eng., Yonsei Univ.)
接受日字 : 1989年 6月 7日

는 고압축 영상부호화에 대한 연구가 진행 중이다.

64kbps로의 영상부호화는 1979년 서독의 Musmann에 의해 흑백영상에 대한 모의실험이 발표된 이후, 1984년 미국의 Widcom사가 화면간 예측과 DCT(discrete cosine transform)를 이용한 복합부호화에 의한 56kbps 코덱의 실시간 부호화 결과를 발표하였다.^[1] 그 후, 미국에서 Pictel의 C-200, CLI의 Rembrant 56, 일본 KDD의 Invite 64, 미쓰비시의 MVC-3000 등이 발표되었다. 이들의 핵심 기술은 이동보상, 배경예측, DCT, VQ(vector quantizer) 등의 복합형태이다.

동 영상을 64kbps로 전송하기 위해서는 TV신호 데이터를 약 1/1500정도로 압축해야 하며, 이로 인하여 시간 및 공간 해상도, 화질 등에 제약이 생기는 것은 피할 수 없다. 그러므로 실제로는 수평, 수직 해상도 및 화면수를 TV신호의 약 1/3-1/5로 저감하고, 이 범위내에서 실용적인 화질을 실현하도록 하고 있다.

본 연구에서는 영상전화기 제작을 목적으로 128×128화소로 구성된 영상을 마이크로프로그래밍, 멀티프로세서 제어, 파이프라인 구조 등의 기술을 사용하여 본논문 I : 이론 및 모의 실험의 알고리즘을 하드웨어로 실현하였다. 그러나 일반적으로 완전한 CODEC(coder/decoder)이 정해진 채널용량에 맞추어 전송하여야할 전송정보의 양에 무관하게 일정 비트율로 전송율을 유지하여야 하나, 본 연구에서는 전송로 부호화(channel coding)를 행하지 않고 정보원 부호화(source coding)만을 연구대상으로 삼았다.

본 연구에서 구현한 시스템은 다음과 같은 4개의 부로 구성되어 있다. 우선 2장의 화면을 연속적으로 입력받는 영상 취득부, 취득한 영상과 이전화면을 이용하여 이동벡터를 추출한 후 이동벡터를 이용하여 이동보상 예측오차(motion compensated frame difference) 신호를 구하고, 이 신호를 근거로 하여 블록부류와 벡터부류를 결정하는 이동보상부, 이동보상부로 부터 여러 종류의 데이터를 큐 메모리를 전하여 전달받아 부호화하고, 이동보상부의 이전화면을 구성하기 위한 분류 벡터양자화부, 마지막으로 이동보상부와 분류 벡터양자화부로 부터 부호화된 정보를 수신하여 원 화상을 재생하는 복호화부로 구성된다.

II. MC-CVQ의 설계

1. 개요

본 연구에서 실현한 부호화장치의 제원을 표 1에

표시하였다. 이동보상시 효율적인 블록매칭 알고리즘은 소프트웨어 모의실험은 DMD(direction of minimum distortion) 방식^[2]으로 행하였으나, 본 하드웨어 실현에서는 TSS(three step search) 방식^[3]을 사용하였다. 소프트웨어 모의실험과 하드웨어 실현에서 다른 알고리즘을 적용한것은, 객관적인 성능을 나타내는 SNR과 엔트로피(entropy)는 두 알고리즘이 유사하나^[4] 제어의 측면에서 TSS 방식이 유리하기 때문이다. 전송율은 종합정보 통신망(ISDN) 1회선 분인 64Kbps 채널 중 음성신호 16Kbps와 제어 신호 8Kbps를 제외한 나머지 40Kbps를 영상신호 전송에 할당하였다.

표 1. 영상 부호화 장치의 제원
Table 1. Specification of the video CODEC.

1 화면	128×128 pixels
MC block size	8×8 pixels
VQ vector size	4×4 pixels
전송율	약 10 frames/sec at 40Kbps

표 1의 부호화 조건에 한 블록의 부호화 허용시간은 $1/(\text{화면수} \times \text{블럭수}) = 390 \mu\text{sec/block}$ 이다. 블록매칭 알고리즘으로 TSS와 오차거리 함수로 MAE(minimum absolute error)를 사용한 경우 소요되는 계산량은 블럭 단위 거리 계산이 25회이고, 각 탐색위치에 대하여 64(8×8)회의 MAE에 절대값 계산이 필요하므로, 비교와 뺄셈연산 각 64회, 덧셈연산 64회를 합한 192회의 연산을 해야 한다. 그러나, 파이프라인 구조를 도입하면, 이 연산을 약 64회의 계산으로 줄일 수 있다. 따라서, 블럭 하나를 이동보상 때 소요되는 계산량은 TSS가 최소 $25 \times 64 = 1600$ 회이고, 한개의 연산은

$$390 \mu\text{sec}/1600 \text{ operations} = 243.75 \text{ nsec/operation}$$

이내에 처리해야 한다. 일반적으로 한 명령 사이클이 여러 클럭으로 구성된 상용 마이크로프로세서로는 이 시간내에 연산처리가 불가능하므로, 마이크로프로그램에 의한 전용처리의 구조가 바람직하다.

그림 1에 시스템 전체의 구성도를 표시하였다. 부호화부는 영상취득장치, 이동보상부, 분류 벡터양자화기로 구성된 영상부호화부와 채널부호화부로 구성되고, 복호부는 채널복호화부와 영상복호화부로 구성된다. 그러나, 채널부호화부에서는 채널실물에 대한 엔트로피 부호화는 고려하지 않았다.

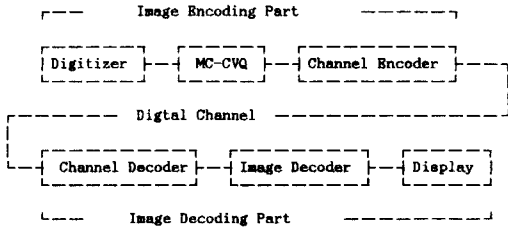


그림 1. 화면간 부호화 장치의 구성
Fig. 1. Block diagram of interframe coder.

본 연구에서는 채널부호화부와 채널복호화부를 병렬버스로 대치하고, 영상부호화부와 영상복호화부를 설계 제작하였다. 영상부호화부는 MC부와 CVQ 부로 분리하여 마이크로프로그래밍 기법에 의해 구성하였다. 2 장에서는 마이크로프로그래밍의 구조, 다중처리기의 구조와 메모리구성, 파이프라인 구조의 연산기 구성 등에 대해서 설명한다.

2. 마이크로프로그램 구조

임의의 시스템을 구현하는 방법은 다음과 같이 3 가지로 구분할 수 있다.

- (1) SSI/MSI hardwired logic
- (2) Microprogrammable bit-slice architecture
- (3) Commercial fixed instruction microprocessor

(1)은 알고리즘 구현이 어렵고, 복잡하며 (3)은 알고리즘 구현은 쉽지만 처리속도가 늦다는 단점이 있다. 계산량이 많고 고속의 처리를 요하는 전용시스템인 경우에는 (2)가 적당하다.

일반적인 마이크로프로그램 구조는 그림 2 와 같이 순서제어기 (sequencer), 마이크로프로그램 메모리, 처리부, 조건 레지스터와 멀티플렉서, 그리고 동기를 위한 파이프라인 레지스터로 구성된다. 마이크로프로그램 메모리는 제어프로그램이 들어가며 순서제어기의 출력에 의해 각 부에 필요한 신호를 발생시킨다.

순서제어기는 처리부의 동작에 따라 조건 분기 (conditional branch) 할 수 있도록, 조건레지스터의 입력 중 하나를 멀티플렉싱하여 사용한다. 실제 시스템 연산을 수행하는 처리부는 마이크로프로그램 메모리로부터의 제어신호에 의해 동작한다.

3. MC-CVQ의 다중처리

MC-CVQ는 그림 3 과 같이 MC부와 CVQ부로 구분하여 설계하였다. MC부는 현화면과 이전화면의 데이터로 이동보상과 분류를 하고, 이동벡터양 만큼 재생메모리에 이동보상을 하며, 부류(class) 정보, 이전화면의 데이터, 이 때의 오차(error) 데이터를 큐

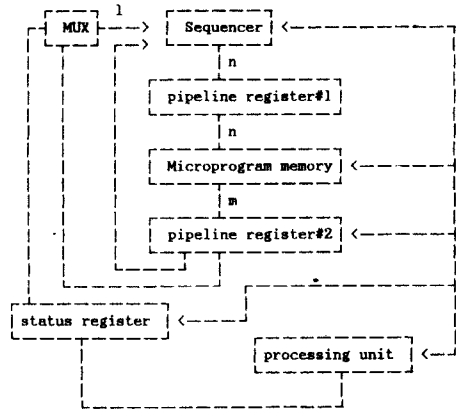


그림 2. 표준 마이크로프로그램 구조
Fig. 2. Standard microprogram architecture.

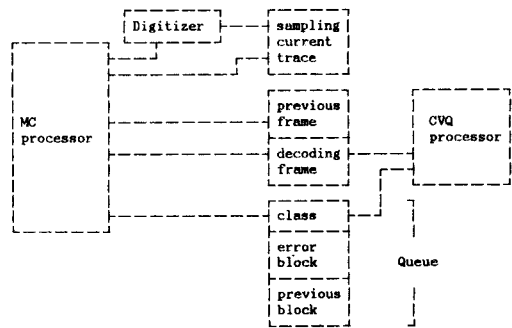


그림 3. MC-CVQ의 구성
Fig. 3. Block diagram of MC-CVQ.

메모리를 통하여 CVQ부에 전달한다. 큐메모리는 FIF0(first-in-first-out)구조를 갖는 소자로서, 본 시스템에서와 같이 두개의 프로세서 동작이 비동기적인 경우 효과적으로 정보를 전달할 수 있다.

CVQ부는 큐메모리로부터 정보를 판단하여, 오차 신호 중 중요화소가 존재하는 벡터만을 양자화하고, 양자화된 벡터는 이전화면의 데이터에 가산하여 재생메모리에 기록한다.

영상 취득부 (digitizer)에는 2 화면분의 메모리를 내장하였으며 그중 하나는 MC부의 명령을 받아 화면을 취득하고, 다른 하나는 바로 이전에 취득한 화면을 이동보상시에 현화면으로 사용되며, 이 동작은 동시에 처리된다.

CVQ부에서는 MC부에서 이동보상시에 필요한 이전화면을 구성하기위해 부호화된 신호로 영상을 복

호하여야 한다. 복호가 완료된 화면은 이동보상부에서 이전화면으로 사용된다. 따라서 2 화면의 화상메모리를 내장하였으며, 이들은 동시 접근 가능해야 한다. 이와 같이, 영상 취득부의 메모리를 page-a0, page-a1이라하고, MC부와 CVQ부의 공통의 메모리를 page-b0, page-b1이라 할 때, 부호화 동작시 메모리 용도는 그림 4와 같다.

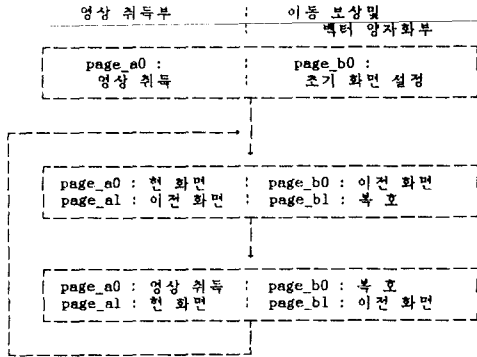


그림 4. 부호화시의 메모리의 사용
Fig. 4. Use of the memory in coding.

4. 연산부의 설계

이동보상에서 이전화면을 부호책 (codebook) 이라고 하면, MC부는 가변벡터 양자화기 (variable VQ) (4)라고 생각할 수 있고, MC부와 CVQ의 연산부는 모두 절대값, 덧셈, 최소거리 비교를 수행하므로 같은 구조로 구성할 수 있으며 구성도는 그림 5에 표시하였다. 그림 5 (a)의 X와 Y는 MC부에서는 각각 현화면의 블럭과 이전화면의 블럭이고, CVQ부에서는 각각 입력벡터와 부호책 벡터이다. 연산부에서 블럭간의 오차거리 $\sum |X-Y|$ 를 계산하여, 최소 오차거리 지표 (pointer)를 구하는 과정은 다음과 같다.

메모리의 어드레스(그림 5 (b)2)는 시스템 클럭(그림 5(b)1)과 동기되어 인가되며, 메모리 데이터(그림 5(b)3)가 출력된다. 이 신호는 2개의 메모리 버퍼 레지스터(MBR)에 저장되며(그림 5(b)4), 각각 X와 Y로 정한다. 논리 회로로 구성된 비교기를 사용하여 한 클럭 기간안에 절대값 계산을 하기 위하여 X-Y를 할 것인가 혹은 Y-X 인가를 결정하고, ALU1에서 $ABS = |X-Y|$ 를 계산하여 레지스터 R1에 저장한다(그림 5(b)5). 이 값은 다음 클럭기간에 합을 저장하는 레지스터 R2의 출력과 ALU2에서 가산된 SUM이 다시 R2로 저장된다(그림 5(b)6). 이때 ALU에서 오버플로우가 발생하면, 절대값의 합이 큰 경

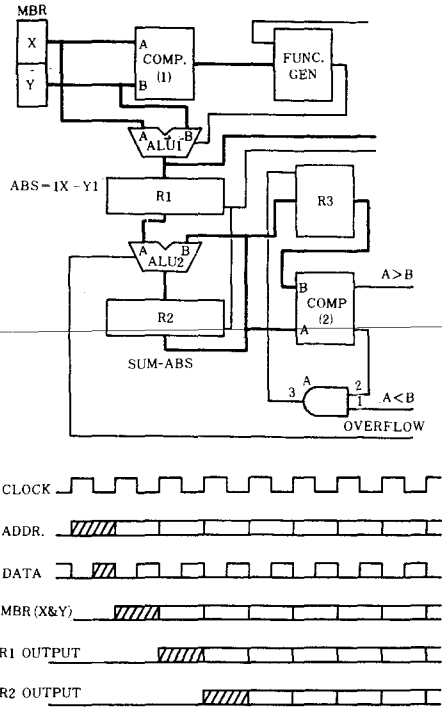


그림 5. 연산부의 구성
Fig. 5. Block diagram of the ALU.

우이므로 이 때는 순서제어기에서 검사하여 현재 계산중인 블럭은 최소거리가 아니라고 판단한다. 마지막으로 R2의 출력은 MAE를 저장하는 레지스터인 R3의 값과 비교기2에서 비교된다. 한 블럭에 대한 절대값 덧셈이 끝나기 이전에 $SUM \geq MAE$ 이면, 최소거리가 아니므로, 이 때도 블럭에 대한 합 계산을 포기하고, 다음 블럭을 계산하도록 한다. 한 블럭에 대한 절대값 계산이 끝났을 때 $SUM < MAE$ 이면 R3의 값을 바꾸어 준다.

연산부에서 절대값 계산을 위하여 입력되는 데이터 폭(width)은 상위 6비트로 제한하였다. 폭을 6비트로 제한한 것은 절대값 계산 이후의 처리시 레지스터와 비교기등의 논리회로가 8비트의 폭내에서 오버플로우없이 계산을 하기 위함이고, 데이터의 폭을 8, 7, 6비트로 정하여 소프트웨어 모의실험을 통하여 성능에 큰 차이가 없음을 보였다. 표 2는 데이터의 폭을 각각 8, 7, 6비트로 정하고, 블럭의 크기를 8 (N=8), 탐색영역을 7 (w=7), 실험영상을 Ellen, Susan, Cronkite에 대하여 PSNR (dB)을 구한 자료이다.

MC부에서 일단 이동벡터가 결정되면 X-Y 연산을 하여 오차신호가 CVQ부로 전달된다. 이때 오차신호

표 2. 데이터 폭에 따른 PSNR 비교

Table 2. PSNR comparison by data width.

	8bits	7bits	6bits
Cronkite	36.06	36.04	35.97
Ellen	38.48	38.49	38.37
Susan	36.99	35.92	35.85

는 임계값과 비교되며, 64단(stage)의 직렬입력 병렬출력 쉬프트 레지스터에 입력시킨 후, 그 출력을 논리조합회로에 입력시켜 블록 및 벡터가 분류하여 CVQ부에 전달한다. 분류를 하기위한 구성도는 그림 6이다. 이와같이 구성되면 별도의 분류시간이 필요하지 않게 된다.

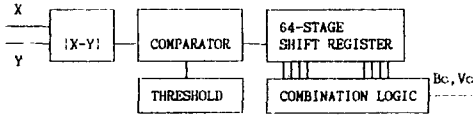


그림 6. 분류기의 구성도

Fig. 6. Block diagram of classifier.

3. MC부의 설계

이동보상부의 구성도를 그림 7에 도시하였다. 이동보상부는 흐름제어부(sequence control part), 주소발생부(address generation part), 이전화면(previous frame)을 저장하기 위한 메모리부(memory part) 그리고 이동량검출(motion vector detection)과 분류를 하기 위한 연산부(ALU part)로 구성된다. 이들 중 흐름제어부, 기억부, 연산부는 2장의 설명과 같다. 여기서는 이동보상을 수행하기 위한 주소발생부와 최소위치 레지스터(minimum pointer register)에 대하여 설명한다.

이동보상부는 현화면(current frame)의 블록과 이전화면(previous fram)의 블록간에 탐색영역(search area)내에서 오차거리가 최소인 위치를 추출해야 하므로, 현화면과 이전화면을 블록단위로 동시에 주소발생을 할 수 있어야 한다. 먼저 현화면에 대한 블록단위 주소발생(block addressing)을 생각해 보자. 블록크기가 8×8화소이고 화면이 128×128로 구성된 경우, 그림 7과 같이 16×16개의 블록으로 구성되므로 H/W를 그림 8과 같이 구성할 수 있다. X와 Y의 블록카운터(block counter)는 4-비트 카운터이고,

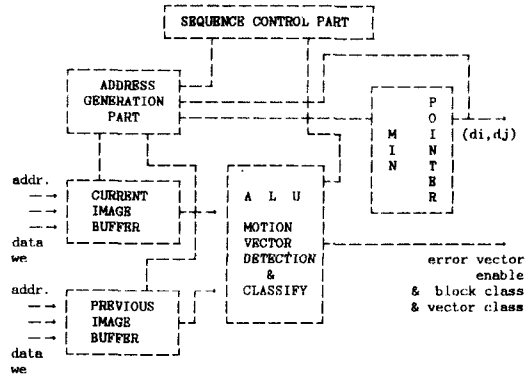


그림 7. 이동 보상부의 구조

Fig. 7. Block diagram of MC processor.

B(0, 0)	B(0, 1)	B(0, 2)		B(0, 15)
B(1, 0)	B(1, 1)	B(1, 2)		B(1, 15)
B(2, 0)	B(2, 1)	B(2, 2)		B(2, 15)
B(15, 0)	B(15, 1)	B(15, 2)		B(15, 15)

그림 8. 블록 단위의 화면 구성

Fig. 8. Frame configuration in block unit.

이 두개는 중속접속(cascade connection)하여 8-비트 카운터로 구성할 수 있다. 카운터 CX, CY는 임의의 값을 입력 가능한(loadable) 7-비트 카운터이고, 블록 주소 BY는 카운터의 load 데이터 중 상위 4-비트에 할당되고 load 데이터 하위 3-비트는 0으로 연결된다. 이렇게 함으로서 블록 주소×8을 간단히 해결할 수 있다. 이 카운터들은 마이크로프로그램에 의하여 제어된다.

이제, 이전화면에 대한 블록단위 주소발생에 대하여 설명한다. 블록 한개에 대한 오차거리가 최소값인 이동벡터(motion vector)를 찾는 배경을 그림10에 표시하였다.

이전 화면에 대한 블록단위 주소발생은 첫째,

- (1) Block counter의 출력 BY, BX
 - (2) 최소 MAE의 pointer dy, dx
 - (3) MC 알고리즘에 의하여 결정되는 di, dj
- 들의 합(BY+dy+di, BX+dx+dj)에 의하여 블록의 좌측 최상점이 결정된다. 둘째, 이동량이 탐색범위를 벗어나는가, 또 탐색점이 화면을 벗어나는가를 검사하여야 한다.

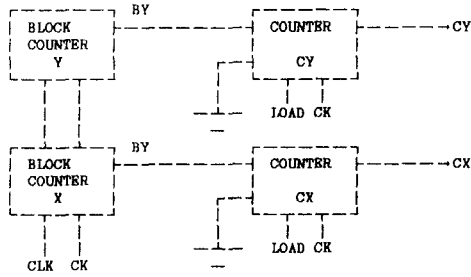


그림 9. Block addressing 구성도
Fig. 9. Block diagram of the block addressing.

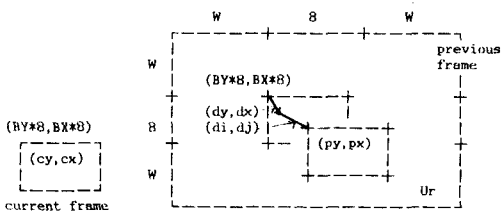


그림 10. 이동벡터 추출
Fig. 10. Motion vector detection.

첫번째 문제는 그림11과 같이 2개의 가산기(adder)에 의하여 해결될 수 있다. 첫번째 가산기는 dy 와 di 를 더하며, 이 계산은 이동량이 음수일 수 있으므로 2의 보수 연산이 된다. 두번째 가산기는 블럭주소 BY 와 첫번째 가산기의 출력을 다음과 같이 비트를 엮갈려 더한다.

$$\begin{array}{r}
 \text{Bit : } 7 \quad 6 \quad 5 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0 \\
 BY : \quad Y3 \quad Y2 \quad Y1 \quad Y0 \quad 0 \quad 0 \quad 0 \\
 dy+di : \quad \quad \quad X3 \quad X2 \quad X1 \quad X0 \\
 \hline
 dy : \quad Z6 \quad Z5 \quad Z4 \quad Z3 \quad Z2 \quad Z1 \quad Z0
 \end{array}$$

여기서 dy 는 2의 보수이다. 따라서 $X3$ 의 값이 1이면 뺄셈이 되고, $X3$ 값이 0이면 덧셈이다. 이 가산기의 출력 dy 를 모든 경우에 대하여 구해 보면 $Z2, Z1, Z0$ 는 $X2, X1, X0$ 와 각각 같게 된다. 따라서 $Y3, Y2, Y1, Y0$ 와 $X3$ 의 조합으로 $Z6, Z5, Z4, Z3$ 를 만들 수 있으며, 이 연산은 32×8 PROM으로 간단히 하드웨어화 할 수 있다.

4. 분류 벡터양자화기의 설계

분류 벡터양자화부의 구성도는 그림12와 같다. 분류 벡터양자화부는 이동보상부에서 처리된 데이터(이동보상 예측오차 벡터 e , 블럭의 부류 Bc , 벡터의

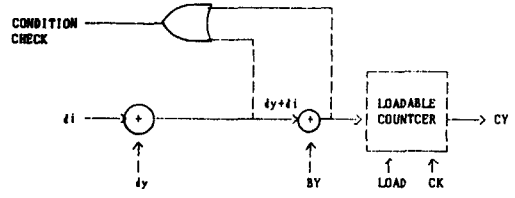


그림 11. CY의 구성
Fig. 11. Block diagram of CY.

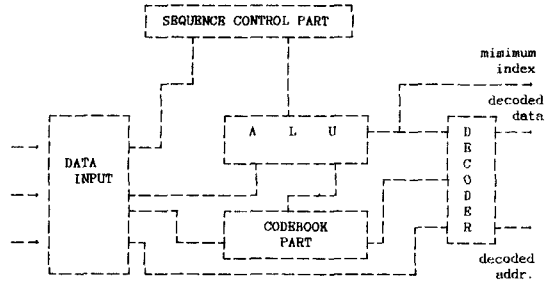


그림 12. 분류 벡터 양자화기 전체 구성도
Fig. 12. Block diagram of the CVQ processor.

부류 Vc , 복호화 주소 (dx, dy) 를 전달받고 저장하기 위한 데이터입력부, 입력벡터에 대해 유사한 벡터들을 저장하고 그 것들 효율적으로 선택하기 위한 부호책부, 입력된 데이터와 부호책내에서 오차거리가 가장 적은 벡터를 찾기 위한 연산부, 그리고 이동보상시에 필요한 이전화면을 만들기 위한 재생부로 구성된다.

데이터 입력부는 이동보상부에서 데이터를 전달받기 위한 큐메모리부와 전달받은 데이터중 이동보상 예측 오차 벡터를 저장하기 위한 스크래치 패드 레지스터(scratch pad register)와 부류정보와 복호화 주소를 저장하기 위한 래치 레지스터(latch register)로 구성되어 있다. 데이터 입력부에서 스크래치 패드 레지스터를 사용한 이유는 레지스터의 출력이 연산부의 파이프라인 구조의 첫 단계와 연결되어 한 파이프라인 사이클에서 데이터 인출 및 산술연산(ADD or SUB)을 하여야 하므로 접근시간(access time)이 빠른 고속의 메모리가 요구되기 때문이다.

부호책부는 일련의 연습벡터(training vector)로 부터 부호책 설계 알고리즘에 의하여 얻어진 대표벡터를 효율적으로 선택하기 위해 적절히 배치해 놓은 메모리부와 그 대표벡터에 체계적으로 접근하기 위한 주소를 발생하는 카운터부로 나뉜다.

부호책부에서 부분 부호책의 대표벡터를 저장하고 있는 메모리부는 대표벡터가 효율적으로 인출되기 위하여, 전 부호책은 부류 0-15까지 할당하고 벡터 부류 0(중요화소를 포함하지 않는 벡터의 부류: CVQ 를 행하지 않음)에 해당하는 영역의 내용은 부호책으로 사용하지 않는다. 만약 이와 같이 부호책부의 메모리를 구성하지 않으면, (부류-1)을 메모리의 상위 주소로 입력하여야 하므로, 부가회로가 필요하게 된다.

대표벡터를 선택하기 위한 주소발생부는 벡터부류 레지스터, 인덱스 카운터와 벡터내 요소선택 카운터로 구성되며, 먼저 데이터 입력부에서 전달받은 벡터부류를 메모리의 상위주소에 입력시켜 각 부류에 따라 저장되어 있는 부분 부호책의 첫 주소를 결정한 후, 인덱스 카운터에 의해서 대표벡터가 결정되면, 벡터내 요소 카운터에 의하여 4×4 단위의 주소를 발생하여 벡터내 요소를 차례로 선택한다.

연산부는 입력된 데이터와 부호책에 있는 데이터 중 같은 부류에서 가장 오차거리가 가장 작은 벡터를 찾기 위한 기능과 재생시에 필요한 덧셈 기능을 수행하여야 한다. 이 기능을 효율적으로 수행하기 위하여는 3단 파이프라인 구조가 효율적이고, 이 구조는 앞절의 MC부와 유사하다.

재생부는 MC부에 새로운 이전화면을 구성하기 위해 필요한 주소를 발생시키기 위한 카운터로 구성되어 있고, 부호책에서 찾아진 데이터와 이전 화면의 데이터를 더해주는 것은 연산부의 파이프라인의 첫 단계를 그대로 이용한다.

마지막으로, 본 시스템은 MC부와 CVQ부가 비동기적, 독립적으로 동작하고, 재생화면을 공유하고 있으므로, 기록시 주소선과 데이터선에 충돌이 발생할 수 있다. 이것을 해결하고자 분류 벡터양자화부에 메모리 사용 우선권을 주고, 각 부가 메모리 사용시에 깃발(flag)을 이용하여 상대방에 알려주어 충돌을 피하였다. 이와 같이 분류 벡터양자화기부에 우선권을 주는 이유는 이동보상부의 연산량이 이동보상부의 연산량이 분류 벡터양자화부의 연산량 보다 많기 때문이다. 본 시스템에서 이동보상부와 분류 벡터양자화부의 연산량 비교는

$$\text{MC부 연산량} = (25 \times 8 \times 8) \times (1 - S_{mc}) : \text{MC 수행시간}$$

$$\text{CVQ부 연산량} = 64 : \text{큐 메모리로 부터 이동보상 예측오차 인출시간} + 5 : \text{큐 메모리로 부터 블럭부류, 벡터부류 인출시간} +$$

$$(64 \times 4 \times 4) \times (1 - S_{cvq}) \times m : \text{CVQ 수행시간} + 20 : \text{이동보상부에 이전 화면 구성에 요구되는 시간}$$

과 같다. 위의 식에서 S_{mc} 와 S_{cvq} 는 각각 이동보상시와 분류 벡터양자화시의 오차거리(error distance)가 가장 작은 블럭과 벡터를 찾을 경우, 비교부의 최소값 레지스터에 있는 값 보다 벡터와 블럭내의 모든 요소를 처리하지 않은 현재의 값이 더 작을 경우, 남은 요소에 대한 연산이 더 이상 필요하지 않으므로 다음 블럭을 새로이 처리하는 비율이다. 이 비율에서 S_{mc} 는 영상 취득부에서 영상 취득시 잡음이나, 원 여상이 갖는 큰 데이터의 분산 때문에 약 0.6 정도이고, S_{cvq} 는 이동보상 예측오차의 적은 분산 때문에 0.3 정도이다. 따라서 실제의 계산시 \bar{m} 를 고려하여도 CVQ의 연산량이 MC부의 연산량 보다 크게 된다.

5. 영상 수신부의 설계

이동보상부와 벡터양자화기로 구성된 영상송신부에서 부호화되어 채널로 전송된 데이터를 수신측에서 재생하는 방법과 채널데이터의 형식을 그림 13에 표시하였다.

채널데이터는 8bit의 이동벡터(dy, dx), 블럭의 중요화소 유무를 표시하는 1비트의 EN, EN이 1일때 중요벡터의 위치를 표시하는 4비트의 블럭부류 B_i ($i = 0, 1, 2, 3$), 그리고 부분 부호책이 64개의 16부류로 구성된 경우 벡터부류 V_c 4비트와 부호책 인덱스 6비트를 합하여 10비트의 K_i 로 구성된다.

이동벡터(dy, dx)와 블럭내 중요화소 유무를 나타내는 EN은 때 블럭마다 발생되며, B_i 와 K_i 는 오차벡터의 중요화소 유무에 따라 다르다. 중요화소가 존재하는가에 대한 정보인 EN이 0일 때는 B_c 와 K_i 는 발생하지 않으며, EN이 1일 때는 4bit의 B_i 와 B_i 의 값이 1인 것에만 K_i 가 발생한다.

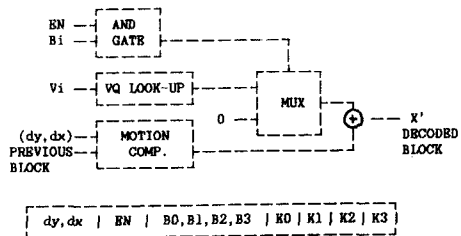


그림 13. 복호 과정과 채널부호
Fig. 13. Decoding operation and channel symbol.

영상을 복호화하는 과정은 다음과 같다.

- (1) 이전화면의 블럭을 (dy, dx)만큼 이동보상한다.
- (2) EN을 검사한다.

EN=0이면, (1)의 결과가 부호화된 블럭이다.

EN=1이면, (3)을 수행한다.

- (3) i=0부터 3까지 Bi를 검사하여

Bi= 1이면, 부호책에서 Ki 번째의 오차벡터를 블럭에 더한다.

이 과정을 실현하기 위하여 그림14와 같이 제어부, 부호책참조(codebook lookup)부, 주소발생부, 메모리부, 연산부로 복호기를 구성하였다. 제어부는 마이크로프로그램을 발생하는 순서제어기로 구성되어 있으며, 각 부를 제어한다. 부호책참조부는 큐메모리의 데이터로 벡터의 주소에 해당하는 오차신호를 발생하는 역할을 한다. 어드레스 발생부에서는 블럭내의 벡터단위로 주소발생할 수 있도록 구성하였고, 이동보상을 위한 이전화면의 주소(previous frame address, PA), 현화면의 재생주소(current frame address, CA)를 발생시킨다. 메모리부는 3화면의 영상데이터를 저장할 수 있도록 하였으며 주소발생부에서 제공되는 2가지의 주소 이외에 D/A address에 의하여 복호가 완료된 영상을 모니터에 표시할 수 있도록 되어있다.

연산부는 마이크로프로그램의 제어에 의해 이전화면의 이동보상된 블럭의 데이터(previous data, PD)와 부호책의 오차벡터(ei)를 더하여 재생된 데이터(current data)를 만든다.

6. 제작및 성능 측정

앞 절에서 설계된 영상 부호화장치는 Intel Multi-bus 규격의 기판에 구현되었다. 영상 입력부, 이동보상부, 분류 벡터양자화부, 복호화부가 각각 한장의 기판에 구현되었으며, 이 모든 부분을 마이크로 프로그래밍기법에 의하여 운용하기 위하여 AMD 사의 시퀀서 AM2910ADC 3개를 사용하였다.

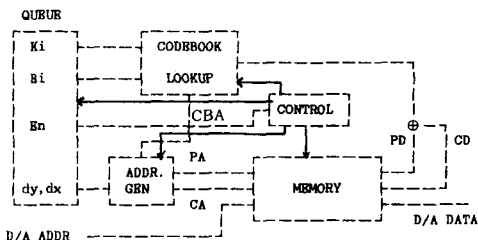


그림14. 복호부의 하드웨어 구성
Fig. 14. Block diagram of the decoder.

화면간 부호화기법을 이용한 영상부호화기는 첫 프레임의 전송이 요구된다. 즉 부호화가 시작되는 시점에서의 이전화면은 화면간 부호화기법에 의해 만들어질 수 없으므로 화면내 부호화기법을 이용하여 부호화하여 전송하여야 한다. 그러나 본 연구에서 첫 화면을 모두 계조도 0으로 설정하고 부호화를 행하였다. 이러한 방법을 채택한 근거는 소프트웨어 모의실험에서 같은 조건으로 부호화를 행하였을 경우 30화면 이내에 첫 화면을 전송한 경우와 성능이 비슷하였다. 따라서 본 부호화기의 속도가 초당 10장 정도의 부호화가 가능하므로 3-4초내에 첫 화면을 전송한 경우와 같은 동작을 행한다.

제작된 장치의 초당 부호화 속도, 화소당 비트율과 전송비트율을 살펴보면 표 3과 같다.

표 3. 초당 화면 부호화 속도
Table 3. Frame coding rate in 1-sec.

최대 부호화 속도	15 frames/sec
일상 부호화 속도	13 frames/sec
최소 부호화 속도	6 frames/sec

이 측정은 최대 부호화속도의 경우 정지화면을 이동보상만을 하였을 경우 측정하였고, 최소 부호화속도는 중요화소를 판단하는 임계값을 0으로 하여 이동보상후에 나타나는 정보를 모두 분류 벡터양자화기에서 모두 중요화소로 만들어서 측정하였다. 그리고 일상 부호화 속도는 본 장치가 영상 전화기의 목적이므로 사람들이 일반적으로 전화로 통화할 때의 상태를 부호화하는데 필요한 속도를 측정하였다.

표 3의 실험결과는 화소당 비트율과 채널전송율을 본 논문 I의 식 6을 이용하여, 설계시 미리 정해진 값(블럭크기=8×8, 이동벡터 비트율 α=8, CVQ인덱스와 벡터부류 Vc의 비트율 β=10, 벡터크기=4×4, 블럭부류 Bc를 위한 t=4)과 로직 어널라이저(Tektronix 1241)를 이용하여 블럭부류 Bc, 벡터부류 Vc, 중요화소의 존재 유무를 나타내는 EN을 측정하고, 소프트웨어로 분석하여 화소당 비트율 (bits/pel)과 채널 전송율 (kbps)를 구하였다.

화소당 비트율은

$$r(MC-CVQ) = \frac{\alpha + 1}{M^2} + \frac{t}{M^2}pb + \frac{\beta \bar{m}}{M^2}pb$$

α : 이동보상시 이동벡터를 전송하기 위한 비트율 (bits/block)

β : 분류 벡터양자화시 인덱스와 벡터부류 V_c 를 전송하기 위한 비트율 (bits/vector)

pb : 전체 블록에서 중요화소가 존재하는 블록의 발생확률

t : 블록내의 벡터의 갯수

N^2 : 이동보상시 블록의 크기

M^2 : 벡터양자화시 벡터의 크기

\bar{m} : 블록내에 중요화소를 포함하는 벡터의 평균 갯수

이코 채널 전송율은

$$R(MC-CVQ) = r \times Nt^2 \times T$$

r : 화소당 비트율

Nt^2 : 영상 취득시의 영상의 크기

T : 초당 부호화 속도

이다. 표 4에서 경우 1에서 경우 4로 갈수록 부호화하여야 할 내용이 많은 정보이다. 즉 움직임이 많이 있는 것을 뜻한다. 표 3에서 보듯이 m 와 pb 가 증가함을 알 수 있다. 마지막으로 채널 전송율은 위의 화소당 비트율에 128×128 을 곱하고, 다시 T (초당부호화 속도)를 곱하여 구한다. 어느 경우이나 40Kbps 내외가 되는 이유는 부호화하여야 할 정보가 많은 경우에는 계산량이 많아 T 가 작기 때문이다.

마지막으로 본 연구에서는 채널 부호화부는 고려하지 않았지만 채널 부호화부에 엔트로피 코딩을 추가시키면 전송율을 더욱 줄어 들 수 있다.

표 4. 실험 결과

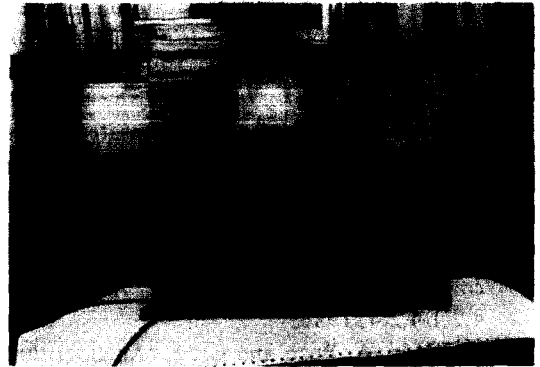
Table 4. Experiment results.

case	m	pb	T	r (bits/peI)	R (kbps)
1	0.24	0.152	12	0.203	39.90
2	0.40	0.245	10	0.263	43.18
3	0.62	0.421	7	0.409	46.99
4	0.91	0.521	5	0.566	46.30

VII. 결 론

본 연구에서는 1초에 10장의 화면의 화면을 부호화하기 위하여 연속적으로 영상을 취득할 수 있는 디지털이저를 구성하고, 부호화부는 MC와 CVQ 부를 구분하여 마이크로프로그램 제어에 의한 멀티프로세서 시스템으로 구성하였다.

MC부와 CVQ부 내부에서 효율적으로 벡터단위 연산을 효율적으로 처리하기 위하여 3-단계 파이프라인 구조를 제안하여 실현하였다.



사 진 1. 제작된 영상부호화 장치
Photo 1. Designed video CODEC.

이동보상부에서 이전 화면은 따로 전송되지 않고, 분류 벡터양자화기에서 재생한 화면을 페이지를 전환하는 구조를 이용하여 전송시간을 제거함으로써, 부호화 시간을 단축시켰다.

일반적으로 동 화상의 영상 부호화장치에서는 화면각 부호화장치와 더불어 영상의 첫 화면을 화면내 부호화장치 (intraframe CODEC)를 이용하여 부호화하고, 이동보상을 이용한 화면각 부호화 도중 양질의 화면을 유지하기 위하여, 화면내 부호화장치를 추가하여야 하나, 본 부호화장치에서는 화면내 부호화장치를 이용하지 않았지만, 회로동작 3-4초 이내에 첫 화면을 전송한 경우와 같은 화질을 할 수 있었다.

위의 사실로써 제안한 분류 벡터양자화기가 좋은 성능임을 입증하였다. 그리고 여러 실제의 영상으로 확인한 결과 1초당 10장 이상 부호화함을 확인하였다.

현재 국제 표준화의 방향으로 진행되는 MC-DCT을 행하기 위한 연구에서 본 연구에서의 MC부는 그대로 이용이 가능하다.

본 연구에서 계속 진행하여야 할 부분은 움직임이 큰 영상을 부호화시에 계산 시간이 대부분 소요되는 CVQ부에서 단축 탐색 방법에 대한 연구와 컬러 동화상 부호화로의 발전에 대한 연구를 진행하여야 할 것이다.

参 考 文 献

[1] H. Yamamoto and Y. Hitori, "Recent technology of video conference and videophone,"
[2] J.R. Jain and A.K. Jain, "Displacement

measurement and its application to inter-frame image coding," *IEEE Trans. on comm.*, voll COM-29, pp. 1799-1808, Dec. 1981.

[3] T. Koga et al., "Motion compensated image coding for video conferencing," *Nat Telecom. Conf*, pp. G5. 3. 1-G5, 3,5, Nov. 29-Dec. 3, 1981.

[4] T. Murakami, et al., "Vector quantization of color image," *IEEE ICASSP Tokyo*, pp. 4.2.1-4.2.4, 1986.

[5] Y. Linde, A. Buzo and Gray, "An algorithm for vector quantizer design," *IEEE Trans. on comm.*, vol. COM-28, no. 1, pp. 84-95, Jan. 1980.

[6] D.E. White, *Bit-Slice Design: Controllers and ALUs*, Garland STPM Press, 1981.

[7] Michael Andrews, *Principles of Firmware Engineering in Microprogram Control*, Computer science Press, 1980.

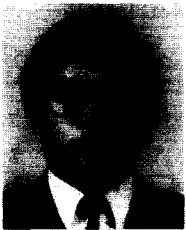
[8] Kai Hwang and F.A. Briggs, *Computer Architecture and Parallel Processing*, McGraw-Hill, 1984.

[9] 김진태, 주창희, 최종수, "움직임 추정을 위한 Block Matching Algorithm의 비교," 텔레콤, 제 4 권 2호, pp 27-33, 1988년 12월

著 者 紹 介

全 重 滿 (正會員) 第27卷 第3號 參照
현재 연세대학교 전자공학과 박사과정

崔 成 男 (正會員) 第27卷 第3號 參照
현재 연세대학교 전자공학과 박사과정



申 泰 民 (正會員)
1959年 12月 15日生. 1981年 2月 연세대학교 전자공학과 졸업. 1983年 2月 연세대학교 전자공학과 졸업(석사과정). 1989年 2月 연세대학교 전자공학과 졸업(박사과정). 현재 경상대학 전자공

학과 근무. 주관심분야는 영상처리, 컴퓨터 아키텍처 등임.

朴 圭 泰 (正會員) 第26卷 第7號 參照
현재 연세대학교 전자공학과 교수