

벡터양자화에서의 계산량과 압축률의 개선

정 일 화[†] · 홍 총 선^{††} · 이 대 영^{†††}

요 약

본 논문에서는 영상 벡터양자화에서의 계산량과 압축률 개선을 위한 새로운 방법을 제안하였다. 제안된 방법은 코드북 부분 검색, 코드벡터 요소의 부분 사용, 검색 중단조건 등을 이용하여 코드북 구성, 인코딩 과정에서의 계산량을 줄일 수 있었다. 또한, 선택된 코드북 인덱스 무손실 압축에서의 압축률을 높이기 위해 코드북 재배열과 가변길이 코딩(Variable Length Coding)을 사용하였다.

An Improvement on Computation Cost and Compression Ratio of Vector Quantization

Il-Hwa Jung[†] · Choong-Seon Hong^{††} · Dae-Young Lee^{†††}

ABSTRACT

In this paper, new image vector quantization method for improvement computation cost and compression ratio is proposed. A proposed method could saved the computation cost of codebook generation and encoding using partial codebook search, partial codevector elements, and interruption criterion. And, to improve compression ratio of codebook index lossless coding, codebook rearrangement and variable length coding scheme are used.

1. 서 론

문자, 그래픽, 오디오, 영상 등의 총합체로서 구성되는 멀티미디어 데이터 중에서 영상신호는 그 의미 전달 효과가 매우 크기 때문에 멀티미디어 데이터의 핵심이라고 할 수 있지만, 데이터량이 매우 방대하므로 영상데이터 압축은 멀티미디어 서비스에서의 핵심과제 중 하나라고 할 수 있고, 이러한 영상데이터의 압축을 위해 매우 다양한 연구들이 이루어져 왔으며, JPEG (Joint Photographic Experts Group), MPEG (Moving Pictures Experts Group) 등에 의해 제정된 표준안이 상용화되어 있는 상황이다[4, 5].

영상 압축 방법은 크게 예측부호화(Predictive cod-

ing) 방식과 변환부호화(Transform coding) 방식으로 대별되는데, 예측부호화 방식은 영상 샘플이 갖는 상관관계를 이용하는 방법으로서, 압축률이 낮을 때는 복원영상의 왜곡이 적고 시스템의 구성이 간단하다는 장점을 갖지만 압축률을 높이면 왜곡이 크게 증가하는 단점을 갖고, 변환부호화 방식은 영상 화소간에 높은 상관도를 갖는 공간영역에서 주파수영역으로의 변환을 통해 상관도를 낮추고 에너지를 집중시킨 후에 높은 에너지 분포를 갖는 변환계수만을 부호화하는 방식으로서, 비교적 높은 압축률에서 좋은 성능을 나타내지만 시스템의 구성이 복잡해지게 된다[7, 12].

반면에 이와는 다른 압축방식인 벡터 양자화(Vector Quantization)가 1980년 Y. Linde, A. Buzo, R.M.Gray 등에 의해 LBG 알고리즘이 제안된 이후로 많이 연구되어져 왔는데, 압축효율에 비해 시스템의 구조가 간단하다는 특히, 수신기의 구조가 매우 간단하다는 장

† 정 회 원 : 신성대학 컴퓨터계열 교수
†† 종신회원 : 경희대학교 전자정보학부 교수
††† 정 회 원 : 경희대학교 전자정보학부 교수
논문접수 : 2000년 8월 16일, 심사완료 : 2000년 10월 24일

점을 지니지만, 코드북(codebook)의 크기를 줄여 압축률을 높이면 복원 영상의 화질, 특히 윤곽(edge) 부분의 화질이 저하되고, 화질 저하를 막기 위하여 코드북의 크기를 크게 하면 코드북 구성 시간, 부호화(en-coding) 시간, 코드북 저장을 위한 기억장소의 증가 등의 문제점 또한 지니고 있는 방법이라 할 수 있다[1,2].

따라서 본 논문에서는 복원영상의 화질저하가 커지지 않는 범위 내에서 코드북 구성과 인코딩 과정에서의 계산량을 줄이는 방법과, 인코딩 과정에 의해 선택되어 전송되어지는 코드북 인덱스값의 무손실압축을 통해 압축률을 높일 수 있는 새로운 방법에 대해 제안하고자 한다.

본 논문의 구성은 2장에서 계산량 감소와 압축률 향상을 위한 기존의 알고리즘에 대해 알아보고, 3장에서 새로운 방법에 대한 제안을 거쳐 4장에서 이에 대한 실험결과와 고찰을 5장에서 전반적인 결론을 맺는 순서로 구성하였다.

2. 기존의 알고리즘

2.1 코드북 구성과 인코딩을 위한 알고리즘

2.1.1 LBG(Lide-Buzo-Gray)[1]

(1) 코드북 구성

- 1) 트레이닝 데이터로 사용될 영상을 $n \times n$ 의 부분블록으로 나누어 이를 n^2 차원의 벡터로 만든다.
- 2) 초기 클러스터링(clustering) 중심 $Y_i(t)$ 를 정한다.
 $i = 0, 1, \dots, N - 1$ (t =반복 횟수, N =코드북의 갯수)
- 3) 트레이닝 벡터를 최소오차를 갖는 클러스터 $C_i(t)$ 로 분류한다.

$$d(x, y) = \sum_{i=1}^n (x_i - y_i)^2 \quad (1)$$

- 4) 클러스터 $C_i(t)$ 의 중심 $Y_i(t)$ 를 식②에 따라 갱신한다.

$$Y_i = \frac{1}{n_i} \sum_{x \in C_i} x \quad (2)$$

(n_i : C_i 로 분류된 트레이닝 벡터의 수)

- 5) 식 (3)의 조건을 만족하면 중단하고, 아니면 3)부터 반복한다.

$$D = \sum_{i=0}^{N-1} \sum_{x \in C_i} \|X - Y_i\|^2, \frac{D(t) - D(t+1)}{D(t)} < \epsilon \quad (3)$$

(2) 인코딩

- 1) 전송하려고 하는 입력영상을 $n \times n$ 의 영역으로 나누어, 코드벡터와 같은 차원의 벡터로 구성한다.
- 2) 이미 구성되어 있는 N 개의 코드벡터와 식 (1)로 주어지는 오차를 계산하여 그 값이 최소가 되는 코드벡터의 인덱스를 전송한다.

2.1.2 Partial Distortion Search(PDS)[3]

- (1) 입력벡터와 가장 오차가 작은 코드벡터를 선택하는 과정을 변형하여 계산량을 줄이는 방법
- (2) 입력벡터 X_i 와 최소오차를 갖는 코드벡터를 찾는 과정에서 다음과 같이 가정한다.

Y_{\min} : 현재까지 최소 오차를 갖는 코드벡터

d_{\min} : X_i 와 Y_{\min} 사이의 오차

코드벡터의 크기 : 4×4 블록으로 구성된 16차원

- (3) 새로운 코드벡터 Y_j 와 X_i 의 오차 $d(X_i, Y_j)$ 를 일괄 적으로 계산하지 않고, 벡터요소에 대해 차례로 오차를 계산하여 누적시켜 진행하다가 그 값이 d_{\min} 을 넘어서면 계산을 중단하고 다음 코드벡터로 이 동작하는 형식을 취하여 계산량을 줄이게 된다.

2.1.3 Mean-distance-ordered Partial codebook Search (MPS)[6]

- (1) 식 (1)에서의 Euclidean distance의 계산여부를 식 (1)를 이용하여 결정하는 방식이다.

$$d_M(X, Y) = \left(\sum_{i=1}^m X_i - \sum_{i=1}^m Y_i \right)^2$$

$$d_M(X, Y) \leq m \times d(X, Y) \quad (4)$$

- (2) 코드북을 평균값 순LBG(Lide-Buzo-Gray)[1]으로 정렬한 후 실행하며, 검색 단계에서 현재까지 결정된 가장 유사한 코드벡터가 Y_{\min} 이라고 가정하면, 코드벡터 Y_i 가 식 (4)에서의 $d_M(X, Y_i) \leq m \times d_E(X, Y_{\min})$ 를 만족할 경우에 만 $d(X, Y_i)$ 를 계산하는 방식을 취한다.

(3) 이 때, $\sum_{i=1}^m Y_i$ 는 미리 계산할 수 있으므로 인코딩 시의 계산량을 많이 줄일 수 있게 된다.

2.1.4 Kick-out condition by Wu and Lin[11]

(1) 식 (1)을 식 (5)에서와 같이 전개하여 사용한다.

$$d(X, Y) = \sum_{i=1}^m (X_i - Y_i)^2 = \sum_{i=1}^m X_i^2 + \sum_{i=1}^m Y_i^2 - 2 \sum_{i=1}^m X_i Y_i \quad (5)$$

(2) 식 (5)에서 $\sum_{i=1}^m X_i^2$ 는 공통으로 포함된 항이고

$\sum_{i=1}^m Y_i$ 는 미리 계산할 수 있는 값이므로 계산량을 줄일 수 있다.

2.2 코드북 인덱스의 무손실 압축을 위한 알고리즘

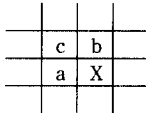
2.2.1 DPCM/Huffman coding[6]

인접영역의 인덱스는 상관도가 높다는 전제에 의해 <표 1>에서 주어지는 예측기를 이용하여 X에 대한 예

<표 1> DPCM에서의 예측기와 화소위치

No	예측치 Y
1	Y=a
2	Y=b
3	Y=c
4	Y=a+b+c
5	Y=a+(b-c)/2
6	Y=b+(a-c)/2
7	Y=(a+b)/2

(a) 예측기



(b) 화소위치

<표 2> JPEG 무손실 모드에서의 허프만테이블

오차값의 범위	코드	부가비트수	총비트수
0	00	0	0
-1 ~ 1	010	1	2
-3 ~ 3	011	2	4
-7 ~ 7	100	3	5
-15 ~ 15	101	4	6
-31 ~ 31	110	5	7
-63 ~ 63	1110	6	8
-127 ~ 127	11110	7	10
-255 ~ 255	111110	8	12

측치 Y를 계산한 후 X와 Y의 오차를 <표 2>의 허프만 테이블을 이용하여 무손실 압축을 행하게 된다.

2.2.2 Search-Order Coding (SOC)[9]

- (1) DPCM/Huffman coding에서와 같이 인접영역의 코드북 인덱스간의 상관관계가 높은 것을 이용한다.
- (2) (그림 1)과 같은 검색 경로에서 같은 인덱스가 발견되면 그 위치를 전송하고, 발견하지 못하면 원래의 인덱스를 전송하는 방식을 사용한다.
- (3) 검색순서를 표현하기 위해 m비트를 사용하면 2^m 개의 인덱스 확인 가능하고, 일치하는 인덱스를 검색하였는지를 표현하기 위해 1비트 추가되므로 전송되는 데이터량은 다음과 같게 된다.

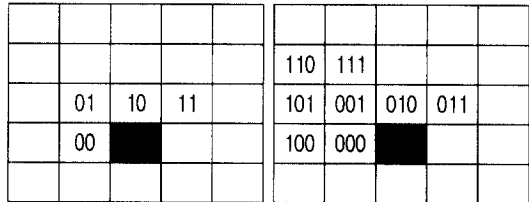
$$\text{데이터량} = (1+m) \times s + (1+\log_2 N) \times (L-s)$$

s : 인덱스가 일치하는 영역의 수

N : 코드벡터의 수

L : 전체 영역의 수

m : 검색경로 표현을 위한 비트 수



(그림 1) SOC에서의 인덱스 검색 경로

2.2.3 SOC with Correction Code[10]

SOC 방법에서와 같은 검색경로와 방법을 사용하지만 인덱스가 일치하는 위치만을 표시하지 않고, 정해진 범위 내에서 오차가 발생하는 부분에 대해서도 표현을 하는 방법이다. 예를 들어 오차표현을 위해 2비트를 사용한다고 가정하면 인덱스의 차이가 -4~4 가지를 포함하게 된다. 하지만 인덱스의 일치여부를 표현하기 위해 1비트만이 추가되는 SOC 알고리즘에 비해 오차 허용 범위 내인지까지를 표현해야 하므로 2비트가 필요하게 된다.

$$\text{데이터량} = (2+m) \times s + (2+m+n) \times c + (2+\log_2 N) \times (L-s-c)$$

s : 인덱스가 일치하는 영역의 수

N : 코드벡터의 수

- c : 오차 범위내의 영역 수
- L : 전체 영역의 수
- m : 검색경로 표현을 위한 비트 수
- n : 인덱스의 오차표현을 위한 비트 수

3. 제안 알고리즘

3.1 코드북 구성과 인코딩을 위한 제안 알고리즘

3.1.1 1단계 : 검색영역의 축소

코드북을 구성하거나 구성된 코드북을 이용한 인코딩에서 최소오차를 갖는 코드벡터를 검색할 때, 최소 오차를 갖는 코드벡터가 자리한 위치에 근접한 영역에서 검색을 시작하면 계산량을 상당히 줄일 수 있다는 것에 착안하여, 구성된 코드북을 에너지 크기로 재배열(rearrangement)한 후, Full search를 통해 최소오차를 갖는 코드벡터 인덱스를 구하고, 단지 에너지 크기가 가장 근접한 인덱스를 구해 이들간의 오차를 확인하여 <표 3>에서와 같이 그 오차가 코드북 전체 크기에 비해 상당히 작다는 결과를 확인하여, 이를 이용하면 검색영역을 상당 부분 줄일 수 있다는 것을 확인하였다.

3.1.2 2단계 : 오차계산 대상의 축소 (코드벡터 요소의 부분 사용)

4×4 영역의 16차원의 코드벡터를 사용하여 검색 동작을 진행할 때, 입력벡터와 코드벡터의 16개 화소에 대해 각각 오차가 계산되어 이를 합한 후 최소오차와 비교하게 된다. 이러한 일련의 동작에서 만약 입력벡터가 오차가 많이 나는 코드벡터와 만나게 되면 16개 화소의 오차를 모두 더하기도 전에 이미 최소오차를 넘어서게 되므로, 나머지 계산은 불필요하게 된다. 따라서 코드벡터와 입력벡터를 여러 영역으로 분리하여 단계적으로 오차를 비교하면 계산량을 상당히 줄일

수 있게 되지만, 이 때에 너무 여러 영역으로 분리할 경우 비교연산이 많이 추가되므로 본 논문에서는 16차원의 코드벡터를 4영역으로 분리하여 적용하였다.

3.1.3 3단계 : 검색중지 조건의 도입

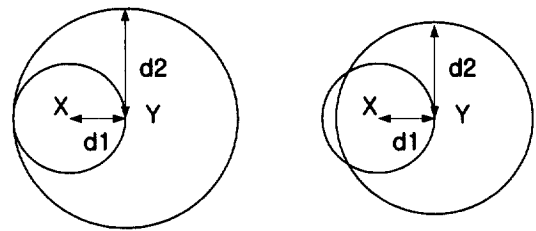
1단계에서의 검색영역 축소로 계산량을 상당히 줄일 수 있지만, 이 때에도 불필요한 계산은 여전히 존재하게 된다. 다시 말해, <표 2>에서 오차범위 -16~16에는 97.54%가 존재하고 오차범위 -32~32에는 99.6%가 존재하므로 -32~-17, 17~32의 범위에는 2.06%만이 존재하지만 이를 무시할 경우 복원영상의 화질저하는 당연한 결과가 되므로 단지 2.06%의 입력벡터에 의해 검색영역은 2배로 늘어나게 된다.

따라서 검색영역은 심각한 화질저하를 막을 수 있는 정도로 유지하되, 입력벡터에 따라 이를 가변적으로 적용할 수 있는 방법을 제안하였다.

(그림 2)에서 X를 인코딩하려는 입력벡터, Y는 X와 최소오차를 갖는 코드벡터의 검색과정에서 현재까지 최소오차를 갖는 코드벡터, d1을 X와 Y 사이의 오차, d2를 코드벡터 Y와 다른 코드벡터 사이의 오차 최소값이라 가정하면, (a)에서와 같이 $2 \times d1 \leq d2$ 의 조건을 만족하면 이는 Y와 X의 오차가 코드벡터 Y와 다른 코드벡터들간의 최소오차 보다 더 작은 오차를 갖게 되는 것이므로 Y를 X의 최소오차 코드벡터로 간주할 수 있다는 것을 의미하므로 검색을 중단해도 무방하다고 할 수 있다.

<표 3> 인덱스간의 오차분포

인덱스 오차	누적비율
0	16.24 %
-4 ~ 4	74.97 %
-8 ~ 8	90.54 %
-16 ~ 16	97.54 %
-32 ~ 32	99.60 %
-64 ~ 64	99.98 %
-255 ~ 255	100.00 %



(a) $2 \times d1 \leq d2$

(b) $2 \times d1 > d2$

(그림 2) 검색 중지 조건

그러므로 이러한 중지 조건을 활용하면 모든 입력벡터가 같은 크기의 영역을 검색하는 것이 아니라 입력벡터의 상황에 따라 검색영역이 가변적으로 적용되므로 계산량을 줄일 수 있게 된다.

3.2 코드북 인덱스의 무손실 압축을 위한 알고리즘

같은 인덱스를 갖는 검색위치를 표현하는 비트를 줄이기 위해 한 방향에 대해서만 검색한다.(본 논문에서는 <표 1> (b)화소위치에서의 위치 b 만을 사용한다.)

검색위치의 인덱스가 기준위치의 인덱스보다 큰 경우와 작은 경우를 따로 표현하여 Correction code를 사용하면 (1)기준위치와 일치하는 경우, (2)기준위 치 보다 큰 경우, (3)기준위치 보다 작은 경우, (4)오차 한계를 벗어나는 경우의 4가지로 표현되므로 이를 표현하는데 각 2비트씩이 필요하게 된다.

따라서, 본 논문에서는 (2)와 (3)의 경우를 합하여 전체를 3가지로 표현한 후 출현 빈도에 따라 1, 00, 01을 각각 할당하는 방식으로 가변길이 부호화 (variablength coding)을 이용하여 표현하였다.

이 때 (2)와 (3)의 구분을 위해 1비트의 부호비트를 추가하였는데, (2)와 (3)이 합쳐진 영역 다시 말해, 오차 범위내의 영역의 확률이 가장 높을 경우 코드 "1"을 할당하고 여기에 부호비트 1비트가 추가되어 원래의 고정비트 방식에서와 같은 결과를 보일 것이고, 경우 (1)이나 경우(4)가 빈도가 높을 경우에는 오차범위내의 영역 표현에 3비트가 사용되지만, 이보다는 더 큰 빈도를 갖는 영역의 표현에서 1비트가 줄어들게 되므로 데이터량의 감소효과를 기대할 수 있게 된다.

4. 결과 및 고찰

본 논문에서 제안한 알고리즘에 대한 검증은 위한 모의실험은 256×256(8bpp) 크기의 10가지 영상에 대해 IBM-PC(Pentium-500MHz)에서 C 언어를 사용하여 수행하였다.

4.1 코드북 구성

코드북 구성시의 계산량 비교가 <표 4> 와 (그림 3)에 제시되었는데, 이 때의 수치는 LBG(Full search) 알고리즘의 계산량을 기준으로 하였을 때의 상대적 계산량을 나타낸 수치이다.

결과를 보면, 본 논문에서 제안한 방식이 LBG알고리즘에 비해서는 79~84%의 계산량 감소가 가능했고, 기존의 알고리즘 중에서 가장 좋은 결과를 보인 MPS 알고리즘에 비해서도 16~39%의 감소가 있었음을 알 수 있다.

<표 4> 코드북 구성시의 계산량 비교(w=검색범위)

(a) 코드북 = 128개 (%)

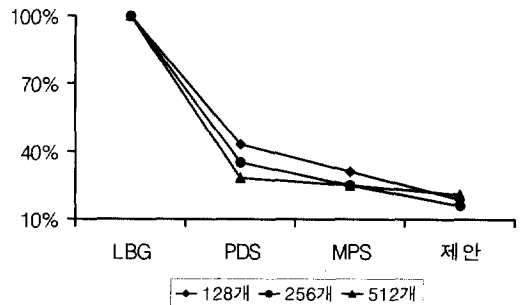
알고리즘	+	-	×	÷	>=<	Total
LBG	100	100	100	100	100	100
PDS	30	30	30	100	253	43
MPS	20	26	26	106	148	31
제안(w=12)	12	11	11	100	148	19

(b) 코드북 = 256개 (%)

알고리즘	+	-	×	÷	>=<	Total
LBG	100	100	100	100	100	100
PDS	23	25	25	98	204	35
MPS	14	20	20	98	135	25
제안(w=20)	9	8	8	90	131	16

(c) 코드북 = 512개 (%)

알고리즘	+	-	×	÷	>=<	Total
LBG	100	100	100	100	100	100
PDS	18	18	18	100	194	28
MPS	13	19	19	109	151	25
제안(w=48)	12	11	12	124	181	21



(그림 3) 코드북 구성시의 계산량 비교

4.2 인코딩(Encoding)

인코딩에서의 계산량 비교가 <표 5> 와 (그림 4)에 제시되었는데, 이 때의 수치는 LBG(Full search) 알고리즘의 계산량을 기준으로 하였을 때의 상대적 계산량을 나타낸 수치이다.

결과를 보면, 본 논문에서 제안한 방식이 LBG알고리즘에 비해서는 83.1~90.9%의 계산량 감소가 가능했고, 기존의 알고리즘 중에서 가장 좋은 결과를 보인 MPS 알고리즘에 비해서도 41~47%의 감소가 있었음을 알 수 있다.

이러한 결과는 코드북 구성시에 비해 인코딩시의 계산량이 상대적으로 더욱 감소하였음을 보여주는데, 이는 코드북 구성은 코드북 갱신이 여러 차례 이루어지

며, 이 때마다 코드북 재배열이 필요하게 되기 때문에 인코딩의 경우에 비해 상대적으로 계산량 감소가 적었다고 생각된다.

또한 실험결과에서 복원영상의 화질의 PSNR 값이 0.4~1% 정도 감소한 것을 볼 수 있는데, 이는 <표 3>의 결과에서 오차의 범위가 상당히 컸던 일부 영역에 의한 것이라 생각되지만 (그림 6)의 복원영상의 화질 비교에서 확인할 수 있듯이 큰 문제가 되지는 않다고 생각된다.

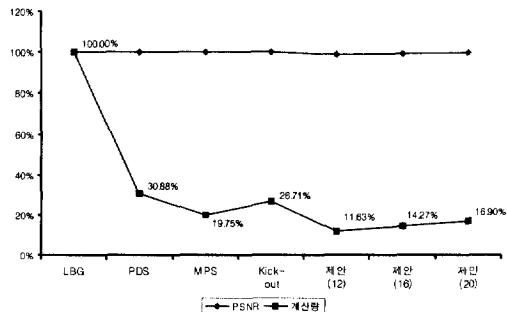
<표 5> 인코딩에서의 계산량 비교

(a) 코드북 = 128개

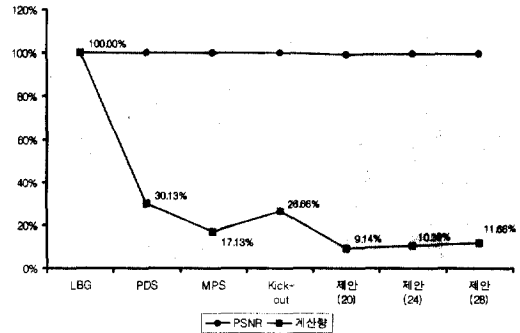
알고리즘	계산량(%)					화질
	+	-	×	>=<	Total	
LBG	100.0	100.0	100.0	100.0	100.0	100.0
PDS	23.6	23.6	23.6	381.2	30.9	100.0
MPS	14.2	19.6	19.6	114.3	19.8	99.9
Kick-out	34.5	4.2	38.8	68.0	26.7	100.0
제안(12)	9.2	8.5	9.2	139.6	11.6	99.0
제안(16)	11.7	10.9	11.7	150.5	14.3	99.3
제안(20)	14.2	13.4	14.2	161.3	16.9	99.4

(b) 코드북 = 256개

알고리즘	계산량(%)					화질
	+	-	×	>=<	Total	
LBG	100.0	100.0	100.0	100.0	100.0	100.0
PDS	23.1	23.1	23.1	370.5	30.1	100.0
MPS	11.3	17.1	17.1	111.3	17.1	100.0
Kick-out	34.4	4.3	38.7	68.3	26.7	100.0
제안(20)	6.8	6.4	6.8	128.8	9.1	99.2
제안(24)	8.0	7.6	8.0	133.8	10.4	99.4
제안(28)	9.2	8.8	9.2	138.9	11.7	99.6

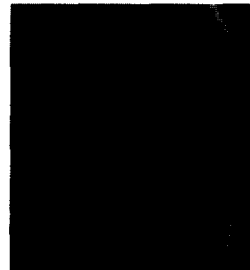


(a) 코드북 128개

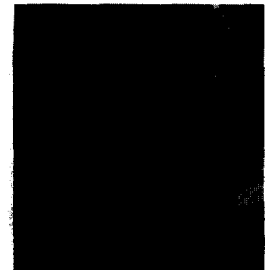


(b) 코드북 256개

(그림 4) 인코딩에서의 계산량 비교



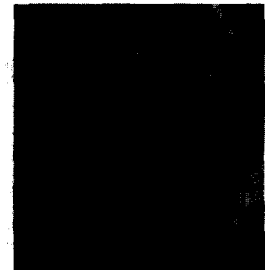
(a) 원영상



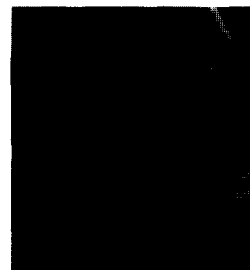
(b) LBG (26.73dB)



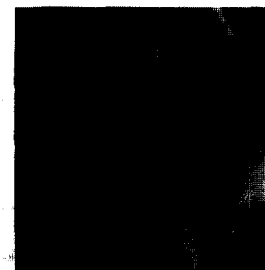
(c) PDS (26.73dB)



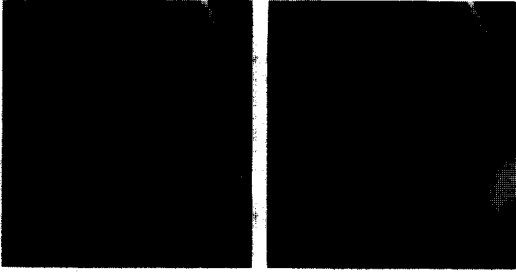
(d) MPS (26.67dB)



(e) Kick-out (26.73dB)



(f) 제안(w=20, 26.4dB)



(h) 제안(w=28, 26.61dB) (g) 제안(w=24, 26.53dB)

(그림 5) 복원영상의 화질 비교

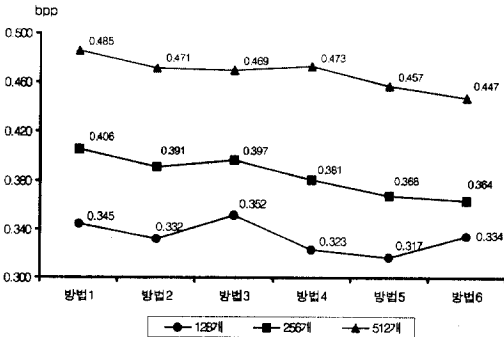
4.3 코드북 인덱스의 무손실 압축

전송되는 인덱스의 무손실 압축을 위한 제안알고리즘의 결과는 <표 6> 과 (그림 6)에 보여졌는데, 제안 알고리즘의 경우 정정부호(correction code)를 3비트(C=3)에서 5비트(C=5)까지 사용하여 다른 방식과 비교하였다.

<표 6> 에서 제안된 알고리즘의 결과는 기존의 알고리즘 중에서 가장 좋은 결과를 보이고 있는 SOC 알고리즘에 비해 4%~7% 정도의 성능 향상을 보이고 있음을 알 수 있는데, 이는 검색영역을 제한하여 검색 위치를 표현하기 위한 데이터량을 줄이고, 가변길이

<표 6> 각 압축방식에서의 화소당 비트율 비교

알고리즘	bpp		
	코드북128	코드북256	코드북512
DPCM/Huffman	0.345	0.406	0.485
SOC	0.332	0.391	0.471
SOC with CC	0.352	0.397	0.469
제안(C=3)	0.323	0.381	0.473
제안(C=4)	0.317	0.368	0.457
제안(C=5)	0.334	0.364	0.447



(그림 6) 각 압축방식에서의 화소당 비트율 비교

코드를 적용한 결과라고 생각되며, 정정부호의 크기와 코드북 크기와의 상관관계 또한 확인 할 수 있었다.

5. 결 론

본 논문에서는 영상 벡터양자화에서의 코드북 구성과 인코딩시의 계산량을 줄이기 위해 코드북 부분 검색, 코드벡터 요소의 부분 사용, 검색 중단조건 등을 이용하고 전송되는 데이터량을 줄이기 위해 코드북 재배열과 가변길이 코딩(Variable Length Coding)을 사용하는 새로운 방식을 제안하였다.

결과에서 볼 수 있듯이 계산량은 기존의 방법 중 가장 좋은 결과를 보인 MPS 알고리즘에 비해 코드북 구성에서는 16~39%, 인코딩에서는 41~47%의 감소를 보였으며, 데이터량의 경우 SOC 방법에 비해 5~7%의 감소가 있었음을 확인할 수 있었다.

하지만 복원영상의 화질에서 미세하지만 약간의 저하가 있었다는 점과 계산량 감소를 위한 코드벡터 요소의 부분 사용에서 최적의 조건을 찾기 위한 준비가 미흡했다는 부분이 아쉬운 점이라 생각된다.

따라서, 영상 영역의 특성에 따라 코드북의 종류와 크기를 달리하는 CVQ(Classified VQ)의 개념을 일부 도입하여 보완한다면, 계산량 감소와 압축률 개선에 더욱 효과가 있으리라 예상되므로 이에 대한 추가 연구가 필요하다고 생각된다.

참 고 문 헌

- [1] Y. Linde, A. Buzo, R. M. Gray, "An Algorithm for Vector Quantizer Design," IEEE Trans. on Communications, Vol.COM-28, No.1, pp.84-95, 1980.
- [2] Robert M. Gray, "Vector Quantization," IEEE ASSP Magazine, pp.4-29, 1984.
- [3] C. Bei, R. M. Gray, "An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization," IEEE Trans. on Commun., Vol.33, pp.1132-1133, 1985.
- [4] Rafael C. Gonzalez, Paul Wintz, Digital Image Processing : Addison-Wesley, 1987.
- [5] Anil K. Jain, Fundamentals of Digital Image Processing : Prentice Hall, 1989.
- [6] S. W. Ra, J. K. Kim, "A Fast Mean-Distance-Ordered Partial Codebook Search Algorithm for Image Vector Quantization," IEEE Trans. on

Circuits and Systems-II : Analog and Digital Signal Processing, Vol.40, No.9, pp.576-579, 1993.

- [7] Pamela C. Cosman, Robert M. Gray, Martin Vatteli, "Vector Quantization of Image Subbands : A Survey," IEEE Trans. on Image Processing, Vol.5, No.2, pp.202-225, 1996.
- [8] C. H. Hsieh, J. C. Tsai, "Lossless Compression of VQ index with Search-Order Coding," IEEE Trans. on Image Processing, Vol.5, No.11, pp.1579-1582, 1996.
- [9] 김백기 "지문영상의 무손실 압축 및 분류 알고리즘에 관한 연구", 박사학위논문, 경희대학교, 1997.
- [10] E. C. Liu, T. C. Wang, "An Improvement on Lossless Compression of VQ Indexes" Proceedings of the Globecom '98 - Vol.3, pp.1699-1704, 1998.
- [11] Kuang-Shyr Wu, Ja-Chen Lin, "Fast VQ Encoding by an Efficient Kick-Out Condition," IEEE Trans. on Circuits and System for Video Technology, Vol. 10, No.1, pp59-62, 2000.
- [12] Jaeho Hwang, Choongseon Hong, Daeyoung Lee, "Improved wavelet image compression using correlation of VQ index," 한국정보처리학회 논문지 제7권 제6호, pp.1956-1963, 2000



정 일 화

e-mail : jungih@shinsung.ac.kr
 1991년 경희대학교 전자공학과 (공학사)
 1993년 경희대학교 전자공학과 (공학석사)
 1993년 ~ 현재 경희대학교 전자공학과 박사과정

1997년 ~ 현재 신성대학 컴퓨터계열 교수
 관심분야 : 영상압축, 영상통신 등



홍 충 선

e-mail : cshong@nms.kyunghee.ac.kr
 1983년 경희대학교 전자공학과 졸업(학사)
 1985년 경희대학교 전자공학과 (공학석사)
 1997년 게이오대학교 정보통신공학과(공학 박사)

1988년~1999년 한국통신 통신망 연구소 선임 연구원/네트워킹연구실장
 1999년~현재 경희대학교 전자정보학부 전임강사
 관심분야 : 인터넷 서비스 및 망 관리 구조, 분산컴포넌트관리, IP 멀티캐스트, 멀티미디어스트리밍 등



이 대 영

e-mail : dylee@nms.kyunghee.ac.kr
 1964년 서울대 물리학과 졸업 (학사)
 1971년 캘리포니아 주립대학원 컴퓨터학과 (공학석사)
 1979년 연세대학교 전자공학과 (공학 박사)

1971년~현재 경희대학교 전자정보학부 교수
 1990년~1993년 경희대학교 산업정보대학원 대학원장
 1999년~2000년 한국통신학회 회장
 관심분야 : 영상처리, 컴퓨터 네트워크, 컴퓨터 시스템