

# 효율적인 Quadtree 분할을 위한 최적의 임계값 설정 기술

論 文  
48A-8-15

## The Optimal Thresholding Technique for an Efficient Quadtree Segmentation

李 恒 燦\*  
(Hang-Chan Lee)

**Abstract**— A Hierarchical vector Quantization scheme is implemented and an optimal thresholding technique of quadtree segmentation for performing high quality low bit rate image compression is proposed. A mathematical model is constructed under the assumption that the standard deviations of sub-blocks are larger than or equal to the standard deviation of the upper level block which is generated by merging of sub-blocks. This thresholding technique based on the mathematical modeling allows producing about 1 dB improved performance in terms of PSNR at most ranges of bit rates over the quadtree coder, which is based on MSE for quadtree segmentation.

**Key Words:** Vector Quantization, Quadtree, Compression, Threshold

### 1. 서 론

영상압축의 궁극적인 목표는 고품질의 압축영상을 출력하는 동시에 효율적인 데이터 저장이나 전송을 위해 비트율을 낮추는데 있다. 1948년에 Claude Shannon에 의해 확률이론에 기반하여 정보의 양을 측정하는 개념이 소개된[1] 이래로 많은 압축기술과 이에 대한 응용 기술들이 개발 되어 왔다. 실제로 디지털 영상에 대한 대부분의 응용 기술들은 효율적인 데이터 압축기술이 적용되지 않는 한 사용에 많은 제약이 따른다.

계산의 복잡성을 회피하고 고품질의 압축영상을 출력하기 위해 1980년 이래로 개발되어 온 벡터 정량화 기술은 LUT(Look Up Table)의 간단한 디코딩 과정을 갖는 반면 Transform Coding에 견줄만한 화질의 압축영상을 출력한다[2]. 이러한 특질때문에 벡터 정량화는 효과적인 데이터 압축 방법으로 연구되어왔다. 패턴인식의 한 형태로 간주되어온 벡터 정량화 기술에서, 입력 패턴은 미리 설정된 표준 패턴 중 가장 유사한 패턴으로 대치된다. 즉 각각의 코드 벡터는 encoding 과정에서 codebook으로부터 찾아낸 가장 유사한 codevector의 index에 의하여 대치된다. encoding 과정에서 계산량은 codebook의 크기(codevector의 갯수)에 따라 선형적으로 증가하게 되며, 압축 비트율 또한 식 (1)과 같이 codebook의 크기에 영향을 받는다.

$$BR = \frac{\log_2 N_c}{k} \quad (bpp) \quad (1)$$

여기서  $BR$  : Bit Rate

$N_c$  : codebook의 크기 (codevector의 개수)

$k$  : codevector의 크기

식 (1)로부터 벡터의 크기  $k$ 가 커질수록 낮은 비트율이 얻어짐을 알 수 있다. 그러나,  $k$ 가 증가할수록 계산량이 지수적으로 증가하므로, codevector의 크기가 클수록 낮은 비트율을 얻을 수 있음에도 불구하고 64이상의 크기를 갖는 벡터는 거의 사용되지 않는다[3].

Encoding 과정에서의 계산량을 줄이기 위하여 신속한 codevector 탐색방법들이 제안되었으나, 이들 방법들은 encoding 속도와 압축영상의 화질사이에 tradeoff를 요구한다 [4]~[12]. 반면 계층적 VQ (Hierarchical Vector Quantizer)에서는 주어진 영상 데이터들이 통계학적 상호 의존도에 따라 여러 가지 크기의 block들로 분할되며 block의 크기가 작을수록 영상의 세밀한 부분들이 잘 표현될 수 있다. 따라서 크기가 큰 block들은 세밀한 부분을 별로 포함하고 있지 않으므로 낮은 정량화 레벨(quantization level)로 대략적인 코딩을 하는 반면 크기가 작은 블록들은 높은 정량화 레벨로 세밀하게 코딩함으로써 인코딩 복잡도를 줄임과 동시에 압축 효율을 높일 수 있다.

HVQ(Hierarchical Vector Quantizer)를 구현하기 위하여 가장 널리 사용되는 알고리즘 중의 하나가 quadtree 분할방법이며 [13][14]~[18] quadtree 분할에 대한 총괄적인 설명은 Samet의 논문에서 찾아볼 수 있다 [16]. Quadtree 분할의 일반적인 개념은 블록의 크기가 8x8 이상이 되는 super-vector의 연속적인 분할로써, super-vector는 주어진 임계값(threshold)에 의하여 균일한(homogeneous) 작은 vector들로 분할된다. 이러한 기술은 블록간의 상호 의존성(dependency)을 줄이는 효율적인 방법이나 하나의 영상을 효율적으로 분할하기 위해 사용될 최적의 임계값을 결정하는 것이 연구의 과제로 남아 있다. 일찍이 Nasrabadi는 주

\* 正 會 員 : 漢 城 大 情 報 電 算 學 部 教 授 · 工 博  
接 受 日 字 : 1999年 3月 25日  
最 終 完 了 : 1999年 7月 22日

어진 블록의 분산(variance)을 임계값으로 이용하여 quad-tree를 분할하였고, Chou et al은 Brieman, Friedman, Oishen, Stone(BFOS)[19] 알고리즘의 변형으로 최적의 블록 구조를 표현하려고 노력했다 [20]. Sullivan은 각각의 quadtree 레벨에 대한 최적의 곱셈값을 발견하기 위하여 Lagrange Multiplier 개념[21]을 사용하였고, 평균이나 기울기(gradient) 같은 plane parameter(PP)를 사용하는 방법이 Strobach[1][22]에 의해 연구되었다. 그러나 어떠한 방법도 화질과 비트율의 적절한 tradeoff라는 관점에서 최적의 해법을 제시하지 못하고 있다.

본 논문에서는 이러한 연구들보다 효율적인 방법으로 최적의 임계값을 결정하기 위해 각각의 quadtree level에서 leaf-node를 발견할 누적확률을 이용하여 수학적 모델을 설정하였다. 이 모델은 quadtree 분할후 생성되는 최종 block의 수를 가능한 줄일수 있도록 quadtree 하위 level block들의 표준편차들은 상위 level block의 표준 편차보다 적지 않다는 가정하에서 유도된다. 따라서 보다많은 sub-block들이 병합하여 super-block을 형성함으로써 최종적으로 생성되는 block의 수를 가능한 최소화하여 낮은 비트율을 얻을 수 있었다. 최적의 임계값 설정을 위한 수학적 모델의 유도 과정과 실험결과는 각각 3장과 4장에서 서술되어 진다.

2. 계층적 구현을 위한 Quadtree 분할

Quadtree 분할의 기본 개념은 주어진 영상을 균일한 작은 block으로 연속적으로 분할하는 것으로써, 모든 block들의 block 표준편차가 주어진 임계값보다 작거나 같을 때까지 순환적으로 반복된다. 만일 주어진 block의 block 표준편차가 낮다면 vector의 크기가 크고 낮은 정량화 레벨을 갖는 codebook으로 코딩되는 반면, block 표준편차가 높으면 이러한 block들은 4개의 작은 block으로 다시 분할되며 vector의 크기가 작고 높은 정량화 level을 갖는 codebook으로 coding된다.

주어진 영상의 quadtree 표현에 있어서, non-leaf 노드는 주어진 영상의 sub-block에 대응하는 4개의 자노드를 갖고, leaf 노드는 더 이상의 분할이 필요치 않은 block으로 정의된다. 근노드(root node)는 완전한 하나의 영상을 의미하고 quadtree의 첫 번째 level은 4개의 sub-block을 포함하며, 이러한 분할절차를 quadtree의 top-down 구축이라고 한다.

그림 (1)은 top-down quadtree 분할과 분할된 구조, 그리고 이에 대응하는 binary map에 대한 보기로써 흰색과 회색 노드는 각각 non-leaf와 leaf 노드를 나타낸다. 그림 1(c)에서 보는 바와 같이 흰색 노드는 위에서 아래로 그리고 오른쪽에서 왼쪽의 순서에 따라 이진수 1로 코딩되는 반면 회색노드는 이진수 0으로 코딩된다. Quadtree로 표현된 영상의 장점은 quadtree 구조를 나타내는데 필요한 비트수(overhead)가 적다는 것이다. Quadtree에서 각각의 노드는 더 작은 노드로 분할이 될 것인지, 더 이상의 분할이 필요없는 leaf 노드가 될 것인지를 결정하는데 단지 1비트(0이나1)만이 필요하다. 예를들어 그림 1(a)의 16x16 block의 경우는 quadtree를 표현하는데 단지 0.0664(17/256) bpp의 overhead만이 필요하다.

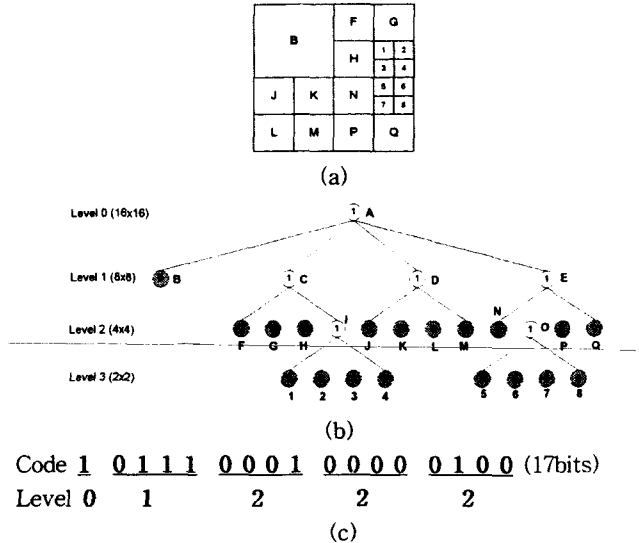


그림 1 16x16 block에 대한 quadtree 표현  
 (a) 분할된 영상 block(16x16) (b) Quadtree 표현  
 (c) Binary Bitmap  
**Fig. 1** Quadtree Representation for a 16x16 block  
 (a) Segmented Image Block(16x16)  
 (b) Quadtree Representation  
 (c) Binary Bitmap

Top-down quadtree를 표현하는 overhead를 일반적인 형태로 표현하기 위해 level k에서 leaf-node를 발견할 실험적 확률(empirical probability)이 다음과 같이 정의되었다.

$$P_k = \frac{N_k}{4^k} \tag{2}$$

여기서  $N_k$ : level k에서 자노드를 갖지 않는 node의 수  
 $k$ : level의 수

Level k에서 leaf-node를 발견할 누적확률(cumulative probability)은 다음과 같이 정의된다.

$$q_k = \begin{cases} q_{k+1} - P_{k+1} & k = 0, \dots, n-1 \\ 1 & k = n \end{cases} \tag{3}$$

하나의 quadtree에 포함된 leaf의 수는 식 (4)와 같이 정의되고 하나의 quadtree를 표현하는데 필요한 bit수는 식 (5)와 같이 나타낼 수 있다.

$$N_{L(qt)} = \sum_{k=0}^n 4^k P_k \tag{4}$$

$$N_{B(qt)} = \frac{4^n - 1}{3} - \sum_{k=0}^{n-2} 4^{k+1} q_k \tag{5}$$

여기서

$N_{L(qt)}$ : 하나의 quadtree내에 존재하는 leaf의 수.

$N_{B(qt)}$ : 하나의 quadtree를 표현하는데 필요한 bit 수.

$n$ : 하나의 quadtree내에 존재하는 level의 수.

결과적으로 overhead의 비트율은 식 (6)과 같이 표현된다.

$$BR_{qt} = \frac{N_{B(qt)}}{4^{n+1}} \tag{6}$$

일반적으로 quadtree 분할에 있어서 가장 큰 block으로 16x16이상의 크기를 선택한다면 encoding을 위한 codebook을 저장하기 위해 많은 용량의 저장공간이 필요할 뿐더러, block 내의 데이터값들의 변화가 크지 않다 하더라도 모자이크 효과가 예상되므로 본 실험에서는 8x8 block을 block의 최대 크기로 선정하였다. 그림 (2)는 HVQ의 block-diagram을 나타내고 있다.

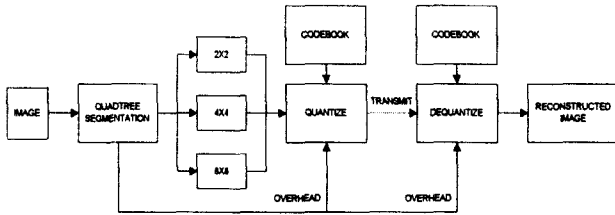


그림 2 HVQ의 Block-Diagram  
Fig. 2 Block -Diagram of HVQ

3. 최적의 임계값

일반적으로, quadtree 분할의 효율은 MSE, bpp, 얻어진 leaf의 수(leaf의 수가 적을수록 비트율을 낮출 수 있으므로)에 의해 측정되는데 이러한 요소들은 quadtree 분할을 위하여 사용된 임계값과 주어진 영상의 데이터 분포에 의존한다. 따라서 최적의 임계값을 발견하기 위하여 많은 연구가 되어 왔으나 최적의 임계값을 발견하기 위한 진정한 해법은 발견되어지지 않았다[19][23]~[26]. 본 장에서는 최적의 임계값을 발견하기 위한 수학적 유도 방법을 소개한다.

하나의 super-block (8x8)이 주어진 임계값에 의하여 분할된 후 sub-block들의 표준편차는 그들의 상위 block (parent block)보다 크거나 같을 수 있고, 또한 작을 수도 있으므로 각각의 quadtree level에 따라 각기 다른 임계값들을 사용할 필요가 있다. 본 장은 각각의 quadtree level에 대한 최적의 초기값을 추정하기 위한 수학적 모델을 제시한다.

식 (2)에서 식 (6)까지는 Top-down quadtree 분할시 필요한 overhead의 계산을 나타내고 있다. 수학적 모델링에 대한 복잡성을 피하기 위하여서 2x2를 가장 작은 block 단위로 하는 bottom-up quadtree 구축을 고려했다. Bottom-up quadtree 구축은 4개의 leaf에 대한 모든 표준편차가 주어진 임계값보다 작거나 같을 때 4개의 leaf를 병합하여 하나의 상위 노드를 만들고 이러한 절차는 근노드를 만날 때까지 순환적으로 반복된다. 위에서 언급한 것처럼, 4개의 block이 병합하여 형성된 block의 표준편차는 병합된 4개 block에 대한 각각의 표준편차보다 크거나 같고 혹은 작을 수도 있다. 그러나 영상압축의 궁극적인 목표는 낮은 비트율로 영상을 표현하는데 있으므로 quadtree에 의하여 분할된 block수를 줄임으로써 낮은 비트율을 얻을 수 있다. 따라서 많은 병합이 일어나기 위하여서는 하위 level block의 block 표준편차 들은 상위 level의 block 표준편차보다 크거나 같아야 한다는 가정을 세울 수 있다. 이러한 가정에 기초해서 첫 번째 level에 대한 표준편차는 식 (7)과 같은 관계식으로 표현될 수 있으며, 이러한 가정을 토대로 하여서 수학적 모델을 구축할 수 있다.

$$STD_1 \leq \frac{1}{4} T_0 [\gamma_0(1) + \gamma_0(2) + \gamma_0(3) + \gamma_0(4)] \tag{7}$$

여기서,  $\gamma_0(k) = \frac{STD_0(k)}{T_0} \leq 1, k=1, 2, 3, 4$

$STD_k$  : level k의 표준편차

$T_k$  : level k의 임계값

최악의 병합일 때  $\gamma_0(k)$ 와  $STD_1$ 은 각각 1과  $T_0$ 가 된다. 마찬가지로 두 번째 level의 관계식은 가설에 의해 식 (8)처럼 유도된다.

$$STD_2 \leq \frac{1}{4} T_1 [\gamma_1(1) + \gamma_1(2) + \gamma_1(3) + \gamma_1(4)] + \frac{1}{16} T_0 [\gamma_0(1,1) + \gamma_0(1,2) + \gamma_0(1,3) + \gamma_0(1,4) + \gamma_0(2,1) + \gamma_0(2,2) + \gamma_0(2,3) + \gamma_0(2,4) + \gamma_0(3,1) + \gamma_0(3,2) + \gamma_0(3,3) + \gamma_0(3,4) + \gamma_0(4,1) + \gamma_0(4,2) + \gamma_0(4,3) + \gamma_0(4,4)] \tag{8}$$

만일 최악의 병합을 다시 고려한다면  $STD_2$ 는  $T_1 + T_0$ 와 같게 되고, 두 번째 level의 표준편차는 식 (9)와 같이 upper-bound된다.

$$STD_2 \leq T_1 + T_0 \tag{9}$$

$$STD_k \leq T_{k-1} + T_{k-2} + T_{k-3} \dots T_0 = \sum_{m=0}^{k-1} T_m \tag{10}$$

여기서,  $STD_k$  : level k의 표준편차

식 (2),(3),(4)의 관계식은, bottom-up 구축의 경우 식 (11),(12),(13)처럼 다시 표현할 수 있다.

$$p_k = \frac{N_k}{4^{n-k}} \tag{11}$$

$$q_k = \begin{cases} 1 & k=0 \\ q_{k-1} - p_{k-1} & k=1, \dots, n \end{cases} \tag{12}$$

$$N_{L(q)} = \sum_{k=0}^n 4^{n-k} p_k \tag{13}$$

여기서,

$p_k$  : level k에서 leaf-node가 발견될 실험적 확률.

$q_k$  : level k에서 leaf-node가 발견될 누적 확률.

$N_{L(q)}$  : leaf의 수.

식 (10)은 식 (14)와 같이 일반화하여 표현할 수 있다.

$$\sum_{k=1}^n p_k \times [STD_k] \leq \sum_{k=1}^n p_k \times [\sum_{m=0}^{k-1} T_m] \tag{14}$$

식 (12)으로부터 식 (14)는 식 (15)처럼 쓸 수 있다.

$$\sum_{k=1}^n (q_k - q_{k+1}) \times [STD_k] \leq \sum_{k=1}^n (q_k - q_{k+1}) \times [\sum_{m=0}^{k-1} T_m] \tag{15}$$

식 (15)의 오른쪽은 다음과 같이 다시 쓸 수 있다.

$$\sum_{k=1}^n (q_k - q_{k+1}) \times \left[ \sum_{m=0}^{k-1} T_m \right]$$

$$= (q_1 - q_2)T_0 + (q_2 - q_3)(T_0 + T_1) + (q_3 - q_4)(T_0 + T_1 + T_2) + \dots + (q_n - q_{n+1})(T_0 + T_1 + \dots + T_{n-1})$$

$$= q_1T_0 - q_2T_0 + q_2T_0 + q_2T_1 - q_3T_0 - q_3T_1 + q_3T_0 + q_3T_1 + q_3T_2 - q_4T_0 - q_4T_1 - q_4T_2 + \dots - q_{n+1}T_0 - \dots - q_{n+1}T_{n-1}$$

$$= q_1T_0 + q_2T_1 + q_3T_2 + \dots + q_nT_{n-1} - q_{n+1}T_0 - \dots - q_{n+1}T_{n-1}$$

$$= \sum_{k=1}^n q_k T_{k-1} - q_{n+1} \sum_{m=1}^n T_{m-1} \tag{16}$$

Level  $n+1$ 은 존재하지 않으므로 식 (15)의 두 번째 식은 0이 되며, 1부터  $n$ 까지  $p_k$ 의 합은 1이므로 식 (15)와 식 (16)로부터 식 (17)을 얻을 수 있다.

$$STD_k \leq \sum_{k=1}^n q_k T_{k-1} \tag{17}$$

Quadtree의 Bottom-up 구축의 경우에 식 (13)는 다음과 같이 다시 쓸 수 있다.

$$N_{L(q)} = \sum_{k=0}^n 4^{n-k} p_k = 4^n p_0 + \sum_{k=1}^n 4^{n-k} p_k$$

$$= 4^n (1 - q_1) + \sum_{k=1}^n 4^{n-k} (q_k - q_{k+1})$$

$$= 4^n \left( 1 - q_1 + \frac{q_1}{4} - \frac{q_2}{4} + \frac{q_2}{16} - \frac{q_3}{16} + \frac{q_3}{64} - \frac{q_4}{64} + \dots + \frac{q_n}{4^n} - \frac{q_{n+1}}{4^n} \right)$$

Level  $n+1$ 은 존재하지 않으므로,

$$N_{L(q)} = 4^n \left( 1 - \frac{3q_1}{4} - \frac{3q_2}{16} - \frac{3q_3}{64} - \dots - \frac{3q_n}{4^n} \right)$$

$$= 4^n \left( 1 - \frac{3}{4} \left( q_1 + \frac{q_2}{4} + \dots + \frac{q_n}{4^{n-1}} \right) \right)$$

$$= 4^n \left( 1 - 3 \sum_{k=1}^n \frac{q_k}{4^k} \right) = 4^n \left( 1 - \frac{3}{4} q_1 - 3 \sum_{k=2}^n \frac{q_k}{4^k} \right) \tag{18}$$

식 (18)로부터 식 (19)와 같이  $q_1$ 을 얻는다.

$$q_1 = \frac{4}{3} (1 - 4^{-n} N_{L(q)}) - \sum_{k=2}^n 4^{1-k} q_k \tag{19}$$

식 (17)은 식 (20)처럼 다시 쓸 수 있다.

$$STD_k \leq q_1 T_0 + \sum_{k=2}^n q_k T_{k-1} \tag{20}$$

식 (20)에서  $q_1$ 에 대해 식 (19)를 대입하면 식 (21)을 얻을 수 있다.

$$STD_k \leq \frac{4}{3} (1 - 4^{-n} N_{L(q)}) T_0 - \sum_{k=2}^n 4^{1-k} q_k T_0 + \sum_{k=2}^n q_k T_{k-1}$$

$$= \frac{4}{3} (1 - 4^{-n} N_{L(q)}) T_0 + \sum_{k=2}^n q_k (T_{k-1} - 4^{1-k} T_0) \tag{21}$$

식 (21)의 오른쪽 첫 번째 식인  $1 - 4^{-n} N_{L(q)}$ 는 항상 양수이므로, 근노드의 표준편차를 최소화시키기 위하여 식 (21)의 두 번째 식은 0보다 작거나 같아야 하기 때문에 식 (22)와 같은 임계값들간의 관계식을 얻을 수 있다.

$$T_{k-1} - 4^{1-k} T_0 \leq 0$$

$$\therefore T_{k-1} \leq 4^{1-k} T_0 \tag{22}$$

식 (22)는 bottom-up quadtree 구축에 대한 각 level 임계값들의 관계식이므로, Top-down quadtree 구축의 경우에는 식 (22)는 식 (23)처럼 수정되어 표현된다.

$$T_{k-1} \leq 4^{k-1} T_0 \tag{23}$$

식 (23)에서처럼 각 level의 임계값은 첫 번째 임계값의 함수로써 upper-bound된다. 즉, 두 번째 임계값은 첫 번째 임계값에 대한 4의 배수로 bound된다. 식 (23)은 식 (7)의 가설로부터 설정된 관계식이지만 주어진 영상의 데이터 분포와 관계없이 대부분의 영상에 대하여서 최적의 임계값을 제공하여 준다. 반면 식 (23)을 만족하는 무한한 해가 존재하므로  $T_0$ 에 대한 최적의 계수를 설정하기 위하여 추가의 컴퓨터 시뮬레이션을 하였고 이에 대한 결과 및 토론은 다음 장에 주어진다.

#### 4. 실험 결과 및 토론

Unix C를 이용하여 HVQ 시스템이 설계되었고 그림 (3)에 보여진 U.S.C. 표준영상을 대상으로 압축 효율을 측정하였다. 3장에서 설명한 바와 같이 식 (4)를 만족하는 해는 무수히 많이 존재할 수 있으나  $T_0$ 의 계수는 일정한 값에서 최소의 MSE를 갖음을 알 수 있었다. 그림 (4)은 0보다 크고 4보다 작은  $T_0$ 의 계수를 변화시켰을 때 발생하는 MSE를 그래프로 나타내었다. 그래프에서 볼 수 있는 것처럼  $T_0$ 에 대한 계수가 2.6일 때 가장 작은 MSE가 발생함을 알 수 있었다.  $T_0$ 에 대한 최적의 계수는 주어진 영상의 데이터 분포에 따라서 약간의 차이가 있으나 대부분의 영상에서  $T_0$ 에 대한 최적의 계수는 2.6 근처로 수렴함을 실험을 통하여서 알 수 있었다. 따라서 본 연구에서는  $T_0$ 에 대한 최적의 계수로써 2.6을 사용 하였고 압축된 영상의 화질을 PSNR의 형태로 측정 하였을 때 비트율 전 영역에서 향상된 화질의 압축영상을 출력 하였다. HVQ를 구현하기 위한 codebook 은 무작위로 선택된 1,000개의 256 x 256 영상으로부터 LBG 알고리즘을 이용하여서 제작하였다 [27].

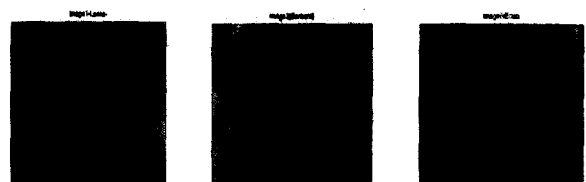


그림 3 Test 영상 (왼쪽으로부터 Lenna, Barbara, Boat)  
Fig. 3 Test Images(From left, Lenna, Barbara, Boat)

6. 참고 문헌

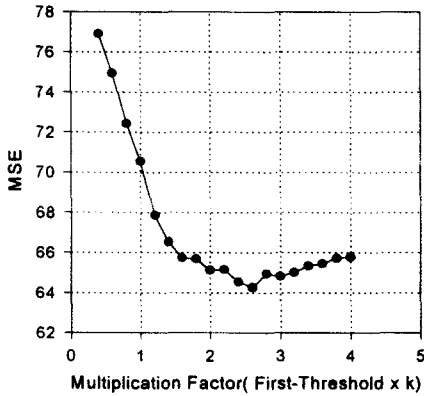


그림 4  $T_0$ 의 계수 변화에 따른 MSE의 변화(1,000개의 256x256 영상에 대한 평균치)

Fig. 4. The variation of MSE according to the coefficient of  $T_0$

본 실험에서 설계된 HVQ 시스템과 최근에 발표된 다른 압축시스템 사이의 압축효율의 비교가 PSNR의 형태로 표 (1)에 주어졌다. 표 (1)에 주어진 것처럼, 다른 압축 시스템보다 1 dB이상 향상된 PSNR을 HVQ 시스템을 통하여 얻을 수 있었다. 비록 그림 (3)에 보여진 세계의 영상에 대한 압축 효율의 비교가 표 (1)에 주어졌지만, 주어진 영상의 데이터 분포와 관계없이 대부분의 다른 영상에 대해서도 본 연구에서의 HVQ 시스템은 향상된 화질의 영상을 출력하였다.

표 1. 여러 가지 압축 시스템의 압축 효율 비교 [13]  
(HVQ: 본 논문에서 구현한 Hierarchical VQ, QT: Quadtree Coder PVQ: Pyramid VQ, AWPVQ: Adaptive Weighted VQ)

Table 1. The Comparison of performance for several Image compression Systems.

영상	비트율 (bep)	PSNR (dB)			
		HVQ	QT	AWPVQ	PVQ
Lenna	0.5	30.61	28.91	28.41	27.62
	1.0	33.31	32.55	32.20	31.67
Barbara	0.5	28.42	27.27	26.84	26.02
	1.0	31.96	30.48	29.93	29.46
Boat	0.5	29.51	28.12	27.67	26.89
	1.0	33.38	31.93	31.36	30.73

5. 결 론

Quadtree 분할을 이용하여 HVQ를 구현할 때 문제시되는, 최적의 임계값을 설정하기 위하여 수학적 모델링에 기반한 해법이 제안되었다. 하위 level block의 표준편차들은 상위 level block의 표준편차보다 작지 않아야 한다는 가정에서 출발한 수학적 모델은 생성되는 block의 수를 가능한 억제한으로써 높은 압축비를 얻을 수 있도록 하였다. 본 논문에서 제시한 임계값 설정 방법으로 구현된 HVQ는 비트율 전 범위에서 다른 압축 시스템에 비하여 1 dB이상 향상된 화질의 압축영상을 출력하였다.

- [1] C. E. Shannon, "The mathematical Theory of Communication", Urbana: University of Illinois Press, 1978.
- [2] R. J. Clarke, "Transform Coding of Images", pp 4, Academic Press, London, 1985.
- [3] A. Gersho and R. M. Gray, "Vector Quantization and Signal Compression", Kluwer Academic Publishers, Boston, 1991.
- [4] D. Y. Cheng and Gersho A., "A Fast Codebook Search Algorithm for Nearest-Neighbor Pattern Matching", *Proceedings of ICASSP*, Tokyo, pp. 6.14.1-6.14.4, 1986.
- [5] D. Y. Cheng, A. Gersho, B. Ramamurthi, and Y. Shoham, "Fast Search Algorithms for Vector Quantization and Pattern Matching", *Proceedings of ICASSP*, pp. 9.11.1-9.11.4, 1984.
- [6] A. Lowry, S. Hassain, and W. Miller, "Binary Search Trees for Vector Quantization", *Proceedings of ICASSP*, pp. 51.8.1-51.8.4, 1987.
- [7] M. R. Soleymani and S. D. Morgera, "A High Speed Search Algorithm for Vector Quantization", *Proceedings of ICASSP*, pp. 45.6.1-45.6.3, 1987.
- [8] K. K. Paliwal and V. Ramasubramanian, "Effect of Ordering the Codebook on the Efficiency of the Partial Distance Search Algorithm for Vector Quantization", *IEEE Transactions on Communications*, Vol. 37, No. 5, pp. 528-540, 1989, 5.
- [9] S. H. Chen and W. M. Hsieh, "Fast Algorithm for VQ Codebook Design", *IEE Proceedings-I*, Vol. 138, No. 5, pp. 357-362, 1991, 10.
- [10] L. Torres L. and J. Huguet, "An Improvement on Codebook Search for Vector Quantization", *IEEE Transactions on Communications*, Vol. 42, No. 2/3/4, pp. 208-210, 1994, 2/3/4.
- [11] M. H. Lee and G. Crebbin, "Classified Vector Quantization with Variable Block-Size DCT Models", *IEE Proc. Vision and Image Processing*, Vol. 141, No. 1, pp. 39-48, 1994, 2.
- [12] V. S. Sitaram, C. M. Huang, and P. D. Israelsen, "Efficient Codebooks for Vector Quantization Image Compression with an Adaptive Tree Search Algorithm", *IEEE Transactions on Communications*, Vol. 42, No. 11, 1994, 11.
- [13] E. Shusterman and M. Feder, "Image Compression via Improved Quadtree Decomposition Algorithms", *IEEE Transactions on Image Processing*, Vol. 3, No. 2, pp. 207-215, 1994, 3.
- [14] C. L. Jackins and S. L. Tanimoto, "Quad-Tree, and k-Trees: A generalized Approach to Recursive Decomposition of Euclidean Space", *IEEE Transactions on Pattern Analysis and Machine*

- Intelligence*, Vol. PAM-5, No. 5, pp. 533-539, 1983, 9.
- [15] W. I. Grosky and R. Jain, "Optimal Quadtrees for Image Segmentations, *IEEE Transactions on Pattern Analysis and Machine Intelligence*", Vol. PAMI-5, No. 1, pp. 77-83, 1983, 1.
- [16] H. Samet, "The Quadtree and Related Hierarchical Data Structures, *Computing Survey*", Vol. 16, No. 2, pp. 187-260, 1984, 6.
- [17] C. A. Shaffer C. A. and H. Samet, "Optimal Quadtree Construction Algorithms", *Computer Vision, Graphics, and Image Processing*, Vol. 37, pp. 402-419, 1987.
- [18] P. Strobach, "Image Coding Based on Quadtree-Structured Recursive Least-Squares Approximation", *Proceedings of IEEE ICASSP*, pp. 1961-1964, 1989.
- [19] C. Y. Chiu and R. L. Baker, "Quad-Tree Product Vector Quantization of Images", in *Proceedings of SPIE Conference, Advances Image Compression Automat. Target Recognition*, Vol. 1099, pp. 142-153, 1989, 3.
- [20] N. M. Nasrabadi, S. E. Lin, and Y. Feng, "Interframe Hierarchical Vector Quantization", *Optical Engineering*, Vol. 28, No. 7, pp. 717-725, 1989, 7.
- [21] G. J. Sullivan and R. L. Baker, "Efficient Quadtree Coding of Image and Video", *IEEE Transactions on Image Processing*, Vol. 3, No. 3, pp. 327-331, 1994, 5.
- [22] P. Strobach, "Quadtree-Structured Recursive Plane Decomposition Coding of Image", *IEEE Transactions on Signal Processing*, Vol. 39, pp. 1380-1397, 1991, 6.
- [23] J. Vaisey and A. Gersho, "Variable block size Image Coding", in *Proceedings of IEEE International Conference Acoust. Speech Signal Processing (ICASSP)*, pp. 25.1.1-25.1.4, 1987, 4.
- [24] P. Strobach, "Quadtree-Structured Linear Prediction Models for Image Sequence", *Processing, IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, pp. 742-748, 1989, 7.
- [25] P. Strobach, "Tree-Structured Scene Adaptive Coder", *IEEE Transactions on Communications*, Vol. 38, pp. 477-486, 1990, 4.
- [26] J. Vaisey and A. Gersho, "Image Compression with variable block size segmentation", *IEEE Transactions on Signal Processing*, Vol. 40, pp. 204.
- [27] Linde Y., Buzo A., and Gray R. M., "An Algorithm for Vector Quantizer Design", *IEEE Transactions on Communications*, Vol. COM.-28, No.1, pp. 84-95, 1980, 1.

## 저 자 소 개



이 항 찬 (李 恒 燦)

1963 3월 11일생. 1986년 인하대 공대 전기 공학과 졸업. 1992년 미 오를라호마 주립대학교 대학원 전기 및 컴퓨터 공학과 졸업(석사). 1997년 동대학원 전기 및 컴퓨터공학과 졸업(박사). 1998년 3월~1999년 3월 청운대학교 교수. 현재 한성대학교 정보 전산학부 교수.

Tel : 02-760-4307

E-mail : lhangch@hansung.ac.kr