

인덱스 테이블을 이용한 고속 VQ 코드북 탐색 알고리즘

황재호[†]·곽윤식^{††}·홍충선^{†††}·이대영^{††††}

요약

본 논문에서는 Wavelet/VQ 코딩 기법에 효율적인 두 가지의 고속 VQ 코드북 탐색 알고리즘을 제안한다. 웨이브렛 계수 블록 내에서 코드북의 코드워드들의 원소, 즉 웨이브렛 계수가 큰 값을 가질 확률이 적다. 이러한 특성을 코드북 탐색에 이용하기 위하여 각 서브밴드에서 코드워드의 에너지 크기로 재정렬된 인덱스 테이블을 이용한다. 코드북 탐색 시 인덱스 테이블을 이용하여 코드워드의 큰 계수 값과 그에 대응하는 입력벡터의 원소를 먼저 비교함으로써 PDE의 탈출 조건을 조기에 만족하게 되어 탐색 시간이 줄어들게 된다.

Fast VQ Codebook Search Algorithms Using Index Table

Jae-Ho Hwang[†] · Yoon-Sik Kwak^{††} · Choong-Seon Hong^{†††} · Dae-Young Lee^{††††}

ABSTRACT

In this paper, we propose two fast VQ codebook search algorithms efficient to the Wavelet/VQ coding schemes. It is well known that the probability having large values in wavelet coefficient blocks is very low. In order to apply this property to codebook search, the index tables of the reordered codebook in each wavelet subband are used. The exit condition in PDE can be satisfied in an early stage by comparing the large coefficients of the codeword with their corresponding elements of input vector using the index tables. As a result, search time can be reduced.

1. 서론

DCT를 기반으로 하는 JPEG보다 영상의 압축 효율을 한층 더 높일 수 있는 기법들에 관한 많은 연구가 이루어지고 있으며 이에 대한 새로운 표준화 작업이 진행 중에 있다. 웨이브렛을 이용한 기법은 이러한 새로운 표준의 한 후보로써 현재 많은 연구가 이루어지고 있다. 웨이브렛을 이용하면 JPEG보다 몇 가지 장점이 있다. 먼저 JPEG은 8×8 블록 단위로 영상을 부

호화 하기 때문에 통신 속도가 느린 경우에는 수신측에서 복호화하여 화면에서 볼 때 블록 단위로 나타나기 때문에 빠른 시간 내에 영상 정보의 내용을 이해하기 어렵다. 그러나 웨이브렛을 이용하면 수신측에서 전체 영상이 흐릿하게 먼저 보여지고 점점 세밀하게 보여지기 때문에 보다 빠른 시간 내에 영상의 내용을 파악할 수 있다. 이는 인터넷 웹 브라우저와 같은 분야에 적용될 수 있다. 또한 부호화 시 웨이브렛 서브 밴드들의 특성에 따라 적절하게 비트를 할당함으로써 압축 효율을 높일 수 있다.

압축 효율을 한층 더 높이기 위하여 웨이브렛 변환을 통하여 생성된 계수들에 벡터 양자화(Vector Quantization)를 행하는 Wavelet/VQ 기법들이 많이 제안되

† 정 회 원 : 극동정보대학 멀티미디어과 교수
†† 정 회 원 : 중주대학교 컴퓨터공학과 교수
††† 중신회원 : 경희대학교 전자정보학부 교수
†††† 정 회 원 : 경희대학교 전자정보학과 교수
논문접수 : 2000년 8월 28일, 심사완료 : 2000년 10월 10일

었다[1]. 벡터 양자화는 스칼라 양자화 보다 코딩 효율이 뛰어나기 때문에 많이 사용되지만 가장 큰 문제는 부호화 시 입력되는 벡터에 대한 코드북의 코드워드, 즉 인덱스 값을 찾는 데 많은 시간이 소요되는 단점이 있다. 이는 곧 전체 부호화 시스템의 성능을 저하시키고 구현 시 높은 비용을 유발한다. 이러한 단점을 보완하기 위해서 고속 부호화 알고리즘들이 제안되었다.

초기의 알고리즘으로 구현이 간단하고 성능이 우수한 PDE(Partial Distance Elimination)[2] 알고리즘이 널리 사용된다. PDE는 입력 벡터와 코드워드의 거리(distance)를 계산 시 중간에 계산된 값이 현재까지 계산된 최소 거리를 초과할 때는 더 이상의 계산을 수행하지 않음으로써 부호화 시간을 단축하게 된다. 그리고 코드북을 미리 에너지 크기 순으로 배열한 테이블과 수학적인 수식을 통해 최소의 거리를 가질 수 없는 코드워드에 대해서는 계산을 하지 않음으로써 탐색 영역을 줄이는 TIE(Triangle Inequality Elimination) 알고리즘들[3-5]이 제안되었다. 최근에는 코드워드를 웨이브렛 변환하여 TIE를 적용한 알고리즘[6]이 제안되었다. 이 외에도 많은 성능이 우수한 고속 부호화 알고리즘들이 제안되었다[7-11].

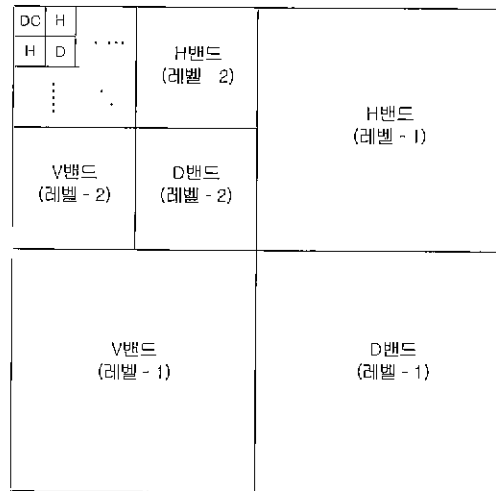
위에서 언급한 고속 부호화 알고리즘 중에서 [7, 8, 11]은 공간 영역, 즉 화소 값 자체에 대한 벡터 양자화에만 적용 가능하며 웨이브렛 영역에 적용하기에는 적합하지 않다. 그리고 [4, 6]과 같이 MTIE(Multiple TIE)를 기본으로 하는 알고리즘들은 성능이 우수하나 많은 메모리를 사용하고 코드북 탐색 과정이 복잡한 단점이 있다.

본 논문에서는 Wavelet/VQ를 기본으로 하는 영상 압축 기법에 이용할 수 있는 효율적인 두 가지의 고속 VQ 코드북 탐색 알고리즘을 제안한다. 알고리즘 I은 PDE의 탈출 조건(exit condition)을 조기에 만족시키기 위하여 코드북의 코드워드를 재정렬한다. 재정렬된 코드북의 인덱스 테이블을 이용하여 큰 웨이브렛 계수 값들을 먼저 비교함으로써 코드북 탐색 시간을 줄이게 된다. 그리고 알고리즘 II는 알고리즘 I과 탐색영역을 줄이는 TIE[5]를 결합하여 사용한다.

본 논문의 2장에서는 Wavelet/VQ 기법에 관하여 설명하며, 3장에서는 PDE 알고리즘에 대하여 논의하며 제안된 기법에 대하여는 4장에서 기술한다. 그리고 5장과 4장에서 각각 실험과 결론으로 끝을 맺는다.

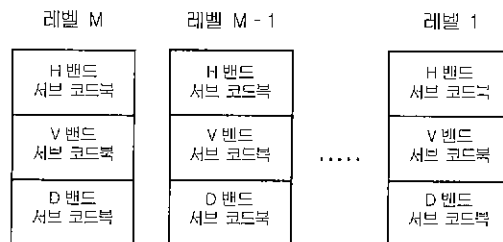
2. Wavelet/VQ 코딩 기법

DWT(Discrete Wavelet Transform)를 이용하면 (그림 1)에 나타난 바와 같이 하나의 영상을 다중해상도의 영상으로 분할할 수 있다[12]. 각 레벨은 3개의 서브밴드, 즉, H밴드(수평 방향 에지 영상), V밴드(수직 방향 에지 영상), 그리고 D밴드(대각선 방향 에지 영상)로 구성되며 마지막 레벨에서는 DC 영상이 존재하게 된다.



(그림 1) DWT를 이용하여 분할된 영상

Wavelet/VQ 기법에서는 각각의 서브밴드에 대해 별도의 서브 코드북을 사용하여 벡터 양자화를 수행하여야 한다. M개의 레벨로 웨이브렛 영상 분할을 수행하였다면 (그림 2)에서 보는 바와 같이 각 레벨에는 3개의 서브밴드(H, V, D밴드)가 존재하기 때문에 이에 해당하는 서브 코드북이 필요하다. 그러므로 전체 서브 코드북의 수는 $M \times 3$ 개가 된다.



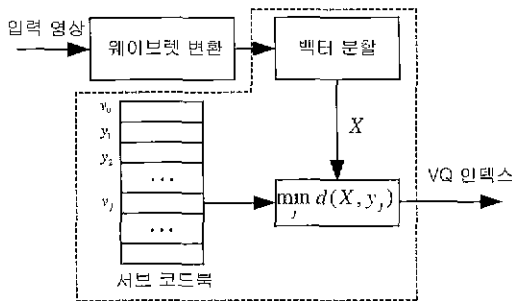
(그림 2) 서브 코드북

Wavelet/VQ 코딩 기법에서는 영상의 최소 값의 블록을 벡터 양자화의 입력으로 사용하는 대신에 DWT를 통하여 생성되는 웨이블릿 계수 블록이 벡터 양자화의 입력으로써 사용된다. Wavelet/VQ의 부호화 과정이 (그림 3)에 나타나 있다. 먼저 입력 영상을 웨이블릿 변환을 통하여 생성된 웨이블릿 계수들을 벡터로 분할하여 부호화 이전에 훈련집합을 통하여 생성된 코드북을 참조하여 최소 거리를 갖는 코드워드의 인덱스를 전송하게 된다.

(그림 3)에서 점선 블록은 벡터 양자화 부분을 나타낸다. 입력 벡터 $X = \{x_0, x_1, x_2, \dots, x_{K-1}\}$ 와 코드북의 코드워드 $y_i = \{y_{i0}, y_{i1}, y_{i2}, \dots, y_{i(K-1)}\}$ 사이의 거리는 다음과 같은 자승 유클리디안 거리(squared Euclidean distance)를 이용하여 구할 수 있다.

$$d(X, y_i) = \|X - y_i\| = \sum_{j=0}^{K-1} (x_j - y_{ij})^2 \quad (1)$$

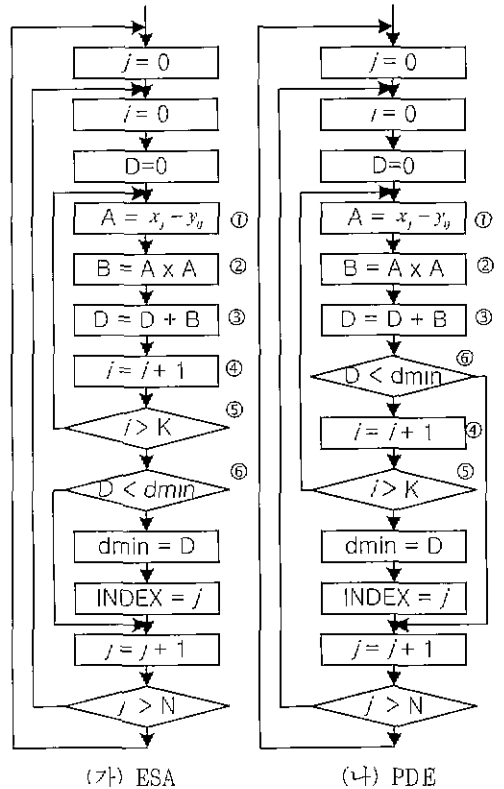
여기서 K는 벡터 차원이다. 이와 같이 벡터 양자화는 주어진 입력 벡터에 대해 최소의 거리를 갖는 코드워드를 찾는 과정이라고 말할 수 있다.



(그림 3) Wavelet/VQ 기법에서의 부호화 과정

3. PDE(Partial Distance Elimination) 알고리즘

Bey와 Gray에 의해 제안된 PDE 알고리즘[2]은 구현이 간단하고 성능이 우수하기 때문에 널리 사용된다. ESA(Exhaustive Search Algorithm)에서는 고속 탐색 알고리즘을 적용하지 않고 식 (1)에 있는 계산을 모든 코드워드에 대하여 수행하게 된다.



(그림 4) ESA와 PDE의 흐름도 비교(K=벡터 차원, N=코드북 크기)

(그림 4)에는 ESA와 PDE 알고리즘을 비교하기 위한 흐름도가 나타나 있다. (가)의 ESA 경우에는 식 (1)에 의해서 코드워드의 원소들이 모두 계산 ((가)에서 ①~⑤) 되고 나서 이서 현재까지의 최소 거리 dmin과 계산된 D를 비교한다. 따라서 벡터 차원과 같은 K개의 곱셈과 2×K-1개의 덧셈이 필요하게 된다. 그러나, (나)의 PDE 경우에는 제1번 D가 계산((나)에서 ①~③) 되고 나서 ⑥에서 dmin과 계산된 D를 비교한다. 비교한 결과 D가 dmin을 초과할 경우에는 나머지를 계산하지 않고 다음의 코드워드로 넘어가게 된다. 이렇게 함으로써 계산량이 현저하게 줄어들게 된다.

4. 제안된 고속 코드북 탐색 알고리즘

(그림 5)에 보여진 바와 같이 입력 벡터(웨이브릿 계수 블록)와 코드워드의 예를 통하여 제안된 기법을 설명한다. (그림 5)에서 벡터 차원 K는 4이다. 그리고

y_{i-1} 까지의 최소 거리 $d_{min} = 45.5$ 로 가정한다.

	x_0	x_1	x_2	x_3
X	2.7	-3.3	1.7	0.2

		y_{i0}	y_{i1}	y_{i2}	y_{i3}
y_i	0.8	3.5	-2.8	10.4	

(가) 입력 벡터 (나) 코드북
(그림 5) 입력 벡터와 코드북의 코드워드 예

ESA의 경우에는 식 (1)을 이용하여 거리 D 를 계산한다. 이 경우에는 (그림 4(가))의 ①~⑤과정이 4번 ($K=4$ 이기 때문) 수행된다.

$$D = (x_0 - y_{i0})^2 + (x_1 - y_{i1})^2 + (x_2 - y_{i2})^2 + (x_3 - y_{i3})^2$$

$$= (2.7 - 0.8)^2 + (-3.3 - 3.5)^2 + (1.7 - (-2.8))^2 + (0.2 - 10.4)^2 = 175.05 \quad (2)$$

이 경우에는 식 (2)에서 4번의 곱셈과 7번의 덧셈이 필요하게 된다.

한편, PDE를 이용하는 경우의 거리 D 는(그림 4(나))의 ①~③을 통한 첫 번째 루프에서

$$D = (x_0 - y_{i0})^2 = 3.61 \quad (3)$$

이 된다. 이때의 D 값이 60.5를 초과하지 못하기 때문에 두 번째 루프에서의 D 값은

$$D = 3.61 + (x_1 - y_{i1})^2 = 49.85 \quad (4)$$

이 된다. 식 (4)에서 D 값 49.85은 $d_{min}(45.5)$ 을 초과하기 때문에 현재 코드워드는 대상에서 제외되므로 더 이상의 계산이 필요치 않게 된다. 따라서 식 (3), 식 (4)에서 2번의 곱셈과 3번의 덧셈이 필요하게 된다.

4.1 알고리즘 I

PDE의 탈출 조건을 조기에 만족시키게 하기 위하여 코드워드의 큰 값을 먼저 입력 벡터와 계산한다. 이를 위하여 코드워드를 그 원소의 크기 순서로 정렬한 인덱스를 저장하는 테이블을 이용한다.

(그림 5(나))의 코드북의 코드워드 예에 대한 순서 인덱스 테이블은 (그림 6)에 나타나 있다.

					\vdots
	t_{i0}	t_{i1}	t_{i2}	t_{i3}	
t_i	3	1	2	0	
					\vdots

(그림 6) 순서 인덱스 테이블

테이블의 워드 $t_i = (t_{i0}, t_{i1}, t_{i2}, t_{i3})$ 는 코드워드 y_i 의 원소 y_{ij} 의 절대값 $|y_{ij}|$ 의 크기 순으로 된 인덱스 값을 가지게 된다. (그림 6)의 예에서 $|y_{i3}| > |y_{i1}| > |y_{i2}| > |y_{i0}|$ 이기 때문에 t_i 는 {3,1,2,0}이 된다.

식 (1)의 자승 유클리디안 거리는 다음 식과 같이 바뀌게 된다

$$d(X, y_i) = \|X - y_i\| = \sum_{j=0}^{K-1} (x_{t_j} - y_{i,t_j})^2 \quad (5)$$

이 경우 첫 번째 $j=0$ 일 때 $t_0=3$ 이기 때문에

$$D = (x_{t_0} - y_{i,t_0})^2 = (x_3 - y_{i3})^2 = 104.04 \quad (6)$$

가 된다. 이 때의 D 값 104.04는 $d_{min}(45.5)$ 을 초과하기 때문에 코드워드 y_i 에 대한 나머지 계산은 필요 없게 된다. 따라서 이 경우에는 1번의 곱셈과 1번의 덧셈만이 필요하다. 식 (3), 식 (4)의 PDE와 비교하여 곱셈이 1/2, 덧셈이 1/3로 줄어들게 된다

알고리즘 I에서 각 서브밴드에 대해 인덱스 테이블을 저장하는 데 사용되는 메모리 크기는 $N \times K \times \log_2 K$ (N : 코드북 크기, K : 벡터 차원)이다

4.2 알고리즘 II

알고리즘 II는 알고리즘 I과 탐색 영역을 줄이는 TIE알고리즘을 결합시킨 것이다. 사용된 TIE 알고리즘은 다음과 같은 식으로 나타낼 수 있는 [5]의 Algorithm II를 기본으로 한다

$$R = \{y_j : \|X - c\| - d_{min} \leq \|y_j - c\| \leq \|X - c\| + d_{max}\} \quad (7)$$

여기서, c 는 제어 벡터, d_{min} 는 $\|X - y_{min}\|$, y_{min} 는 코드북 탐색 이전에 최소 거리를 갖는 초기 코드워드를 나타낸다. 제어 벡터로써 모든 원소가 0으로 구성

된 등방향(isotropic) 벡터 $c=(0,0,0, \dots, 0)$ 를 사용한다. 그러므로 식 (7)은 다음과 같이 간략화될 수 있다.

$$R=\{y, \cdot \|X\| - d_{min} \leq \|y, \cdot\| \leq \|X\| + d_{min} \quad (8)$$

알고리즘 II는 코드북 탐색을 시작하기 전에 시행하는 전처리 단계와 실제 코드북을 탐색하는 온라인(On-line) 처리 단계의 2단계로 구성된다

<전처리 단계>

가) TIE를 위한 전처리

- i) 코드북의 코드워드의 놈 $\|y_0\|, \|y_1\|, \dots, \|y_{N-1}\|$ 을 계산하여 오름차순으로 정렬된 테이블 T_0 을 생성한다.
- ii) 테이블 T_0 를 이용하여 코드북을 정렬한다.

나) 순서 인덱스 테이블 생성

가)에서 정렬된 코드북의 코드워드 y_0 에서 $|y_w|$ 의 오름차순으로 된 인덱스 워드 i_0 를 구한다 같은 방법으로 y_1 으로부터 t_1, \dots, y_{N-1} 로부터 t_{N-1} 을 얻는다. 이와 같이 하여 2차원 순서 인덱스 테이블 T_1 을 생성한다.

<온라인 처리 단계>

- i) 식 (8)의 초기 코드워드 y_{min} 를 이전 벡터에서 구한 코드워드로 설정하여 d_{min} 를 구한다.
- ii) 입력 벡터 X의 놈 $\|X\|$ 을 구하여 테이블 T_0 의 내용물과 비교하여 가장 가까운 위치를 찾아 코드북 탐색의 시작점 S를 구한다. 그리고 시작점을 중심으로 위 아래로 빈갈아 가면서 탐색을 시작한다
- iii) 식 (8)을 이용하여 탐색 영역을 결정한다.
- iv) 만약 모든 입력 벡터를 전부 처리하였으면 vii)로 간다.
- v) 만약 현재의 코드워드 y_i 가 탐색 영역 내에 없으면 현재까지의 d_{min} 을 가지는 코드워드의 인덱스를 저장하고 다음 입력 벡터를 읽어 들이고 i)로 간다.
- vi) 테이블 T_1 을 이용하여 D값을 계산한 다음 d_{min} 과 비교하여 작으면 $d_{min} = D$ 를 설정한다. 다음 코드워드를 읽어들이고 iv)로 간다
- vii) 하나의 서브 코드북의 모든 입력벡터에 대한 코드북 탐색을 끝낸다. i)~vi)의 과정이 (그림 2)

의 모든 서브 코드북에 대한 코드북 탐색이 이루어져야 한다

5. 실험

실험에는 펜티엄 III PC(600MHz)를 사용하였으며 C++ 언어를 사용하여 윈도우 환경에서 시뮬레이션을 수행하였다. 9/7 랩 쌍직교(biorthonormal) 웨이브렛 필터[12]를 이용하여 입력 영상을 3개의 레벨로 웨이브렛 분할(wavelet decomposition)을 수행하였다. (그림 2)에 있는 서브코드북들은 널리 사용되는 LBG(Linde-Buzo-Gray) 알고리즘[13]을 이용하였으며 훈련 집합(training set)으로써 다양한 그레이 레벨 분포를 가지는 10장의 영상(512×512, 256 그레이 레벨)을 이용하였다.

제안된 고속 코드북 탐색 알고리즘의 성능을 검증하기 위하여 실험 영상으로써 512×512, 256 레벨의 10장의 영상(Lena, Barbara, Couple, Harbor, Peppers, Bridge, Goldhill, Man, Woman1, Woman2)을 사용하였다. <표 1>은 PDE[2], TIE([5]의 Algorithm II), PDE + TIE등과 제안된 알고리즘(알고리즘 I과 II)에 대해 화소당 소요되는 연산 수를 비교하였다 여기서 사용된 코드북 크기는 256이며 레벨3, 레벨2, 레벨1에 대해 각각 2×2, 4×4, 8×8의 벡터 차원이 사용되었다.

<표 1>에서 나타나 있는 연산들 중 곱셈과 루트 연산은 덧셈과 비교 연산에 비해 많은 하드웨어 또는 소프트웨어 비용을 요구한다. <표 1>에서 루트 연산은 많이 사용되지 않았기 때문에 곱셈의 수가 코드북 탐색 시간에 지대한 영향을 미친다 <표 1>에서 알 수 있듯이 알고리즘 I은 곱셈의 수가 PDE에 비해 42%, TIE에 비해 40% 절약되었다 그리고 PDE + TIE와 비교하여서는 약간 나은 결과를 나타내었다. 알고리즘 II는 알고리즘 I 보다 곱셈의 수가 22%정도 감소하였다. 참고로 (그림 7)과 (그림 8)에는 <표 1>에서 사용된 Lena 영상과 Barbara 영상에 대해 원영상과 벡터 양자화 후 복호화된 영상을 나타낸다. 여기서 부호화에 사용된 비트율은 0.4bpp(bit per pixel)에서 각각 30.47dB와 28.35dB의 PSNR 값을 얻었다.

<표 2>는 <표 1>과 같으나 단지 벡터 차원을 모든 서브코드북에 대해 4×4로 하였을 때의 결과이다. <표 2>에서도 제안된 알고리즘 I과 II가 다른 알고리즘 보다 우수함을 알 수 있다 <표 3>과 <표 4>에서는 각각 <표 1>과 <표 2>의 실험을 10장의 영상에 대하여 소요되는 화소당 평균 연산 수의 비교를 나타낸다.

<표 1> 각 알고리즘들에 대한 화소당 소요되는 연산수의 비교(코드북 크기 : 256, 벡터 차원 : 2x2, 4x4, 8x8)

Lena 영상				
알고리즘	×	+	<	√
PDE	50.1	91.6	50.4	-
TIE	48.3	94.1	4.6	0.2
PDE+TIE	31.1	59.8	32.4	0.2
I	29.2	37.9	29.2	-
II	22.7	42.9	23.9	0.2
Barbara 영상				
알고리즘	×	+	<	√
PDE	59.7	110.3	59.7	-
TIE	60.2	117.6	5.7	0.2
PDE+TIE	36.9	71.0	38.8	0.2
I	34.3	59.6	34.3	-
II	27.2	51.5	29.1	0.2

* ESA의 경우에는 × : 256.0, + : 502.8, < : 9.1이다

<표 2> 각 알고리즘들에 대한 화소당 소요되는 연산수의 비교(코드북 크기 : 256, 벡터 차원 : 4x4, 4x4, 4x4)

Lena 영상				
알고리즘	×	+	<	√
PDE	56.8	97.7	56.8	-
TIE	52.4	100.6	9.9	0.3
PDE+TIE	29.3	54.2	34.1	0.3
I	27.1	42.2	27.1	-
II	20.3	36.4	25.1	0.3
Barbara 영상				
알고리즘	×	+	<	√
PDE	66.5	117.0	66.5	-
TIE	64.4	123.8	12.2	0.3
PDE+TIE	33.9	62.9	40.2	0.3
I	30.5	63.0	30.5	-
II	23.8	42.7	30.1	0.3

ESA의 경우에는 × : 256.0, + : 496.0, < : 16.0이다.

<표 3> 10장의 영상에 대하여 화소당 소요되는 평균 연산수의 비교(코드북 크기 : 256, 벡터 차원 : 2x2, 4x4, 8x8)

알고리즘	×	+	<	√
PDE	65.1	116.5	65.1	-
TIE	72.8	140.6	12.2	0.3
PDE+TIE	41.3	77.6	47.7	0.3
I	37.0	57.9	37.0	-
II	29.3	53.6	35.6	0.3

<표 4> 10장의 영상에 대하여 화소당 소요되는 평균 연산수의 비교(코드북 크기 : 256, 벡터 차원 : 4x4, 4x4, 4x4)

알고리즘	×	+	<	√
PDE	69.6	127.8	69.6	-
TIE	76.2	148.4	9.1	0.3
PDE+TIE	45.5	87.1	49.7	0.3
I	40.2	71.0	40.2	-
II	32.2	64.4	36.7	0.3



(가) 원 영상



(나) 복원된 영상

(그림 7) <표 1>의 Lena 영상에 대한 원 영상과 복원된 영상(0.4bpp, PSNR : 30.47dB)



(가) 원 영상



(나) 복원된 영상

(그림 8) <표 1>의 Barbara 영상에 대한 원 영상과 복원된 영상(0.4bpp, PSNR : 28.35dB)

6. 결 론

본 논문에서는 Wavelet/VQ 압축 기법에서 부호화 시 사용되는 VQ 코드북 탐색 과정에서 탐색 시간을 줄이는 두 가지 알고리즘을 제안하였다. 알고리즘 I은 웨이블릿 영역에서 코드북을 계정렬한 인덱스 테이블을 이용하여 코드워드에서 매칭되는 원소들의 순서를 바꿈으로써 PDE에서 사용되는 탈출조건을 조기에 만족시키게 한다. 제안된 알고리즘 I은 PDE에서 필요로 하지 않는 인덱스 테이블에 해당하는 메모리를 요구하지만 VQ 코드북 탐색 시간을 줄임으로써 부호화 시간을 단축시킬 수 있다. 그리고 알고리즘 II에서는 알고리즘 I과 탐색 영역을 줄이는 TIE 알고리즘을 결합하여 사용함으로써 한층 더 코드북 탐색 시간을 줄일 수 있었다.

향후 이러한 고속 VQ 코드북 탐색 알고리즘을 이용하여 고화질을 유지하면서 압축 효율을 높이는 Wavelet/VQ 코딩 기법에 관한 연구가 필요하다고 하겠다.

참 고 문 헌

[1] P. C. Cosman, R. M. Gray, M. V. Vetterli, "Vector Quantization of Image Subbands : A Survey," IEEE Trans. On Image Process., Vol.5, No.2, Feb 1996.
 [2] C. D. Bei and R. M. Gray, "An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization," IEEE Trans Commun., Vol.COM-33, No.10, pp.1132-1133, Oct 1985
 [3] S. H. Huang and S. H. Chen, "Fast Encoding Algorithm for VQ-Based Image Coding," Electronics Letters. Vol.26, No.19, Sep 1990.
 [4] W. Li and E. Salar, "A Fast Vector Quantization Encoding Method for Image Compression," IEEE Trans. On Cir. & Sys For Video Tech, Vol.5, No.2, Apr 1995
 [5] C. M. Huang, Q. Bi, G. S. Stiles, and R. W. Harris, "Fast Full Search Equivalent Encoding Algorithms for Image Compression Using Vector Quantization," IEEE Trans On Image Process, Vol.1, No.3, July 1992
 [6] C. H. Hsieh and Y. J. Liu, "Fast Search Algorithms for Vector Quantization Images Using Multiple Triangle Inequalities and Wavelet Transform," IEEE Trans On Image Process Vol.9, No.3, Mar 2000
 [7] S. W. Ra and J. K. Kim, "A Fast Mean-Distance-

Ordered Partial Codebook Search Algorithm for Image Vector Quantization," IEEE Trans. Cir. & Sys.-II : Anal. & Dig. Sig. Process., Vol.40, No.9, pp.576-579, Sep 1993
 [8] M. R. Soleymani and S. D. Morgera, "A Fast MMSE Encoding Technique for Vector Quantization," IEEE Trans. On Comm., Vol.37, No.6, pp.656-659, June 1989
 [9] G. Poggi, "Fast Algorithm for Full-Search VQ Encoding," Electronics Letters. Vol.29, No.12, June 1993.
 [10] L. Jo and M. R. Kamal, "A Fast Second-Generation Encoding Algorithm for Vector Quantization," IEEE Signal Processing Letters, Vol.6, No.11, Nov 1999
 [11] C. H. Lee and L. H. Chen, "A Fast Search Algorithm for Vector Quantization Using Mean Pyramids of Codewords," IEEE Trans. On Commu., Vol.43, No.2, Apr 1995.
 [12] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image Coding using Wavelet Transform," IEEE Trans Image Processing, Vol.1, No.2, pp.205-220, Apr. 1992.
 [13] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design," IEEE, Trans. Commun., Vol.COM-28, pp.84-95 Jan 1980

황 재 호



e-mail : mjrine@unitel.co.kr
 1986년 경희대학교 전자공학과 졸업(학사)
 1991년 경희대학교 전자공학과 (공학박사)
 2000년 경희대학교 전자공학과 (박사수료)

1991년~1998년 한국통신 전송기술연구소 전임 연구원
 1998년~현재 극동정보대학 멀티미디어과 전임강사
 관심분야 : 영상 및 비디오 압축, 인터넷 서비스, 객체 지향 시스템 등

곽 윤 식



e-mail : yskwak@gukwon.chungju.ac.kr
 1984년 청주대학교 전자공학과 (공학사)
 1986년 경희대학교 전자공학과 (공학석사)
 1994년 경희대학교 전자공학과 (공학박사)

1991년~현재 충주대학교 컴퓨터공학과 부교수
 관심분야 : 디지털 신호처리



홍 충 선

e-mail : cshong@nms.kyunghee.ac.kr
1983년 경희대학교 전자공학과
졸업(학사)
1985년 경희대학교 전자공학과
(공학석사)
1997년~현재 경희대학교 전자정
보학부 전임강사

관심분야 : 인터넷 서비스 및 망 관리 구조, 분산컴포
넌트워크, IP 멀티캐스트, 멀티미디어스트리
밍 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