

# DCT와 블록 계층 분할 유한상태 벡터 양자화를 이용한 영상 부호화

조 성 환<sup>†</sup> · 김 응 성<sup>††</sup>

요 약

본 논문에서는 DCT를 이용하여 영상의 블록들을 계층적으로 분할하고, 각 블록에 대해 유한상태 벡터 양자화를 수행하는 알고리즘을 제안한다. DCT의 계수 특징을 이용하여 영상을 크기가 큰 매직 블록과 크기가 작은 윤곽선 블록으로 계층적으로 분할하고, 블록 계층 분할 정보를 전송한다. 각 계층에 속한 블록들에 대해 따로 부호화를 설계하고 부호 비트 수를 줄이기 위해 유한상태 벡터 양자화를 이용하여 부호단어의 인덱스를 계층 분할 정보와 함께 전송한다. Zelda와 Boat 영상에 대하여 본 알고리즘으로 영상을 부호화 했을 때 SMVQ 알고리즘보다는 각각 1.97 dB과 2.85 dB만큼, HFSVQ 알고리즘보다는 각각 약 1.78 dB과 1.85 dB만큼 더 좋은 영상의 화질을 얻을 수 있었다.

## Image Coding Using DCT and Block Hierarchical Segmentation Finite-State Vector Quantization

Seong-Hwan Cho<sup>†</sup> · Eung-Sung Kim<sup>††</sup>

ABSTRACT

In this paper, we propose an algorithm which segments hierarchically blocks of image using discrete cosine transform(DCT) and executes finite-state vector quantization(FSVQ) for each block. Using DCT coefficient feature, image is segmented hierarchically to large smooth block and small edge block, then the block hierarchy informations are transmitted. The codebooks are respectively constructed for each hierarchical blocks, the encoder transmits codeword index using FSVQ for reducing encoded bit with hierarchical segmentation information. Compared with side match VQ(SMVQ) and hierarchical FSVQ(HFSVQ) algorithm, about Zelda and Boat image, the new algorithm shows better picture quality with 1.97 dB and 2.85 dB differences as to SMVQ, 1.78 dB and 1.85 dB differences as to HFSVQ respectively.

### 1. 서 론

벡터 양자화(Vector Quantization : VQ)[1]는 적은 비트율, 즉 영상의 화소당 1 비트 이하로 영상을 부호화하는 효과적인 공간 영역 영상 부호화 방법이다. 간단한 벡터 양자화는  $k$  차원의 유클리디언 공간  $R^k$ 의

$Q$ 를  $R^k$ 의 유한 부분 집합  $Y$ 로의 맵핑(mapping)으로 정의할 수 있다. 즉  $Q: R^k \rightarrow Y$ 로 쓸 수 있는데  $Y = \{y_i, i = 1, 2, \dots, N\}$ 는 제구성 벡터 집합이고  $N$ 은  $Y$ 에서 벡터의 수이다. 이때 부분 집합  $Y$ 를 부호책(codebook)이라 부른다.  $k$  화소들의 하나의 영상 블록  $X$ 는 부호책에 있는 부호단어(codeword)라고 하는 대표 벡터에 의해 표현된다. 단지 부호단어의 주소만이 전송되거나 저장되기 때문에 부호책 크기가  $N$ 인

<sup>†</sup> 정 희 원 : 동서울대학 전자계산학과 교수  
<sup>††</sup> 정 희 원 : 경기공업대학 컴퓨터응용과 교수  
논문집수 : 1999년 3월 13일, 심사완료 : 1999년 12월 1일

VQ를 이용한 영상 부호화의 비트율은  $(\log_2 N)/k$ 로 쓸 수 있다.

복원 영상의 화질을 좋게 하고 효과적인 압축 기법을 위해서 VQ에 대한 여러 가지 알고리즘들이 개발되고 있다[1-3]. 이들 알고리즘에는 블록의 크기를 변화시켜 각각에 대해 양자화를 수행하는 가변 블록 벡터 양자화[4]나 다른 화소 값 변화의 특징을 가지는 블록을 다른 계층(layer)에 할당시키는 계층적 다중비율 벡터 양자화(Hierarchical Multirate VQ H MVQ)[5] 등이 있는데, 이들 알고리즘에서는 영상의 특징들을 강조함으로써  $(\log_2 N)/k$ 의  $k$  값을 증가시켜 비트율을 감소시킨다. 또한 부호단어에 할당하는 비트를 감소시키는 알고리즘[6]도 연구되었는데, 즉, 대표 벡터 선택을 위한 부호책의 부호단어 수를 조절함으로써  $(\log_2 N)$  값을 감소시킨다.

또한 블록 경계를 가로지르는 이웃 화소들의 상관관계(correlation)를 이용함으로써 최적 대표 벡터를 찾기 위해 이전에 부호화된 블록들을 사용하는 유한상태 벡터 양자화(Finite-State Vector Quantization: FSVQ)[7, 8] 알고리즘이 있는데, 이와 같은 이웃 블록간의 상관관계를 이용함으로써 일반적인 벡터 양자화기로 가능한 것 이하로 비트율을 감소시킬 수 있다.

Yu와 Venetsanopoulos는 FSVQ를 이용하여 영상 부호화를 수행하는 벡터 양자화(Hierarchy FSVQ, HFSVQ) 알고리즘을 제안하였다[9]. 이 알고리즘에서는 영상을 부분블록(subblock)으로 나누어 각 블록의 최대 화소 값 차이를 계산하여 각각의 블록을 여러 계층으로 할당한 후, 각 계층에 대해 FSVQ를 적용한다. 그러나 이 알고리즘에서는 한 블록의 특성에 대한 특징 파라미터로서 화소 값의 차이만을 사용하므로 거의 배경에 가까운 블록도 윤곽선(edge) 영역으로 판별하여 전체 복원 영상의 화질을 열화시키는 문제가 발생한다.

본 논문에서는 영상의 특징 판별을 확실하게 하고 전송 비트율을 낮추기 위하여 DCT(Discrete Cosine Transform)를 사용하고 영상을 여러 블록 크기를 갖는 계층으로 분할하여 유한상태 벡터 양자화를 수행하는 알고리즘을 제안한다. 본 논문의 알고리즘에서 원 영상은 DCT를 이용한 블록 분할 작업에 의해 각각 다른 크기를 갖는 블록들로 분해하고, 서로 다른 계층으로 할당한다. 영상의 배경 영

역에 해당하는 블록은 크기가 더 크며, 높은 계층으로 할당한다. 화소 값의 변화가 큰 윤곽선 영역에 위치한 블록들에 대해서는 그에 해당하는 주부호책(master codebook)의 모든 부호단어들이 대표 벡터 선택을 위해 사용된다. 제안 알고리즘을 이용한 영상 부호화의 부호는 두 부분으로 구성되는데, 영상 블록의 계층 할당에 대한 정보를 제공하는 계층 분할 정보 부호와 부호책에서의 부호단어 인덱스 부호로 구성된다. 본 논문의 제안 알고리즘은 DCT를 사용하여 각 블록의 특징을 추출함으로써 영상의 블록 계층 분할을 확실하게 할 수 있고, FSVQ를 사용함으로써 벡터 양자화의 부호단어 인덱스 전송 비트 수를 적게 유지시킬 수 있다. 시뮬레이션 결과 본 논문에서 제안한 방법으로 영상 부호화를 했을 경우 기존의 벡터 양자화 기법인 SMVQ[8]나 HFSVQ[9]와 비교하여 더 좋은 화질의 영상을 얻을 수 있었다.

## 2. DCT를 이용한 블록 분할

벡터 양자화 기법은 일반적으로 윤곽선 근처에서 고주파수 양자화 오류를 일으키고, 배경 부분에서는 저주파수 양자화 오류를 발생시킨다. 날카로운 윤곽선 부분의 주파수 오류는 윤곽선을 무디게 만들며 이는 복원 영상의 화질에 큰 영향을 준다. 배경 영역에서의 왜곡(distortion)은 블록화 현상(blocking effect)이 없다면 인간의 눈에 명백하게 관찰되지 않는다.

본 논문에서는 이와 같은 일반적인 벡터 양자화 기법에서 발생하는 오류를 최소화하기 위해 블록들을 계층적으로 분할하고, 각 계층에 속하는 블록들에 따라 벡터 양자화를 수행하는 알고리즘을 제안한다. 영상의 각 블록에 대해 그 블록이 어느 계층에 속하는지 여부를 결정하기 위하여 2차원 DCT를 계산한다.

DCT를 이용한 각 블록의 계층 분할은 다음과 같은 3 단계로 수행한다.

[단계 1] 원 영상을 좌측상단으로부터 우측으로 16 화소, 하단으로 16 화소씩  $16 \times 16$ 의 부분블록(subblock)으로 나누어 2차원  $16 \times 16$  DCT를 계산한다. 계산 결과에 대해 (그림 1) (a)와 같이 DCT 계수 값을 중에서 좌측 상단의 현

제 블록의 DC 평균값인 ①의 값과 저주파수의 AC 값인 ②, ③, ④, ... ⑦의 값을 포함한 7개의 값은 그대로 유지하고 나머지 값은 0으로 만들어 2차원 DCT의 역변환을 취한다. 그 결과 값들과 원 영상의 해당 블록과의 평균 제곱 오차(Mean Square Error : MSE)를 계산하여 그 값이 문턱치(threshold)  $\delta_1$ 보다 작으면 현재  $16 \times 16$  블록을 계층  $L_1$ 에 할당하고,  $\delta_1$ 보다 크거나 같으면 현재 블록을 4개의  $8 \times 8$  블록으로 분할한다.

①	②	③	④		...	
⑤					...	
⑥					...	
⑦					...	
					...	
⋮	⋮	⋮	⋮	⋮	⋮	⋮
					...	

(a)

①	②	③				
④						
⑤						

(b)

①	②		
③			

(c)

(그림 1) 각 블록에서의 DCT 계수 값 선택  
(a)  $16 \times 16$  블록 (b)  $8 \times 8$  블록 (c)  $4 \times 4$  블록

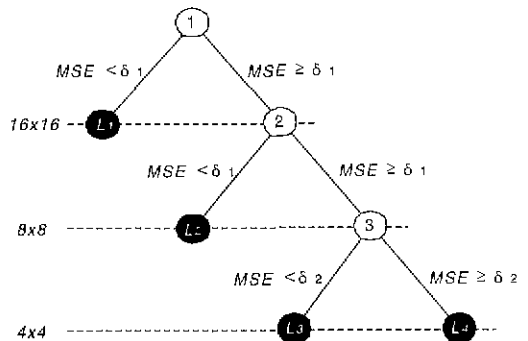
[단계 2]  $16 \times 16$  블록에서 분할된 각각의 4개의  $8 \times 8$  블록에 대하여 2차원  $8 \times 8$  DCT를 계산한다. 계산 결과에 대해 (그림 1) (b)와 같이 ①, ②, ③, ④, ⑤의 DCT 계수 값들은 그대로

유지하고 나머지 값은 0으로 만들어 2차원 DCT의 역변환을 취한다. 그 결과 값들과 원 영상의 해당 블록과의 평균 제곱 오차를 계산하여 그 값이 단계 1과 같은 문턱치  $\delta_1$ 보다 작으면 현재  $8 \times 8$  블록을 계층  $L_2$ 에 할당하고,  $\delta_1$ 보다 크거나 같으면 4개의  $4 \times 4$  블록으로 분할한다.

[단계 3] 분할된 4개의  $4 \times 4$  블록 각각에 대하여 2차원  $4 \times 4$  DCT를 계산한다. 그 결과 (그림 1) (c)와 같이 ①, ②, ③의 DCT 계수 값들은 그대로 유지하고 나머지 값은 0으로 만들어 2차원 DCT의 역변환을 취해 원 영상의 해당 블록과의 평균 제곱 오차를 계산하여 그 값이 문턱치  $\delta_2$ 보다 작으면 현재  $4 \times 4$  블록을 계층  $L_3$ 에 할당하고,  $\delta_2$ 보다 크거나 같으면 현재  $4 \times 4$  블록을 계층  $L_4$ 에 할당한다. 이 단계에서 새로운 문턱치  $\delta_2$ 를 사용하는 것은 계층  $L_3$ 과  $L_1$ 는 윤곽선 영역으로서 좀 더 정확히 분할되어야 하기 때문이다.  $\delta_2$ 는  $\delta_1$ 보다 더 큰 값을 가진다.

### 3. 계층 분할 정보 전송

DCT를 이용하여 각 블록을  $L_1, L_2, L_3, L_4$ 의 4개의 계층으로 분할하면, 각 계층에 속하는 블록들의 크기가  $L_3$ 와  $L_1$ 를 제외하곤 서로 다르게 된다. 2절에서 설명한 DCT를 이용한 영상의 블록 계층 분할은 (그림 2)와 같이 나타낼 수 있다. 즉, 트리(tree) 구조에서 1번 노드에 영상을  $16 \times 16$  크기의

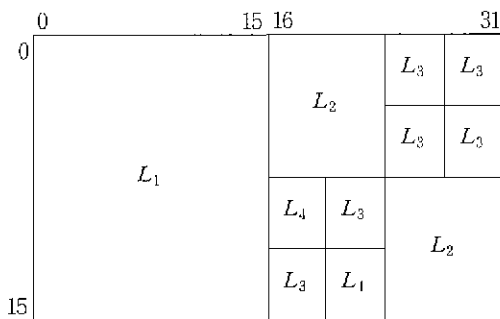


(그림 2) 블록의 계층 분할

블록들로 나눈 한 블록에 대해 DCT를 계산하여 원 영상의 해당 블록의 MSE를 계산하여 문턱치  $\delta_1$ 보다 작으면 왼쪽의 계층  $L_1$ 에 현재 블록이 할당되는 것이다. MSE가 문턱치  $\delta_1$ 보다 크거나 같을 경우는 현재  $16 \times 16$  블록을 2번 노드로 입력시켜 4개의  $8 \times 8$ 로 분할하여 어느 계층에 속하는지 결정한다.

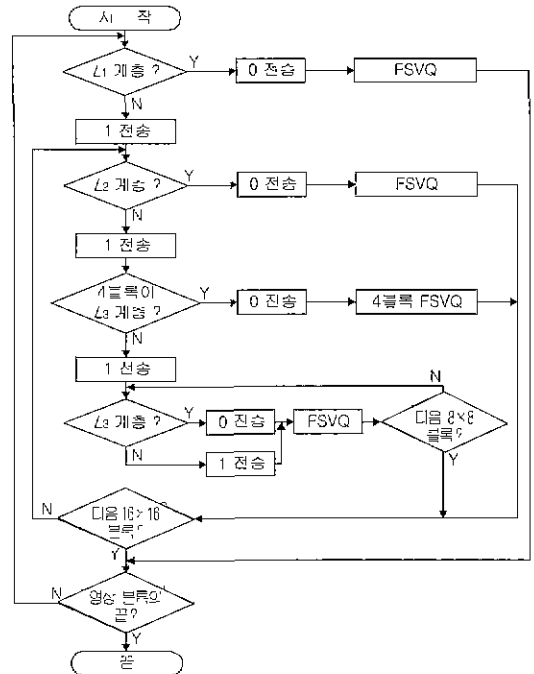
이와 같이 (그림 2)에서 영상의 각 블록은 4개의 계층  $L_1, L_2, L_3, L_4$  중 하나에 속하고, 각 계층에 속하는 블록들은  $L_3$ 와  $L_4$ 를 제외하고 서로 크기가 다르다 그 결과, 높은 계층으로 갈수록 배경 영역에 가까운 블록들을, 낮은 계층으로 갈수록 윤곽선과 같은 비교적 블록내의 화소 값 변화가 큰 블록들을 포함하게 된다 보다 더 정확한 윤곽선 영역의 부호화를 위해 같은  $4 \times 4$  블록에 대해서는  $L_3$ 와  $L_4$ 로 분할하는데, 계층  $L_4$ 는  $\delta_1$ 보다 더 큰 값을 가지는 문턱치  $\delta_2$ 를 사용하여 큰 화소 값 변화를 가지는 블록들을 할당한다.

영상의 각 블록에 대해 계층 분할을 한 다음, 각 블록에 대한 계층 할당 정보를 전송하는데, 이는 유한상태 벡터 양자화와 동시에 수행하는 것으로 블록 계층 분할에 대한 오버헤드 정보이다.



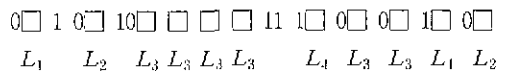
(그림 3) 계층 분할된 블록의 예

(그림 3)에는 두 개의  $16 \times 16$  블록에 대한 계층 분할된 블록들의 한 예가 나타나 있다. 그림과 같이 계층 할당된 각 블록에 대한 정보는 블록의 벡터 양자화 부호단어 인덱스 전에 전송한다. (그림 4)에는 계층 분할 정보를 전송하는 알고리즘이 나타나 있다.



(그림 4) 계층 분할 정보 전송 알고리즘

(그림 5)는 (그림 4)의 알고리즘을 사용하여 (그림 3)의 블록 계층 분할의 예에 대한 계층 분할 정보를 전송한 결과이다.



(그림 5) 그림 3의 계층 분할 정보

(그림 5)에서 □는 계층 분할 정보를 전송한 후에 해당 블록에 대한 유한상태 벡터 양자화를 수행한 후 전송하는 부호단어에 대한 인덱스 정보를 의미한다 그림에서  $16 \times 16$  크기의 두 블록에 대해 부가되는 계층 분할 정보량이 크게 증가하지 않는다는 것을 알 수 있다 기존의 벡터 양자화에 비해서는 이 정보가 오버헤드 정보로서 부가되지만 블록을 계층적으로 분할하였고, 전송 비트 수를 줄이기 위해 벡터 양자화에는 유한상태 벡터 양자화를 수행하기 때문에 전체 전송 비트율을 낮게 유지할 수 있다. 또한 (그림 5)에서 알 수 있듯이 낮은 계층에 해당하는 블록에 대해서는 더 많은 부호가 할당됨을 알 수 있다. 일반적으로 영상에서 높은

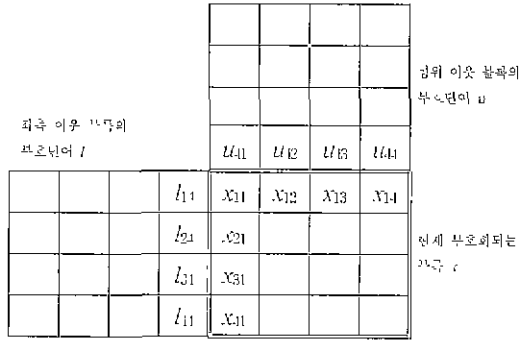
계층에 할당되는 배경 부분에 가까운 블록들의 수가 더 많기 때문에 계층 정보 전송량은 크게 증가하지 않는다.

#### 4. 계층적 유한상태 부호화

(그림 5)에서와 같이 각 블록에 대한 계층 분할 정보 전송 후에는 벡터 양자화의 결과인 부호단어 인덱스를 전송해야 한다. 본 논문에서는 각 블록의 부호화에 유한상태 벡터 양자화(FSVQ)를 사용하는데, 이 FSVQ는 이웃 블록들 사이의 상관관계를 이용하기 때문에 복원 영상의 화질을 개선하고 벡터 양자화기의 전송 비트율을 감소시킬 수 있으며, 이전 블록의 상태를 기억하여 부호화를 수행한다[7-11].

유한상태 벡터 양자화기는 일반적인 VQ의 유한한 집합, 즉 부호화기 상태에 따라서 자신의 부호책을 가지는 벡터 양자화기로 생각할 수 있다. 현재 입력 벡터의 상태는 이전에 부호화된 벡터에 의해 정의되며, 각각의 입력 벡터를 현재 부호화기 상태에 의하여 결정된 부호책을 사용하여 부호화한다. 유한상태 벡터 양자화기에는 주 부호책(master codebook)과 상태 부호책(state codebook)이 존재하는데,  $S = \{s_i : i = 1, \dots, M\}$ 를 상태 공간이라 할 때, 각 상태  $s_i$ 에 대하여  $s_i$ 의 상태 부호책으로서 주 부호책  $Y = \{y_i : i = 1, \dots, N_m\}$ 으로부터  $N_f$ 개의 부호단어를 선택한다. 이전에 부호화된 벡터의 부호단어를  $p$ 라 할 때, 상태 함수  $f(p)$ 가 현재 입력 벡터  $x$ 의 상태를 결정하기 위하여 사용된다. 입력 벡터  $x$ 를 부호화하기 위하여 부호화기는 현재 상태  $s = f(p)$ 를 찾고 그에 해당하는 부호단어를 찾기 위하여 주 부호책  $Y$ 가 아닌 상태 부호책  $SC_i$ 를 탐색한다. 일반적으로 상태 부호책의 부호단어 수인  $N_f$ 는 주 부호책의 부호단어 수  $N_m$ 보다 훨씬 적다. 현재 부호화하려는 블록에 대해 상태 부호책을 선택하고, 그 상태 부호책으로부터 가장 작은 왜곡을 가지는 부호 단어를 선택하여 그 인덱스를 복호화기로 전송한다. 본 논문에서 부호책은 계층별로 작성하기 때문에 각 계층은 자신의 부호책을 따로 가지고 있다. 각각의 주 부호책 작성 후에 상태 부호책의 설계는 영상의 이웃 블록들 사이의 상관관계를 이용하여 작성한다.

(그림 6)과 같이 현재 부호화되는 입력 벡터, 즉  $x$ 에 대한 이전에 부호화된 상위 블록과 좌측 블록과의 왜



(그림 6) 유한상태 벡터 양자화기의 부호화 (블록 크기 4×4)

곡(distortion)을 계산하여 상태부호책을 작성한다. 현재 입력 벡터  $x$ 의 상태 부호책  $SC_i$ , 선택은 각 상태 부호책의 대표 벡터  $p$ 와 현재 입력 벡터의 각각 이웃 열과 이웃 행을 따라서  $x$ 의 상위 블록  $u$ 와 좌측 블록  $l$ 에 가장 잘 부합되는 상태 부호책으로 한다. 즉, 크기  $m \times n$ 의 블록을 정의했을 때, 상태 부호책의 대표 벡터와 현재 입력 벡터의 상위 블록과의 수평 방향 왜곡  $hd(p, u)$ 를

$$hd(p, u) = \sum_{j=1}^n (p_j - u_{mj})^2 \quad (1)$$

라 하고, 수직 방향의 왜곡  $vd(p, l)$ 를

$$vd(p, l) = \sum_{i=1}^m (p_i - l_{in})^2 \quad (2)$$

라 할 때, 입력 벡터  $x$ 의 왜곡  $dist(x)$ 는 다음과 같이 정의한다.

$$dist(x) = hd(p, u) + vd(p, l) \quad (3)$$

여기 상태 부호책  $SC_i$ 들의 대표 벡터  $p$ 에 대하여 가장 작은 왜곡  $dist$ 를 가지는  $SC_i$ 를 현재 입력 벡터의 상태 부호책으로 선택한다. 선택된 상태 부호책의 각 부호단어들과 현재 블록의 화소 값들의 MSE를 계산하여 그 값이 가장 작은 부호단어의 인덱스를 부호화하여 복호화기로 전송한다. 이와 같이 현재 입력 벡터에 대한 상태 부호책의 선택에 이전 블록의 정보를 이용하기 때문에 복호화기에서 상태 부호책의 선택을 위한 추가 정보를 부호화기에서 전송할 필요가 없다. 이는 결국 큰 주 부호책에서 부호 단어를 선택하

여 그에 대한 인덱스를 전송하는 것에 비하여 작은 크기의 상태 부호책의 인덱스만을 전송하여 비트율을 줄이는 효과를 가져온다.

만일 현재 부호화 블록  $x$ 의 좌측 블록  $l$ 과 상위 블록  $u$ 가 같은 계층에 속하지 않는다면 블록의 크기도 다르다. 이 경우에 좌측 블록  $l$ 과 상위 블록  $u$ 는 현재 부호화 블록  $x$ 의 크기로 조정되어야 한다 예를 들어  $x$ 가 계층  $L_1$ 에 속하지만 그 이웃 블록들이 다른 계층에 속한다면,  $u$ 는 행의 화소 개수가 16개인  $x$  위쪽의 몇몇 블록들을 조합하고, 마찬가지로  $l$ 을 구성하기 위해서 열 방향으로 16개의 화소를 구성해야 한다 그러나  $u$ 와  $l$ 은 현재 블록  $x$ 에 대해 이전에 이미 부호화된 블록들이기 때문에 16개의 화소를 구성할 수 있다. 한편  $x$ 가 낮은 계층에 속한다면 이웃 블록들은 더 높은 계층에 속하게 된다. 즉, 이 경우  $x$ 는  $u$ 와  $l$ 의 크기보다 작은 블록이 되므로 식 (3)에서의 현재 블록에 대한 상태 부호책 선택 계산을 수행할 수 있다.

계층 분할 정보를 전송한 후에 해당 블록에 대한 부호단어의 인덱스 경보는 따로 부호화를 수행하지 않는다 즉, 계층  $L_1$ 과  $L_2$ ,  $L_3$ 에 대한 상태 부호책의 크기가  $M$ 이라고 하면, 이 계층들에 속한 블록에 대한 부호단어 인덱스는  $(\log_2 M)$ 비트를,  $L_1$ 에 대한 주 부호책 크기를  $N_m$ 이라고 하면  $(\log_2 N_m)$ 비트를 전송한다.

## 5. 실험 결과

제안 알고리즘의 평가는  $512 \times 512$  크기의 Zelda와 Boat 영상에 대해 수행하였다. Zelda 영상은 비교적 윤곽선이 적고, Boat 영상은 전체적으로 윤곽선 부분이 많은 영상으로, 윤곽선의 양에 의한 시뮬레이션 결과를 비교하기 위해서 두 영상을 선택하였다. 제안 알고리즘과 기존 알고리즘을 비교하기 위해 SMVQ[8]와 HITSVQ[9]를 실험 영상에 적용하였고, 객관적 평가를 위해 복원 영상에 식 (4)의 PSNR을 계산하였다.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{ [dB]} \quad (4)$$

SMVQ에서 주 부호책의 크기는 256, 상태 부호책의 크기는 16으로 하였고, 제안 알고리즘에서 계층

$L_1$ 과  $L_2$ ,  $L_3$ 에 대한 주 부호책 크기는 256, 상태 부호책 크기는 16으로 하였고,  $L_1$ 에 대한 주 부호책 크기는 64로 하였다. 계층  $L_4$ 에 할당된 블록의 부호화에는 블록 경계에 확실한 윤곽선이 존재하는 경우이므로 복원 영상의 화질을 더 좋게 하기 위해 벡터 양자화에 상태 부호책이 아닌 주 부호책을 사용하였다. 블록의 계층 할당을 위한 문턱치  $\delta_1$ 은 30으로,  $\delta_2$ 는 200으로 하였고, 이 수치는 실험적으로 결정하였다 각 알고리즘에서 주 부호책은 LBG 알고리즘[12]을 사용하여 설계하였다.

(그림 7)에는 Zelda 영상과 Boat 영상에 대한 블록들의 계층 분할 결과가 나타나 있는데, 그림에서 밝을수록 높은 계층, 즉  $L_1$  계층에 가까운 것이고, 어두울수록  $L_4$  계층으로 할당된 블록들을 나타낸다.



(a)



(b)

(그림 7) 블록 계층 분할 결과  
(a) Zelda 영상 (b) Boat 영상

(그림 7)을 보면 Zelda 영상의 경우, 얼굴 중에서 머리 부분에 낮은 계층인 윤곽선 블록으로 분할된

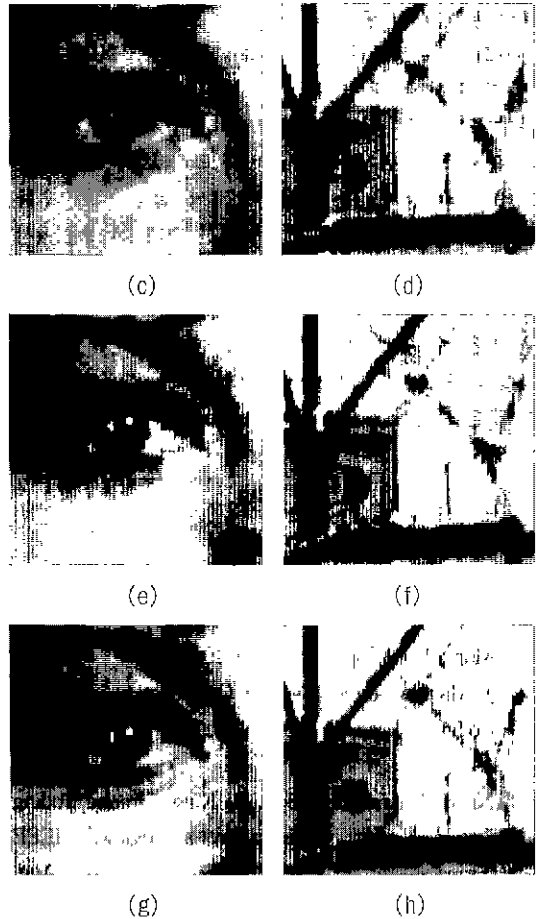
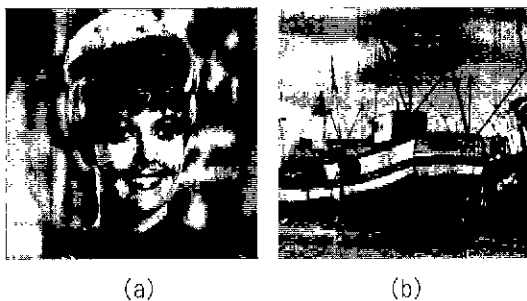
블록이 많다는 것을 알 수 있고, Boat 영상은 Zelda 영상보다 윤곽선이 많기 때문에 전체적으로 낮은 계층 블록들이 많이 분포하고 있다는 것을 알 수 있다.

<표 1>은 기존 알고리즘인 SMVQ와 HFSVQ, 그리고 제안 알고리즘의 두 영상에 대한 실험 결과표식 (4)를 이용하여 PSNR 값을 계산한 것이다. 표에서 알 수 있듯이 Zelda 영상은 기존의 유한상태 벡터 양자화 기법인 SMVQ에 비하여 1.97 dB, HFSVQ에 비하여 1.78 dB만큼 화질이 더 좋았으며, Boat 영상에 대해서는 SMVQ에 비하여 2.85 dB, HFSVQ는 1.85 dB만큼 복원 영상의 화질이 좋았다. 두 영상에 대해 SMVQ와 HFSVQ보다 제안 알고리즘의 비트율이 낮았고, 전송 비트 수가 더 적은 상황에서 결과 PSNR이 제안 알고리즘이 더 좋았다 이로써 본 논문에서 제안한 알고리즘이 더 좋은 성능을 나타냄을 알 수 있다.

<표 1> SMVQ, HFSVQ와 제안 알고리즘의 결과 PSNR(dB) 비교

	SMVQ	HFSVQ	제안 알고리즘
Zelda 영상	32.05 (0.25 bpp)	32.24 (0.12 bpp)	34.02 (0.10 bpp)
Boat 영상	27.62 (0.25 bpp)	28.62 (0.18 bpp)	30.47 (0.17 bpp)

(그림 8)에는 Zelda와 Boat 영상에 대한 SMVQ 알고리즘과 HFSVQ 알고리즘, 그리고 본 논문에서 제안한 알고리즘의 <표 1>에서의 시뮬레이션 결과 영상이 나타나 있다 (a)와 (b)는 원 영상이고, 그 아래는 각 알고리즘간의 비교를 확실히 하기 위해 영상의 한 부분을 확대한 그림이다. 그림에서 각각의 SMVQ, HFSVQ 결과 영상과 제안 알고리즘의 결과 영상을 비교해 보면 영상의 윤곽선(edge) 영역에서 다른 알고리즘보다 본 논문의 제안 알고리즘이 더 적게 왜곡이 일어난 것을 볼 수 있다.



(그림 8) "Zelda"와 "Boat" 영상에 대한 SMVQ, HFSVQ, 제안 알고리즘의 비교  
(a) "Zelda" 원 영상, (b) "Boat" 원 영상,  
(c),(d) SMVQ 결과 확대 영상, (e),(f) HFSVQ 결과 확대 영상  
(g),(h) 제안 알고리즘의 결과 확대 영상

## 6. 결 론

본 논문에서는 영상의 벡터 양자화를 위한 새로운 부호화 알고리즘을 제안하였다. 영상을 매경 영역과 윤곽선 영역으로 분할하여 각 영역에 대해 따로 벡터 양자화를 수행하기 위해서 DCT를 사용하였다. 영상을 16×16 크기의 블록으로 분할하여 DCT를 계산하고, 계수 값 생략 후 역 DCT를 취한다. 그 다음 원 영상과 왜곡을 계산하여 문턱치에 따라 다시 그 블록을 4 등분하여 하위 계층으로 분류한다. 이와 같이 4×4 크기가 될 때까지 계층

분할된 각각의 블록에 대해 따로 유한상태 벡터 양자화를 수행한다. 유한상태 벡터 양자화 수행시에 각 블록에 대한 계층 분할 정보도 함께 전송한다. 계층 분할 정보가 실제 벡터 양자화의 부호단어 인덱스 정보가 아닌 오버헤드 정보이지만 유한상태 벡터 양자화를 사용함으로써 전송 비트 수를 많이 줄일 수 있었다. 시뮬레이션 결과 본 논문에서 제안한 방법으로 영상 부호화를 수행했을 경우 SMVQ 나 HFSVQ 알고리즘과 비교하여 더 좋은 화질의 영상을 얻을 수 있었으며, 추후 연구 과제로는 블록 계층 분할에 사용하는 문턱치들을 실험적으로 결정하였는데, 이 값들을 영상에 따라 적응적으로 정해주는 알고리즘의 연구가 필요하다.

**참 고 문 헌**

[1] R. M. Gray, "Vector Quantization," IEEE ASSP Mag., Vol.1, pp.4-29, April 1984.  
 [2] J. Makhoul et al., "Vector Quantization in Speech Coding." Proc. IEEE, Vol.73, pp.1551-1588, Nov. 1985.  
 [3] N. M. Nasrabadi and R. B. King, "Image Coding Using Vector Quantization a review," IEEE Trans. Commun., Vol.COM-36, pp.957-971, Aug 1988.  
 [4] D. J. Vaisey and A. Gershot, "Variable Block-Size Image Coding," in Proc ICASSP, 1987, pp.1051-1054.  
 [5] P. Yu and A. N. Venetsanopoulos, "Hierarchical Multirate Vector Quantization for Image Coding," Signal Processing Image Commun., Vol.4, No.6, pp.497-505, Nov. 1992.  
 [6] A. Gersho and R. M. Gray, "Vector Quantization and Signal Compression," Boston : Kluwer Academic Publishers, 1992.  
 [7] J. Foster, R. M. Gray, and M. O Dunham, "Finite State Vector Quantization for Waveform Coding," IEEE Trans Inform. Theory, Vol.IT-31, pp.348-355, May 1985.  
 [8] T Kim, "New Finite State Vector Quantizers for Images," in Proc. ICASSP, 1988, pp 1180-1183.

[9] P. Yu and N. Venetsanopoulos, "Hierarchical Finite-State Vector Quantization for Image Coding," IEEE Trans Commun., Vol.42, No 11, pp.3020-3026, Nov 1994.  
 [10] R. L Baker and H. H. Shen, "A Finite-State Vector Quantizer for Low-rate Image Sequence Coding," in Proc ICASSP. 1987, pp 760-763.  
 [11] N. M. Nasrabadi and Y. Feng, "A Dynamic Finite-State Vector Quantization Scheme," in Proc. ICASSP, 1990, pp.2261-2264.  
 [12] 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.



**조 성 환**

e-mail : shcho@haksan.dsc.ac.kr  
 1980년 성균관대학교 전자공학과 졸업(학사)  
 1982년 성균관대학교 대학원 전자공학과(공학석사)  
 1991년 성균관대학교 대학원 전자공학과(공학박사)  
 1982년~1985년 해군사관학교 전기공학과 전임강사  
 1997년 미국 Columbia 대학 Visiting Scholar  
 1985년~현재 동서울대학 전자계산기와 부교수, 전자계산소장  
 관심분야 : 영상처리, 신경회로망, 패턴인식 등



**김 응 성**

e-mail : imagecap@chollian.net  
 1989년 성균관대학교 전자공학과 졸업(학사)  
 1992년 성균관대학교 대학원 전자공학과(공학석사)  
 1998년 성균관대학교 대학원 전자공학과(공학박사)  
 1995년~2000년 성균관대학교 과학기술연구소 연구원  
 2000년~현재 경기공업대학 컴퓨터응용과 전임강사  
 관심분야 : 영상처리, 영상통신, 신경회로망 등