論文2001-38CI-6-9

# Signal Set Partitioning을 이용한 격자 양자화의 비 손실 부호화 기법

# (Lossless Coding Scheme for Lattice Vector Quantizer Using Signal Set Partitioning Method)

金 元 河 *

(Won-Ha Kim)


## 요 약

격자 벡터 양자화의 비 손실 과정에서는 생성된 코드단어들을 radius 열과 Index 열로 열거한다. radius 열은 run-length 부호화한 한 다음 Entropy 부호화한다. 또한 index 열들은 이진의 고정길이로 표현한다. 비트율이 증가함에 따라 index 비트는 선형적으로 증가하여서 부호화 성능을 감소시킨다. 이 논문에서는, 넓은 비트율의 범위에서 index 비트를 줄이기 위해서, set partitioning 방식을 채택한 새로운 열거 알고리즘을 개발하였다. 제안된 열거 방법은 큰 index 값을 작은 값들을 천이 시켜서 index 비트를 줄인다. 제안된 비손실 기법을 웨이블릿 기반의 영상 부호화에 적용시켰을 때, 0.3 bits/pixel 이상의 비트률에서 기존의 비손실 부호화 방식보다 10%이상의 비트율을 감소시켰다.

## Abstract

In the lossless step of Lattice Vector Quantization(LVQ), the lattice codewords produced at quantization step are enumerated into radius sequence and index sequence. The radius sequence is run-length coded and then entropy coded, and the index sequence is represented by fixed length binary bits. As bit rate increases, the index bit linearly increases and deteriorates the coding performances. To reduce the index bits across the wide range of bit rates, we developed a novel lattice enumeration algorithm adopting the set partitioning method. The proposed enumeration method shifts down large index values to smaller ones and so reduces the index bits. When the proposed lossless coding scheme is applied to a wavelet based image coding, the proposed scheme achieves more than 10% at bit rates higher than 0.3 bits/pixel over the conventional lossless coding method, and yields more improvement as bit rate becomes higher.


* 正會員, 明知大學校 電子情報通信工學部

(Myongju University, Division of Information & Communication Eng,)

※ 본 연구는 한국과학재단 목적기초연구(2001-1-30200-024-2) 지원으로 수행되었음

接受日字:2001年1月12日, 수정완료일:2001年8月2日

## I. Introduction

A Lattice is a set of regularly arranged points in a vector space. A Lattice Vector Quantizer(LVQ) uses lattice points as codewords. Thus, LVQ eliminates the need for a lengthy training process in designing the code book due to the regular structure of lattice[1]. Therefore, LVQ has been successfully used in a practical coding system, because of its low complexities and competitive performance[2][3].

In the LVQ lossless coding, lattice codewords are enumerated into radius and index[4]. The radius is the norm of the lattice and the index is the lattice's position on the surface at its norm. Since the radius sequence often includes long streams of zeros, run-length coding and entropy coding are usually applied to the radius sequence. On the other hand, index sequence is almost decorrelated and simply represented by binary bits with minimum bit accuracy required to cover the index range[2][4].

In this paper, we have improved the LVQ lossless coding scheme, while most previous research has focused on improving quantization process[2][3][5]. We have analyzed the performance of LVQ lossless coding for a Laplacian source to observe that 1) the actual performance of LVQ lossless coding deviates more from the theoretical bound at higher bit rates and 2) the portion of index bit hikes as bit rate increases. From these observations, we conjectured that a LVQ lossless coding can be improved by reducing the large index bits.

To prevent the need for too large length of index bits, we have developed a novel lattice codeword enumeration algorithm adopting the set partitioning method[6]. The algorithm partitions the index set to shift down large index values to smaller ones and so reduce index bits. The partitioning information is carried on the radii of the partitioned indices. Although the partitioning process increases the entropy of the radius sequence, the index bit reduction surpasses the increment on radius bit, and thus the total bits are decreased. When we applied the proposed lossless coding to a wavelet based image coding, the proposed scheme improves 15%~20% of coding performance over the conventional lossless scheme at bit rates more than 0.25 bits/pixel, and achieves more improvement at higher bit rates.

The paper is organized as follows. In Section 2, we review lattice vector quantization(LVQ) encoding a Laplacian source, and the conventional lossless coding scheme. Next, in Section 3, we analyze in detail the performance of LVQ lossless coding. Then, in Section 4, we develop the novel LVQ lossless coding method. Finally, in Section 5, we demonstrate the superior performance of the proposed LVQ lossless coding, and in Section 6, we make some brief concluding remarks.

## Ⅱ. Preliminaries

### 1. Laplacian Source

In view of image and video coding, a Laplacian has been verified to be a suitable model for DCT transformed data or sub-band coded image/audio data(except the DC-band data)[3][7]. Therefore, we adopt a Laplacian source as the model for an input source.

Let $\{x_i\}$ be a sequence of identical and independent distribution(i.i.d) Laplacian random variables with zero mean and variance $2/\lambda^2$. Then, the joint *pdf* of an n-dimensional Laplacian random vector $X=[x_1, x_2, \cdots, x_n]^T$ is

$$f(X)=f(x_1, x_2, \cdots, x_n)=\prod_{i=1}^{n} f(x_i)=\left(\frac{\lambda}{2}\right)^n e^{-\lambda \sum_1^n |x_i|}$$

Observing that $\|X\|_1=\sum_1^n |x_i|$, the Laplacian pdf is constant at the contours specified by $l^1$-norm distance. The equi-probable contours form pyramidal surfaces. In order to be consistent with the source's geometric properties, the proper distance measure for a Laplacian source must be $l^1$-norm[3][5]. Setting a random variable to be $r=\sum_1^n |x_i|$, we can transform the Laplacian pdf to gamma pdf with random variable $r$ such as

$$g(r)=\frac{\lambda^n}{\Gamma(n)} r^{n-1} e^{-\lambda \cdot r}, \qquad \text{where} \quad r>0.$$

The Laplacian source and the gamma function are related as follows :

$$\int_{X \le R} f(X)\, dX=\int_0^R g(r)\, dr,$$

$$\int_{X \geq R} f(X)\,dX = \int_R^{\infty} g(r)\,dr. \quad (1)$$

2. Lattice Vector Quantization(LVQ)

The lattice code words partition the n-dimensional space into identical Voronoi cells each with a code word as its centroid. The fundamental Voronoi cell $V_0$ is centered at the origin. The volume of the Voronoi cell $V_i$ of the lattice point $Y_i$ is $Vol(V_i) = Vol(V_0)$. If a lattice is c-time scaled, the volume of the scaled Voronoi cell is calculated by $Vol(cV_i) = c^n Vol(V_0)$[1].

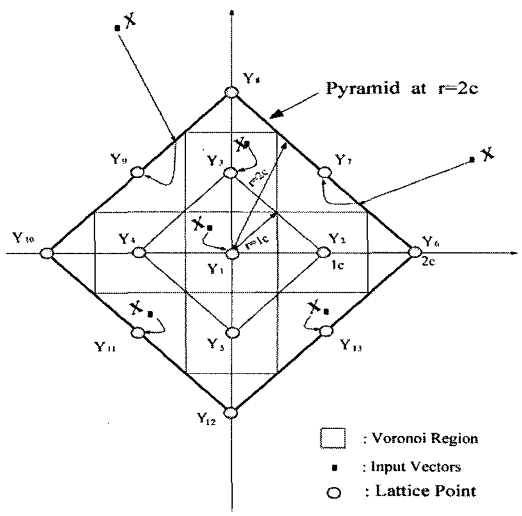



그림 1. 2차원 입방( $Z^2$ )격자의 Pyramid 격자 벡터 양자화

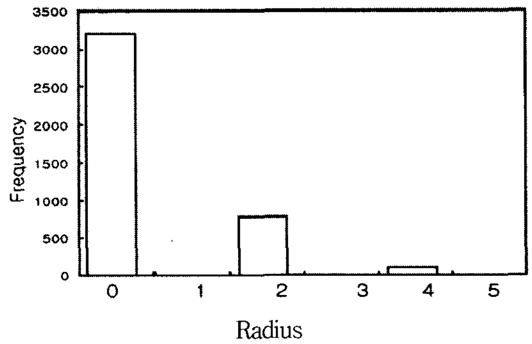
Fig. 1. Pyramidal Lattice Vector Quantization Scheme for 2-dimensional cubic( $Z^2$ ) lattice. $B^{gr}(2c) = \{X: \|X\|_1 \leq 2c\}$, $B^{ov}(2c) = \{X: \|X\|_1 > 2c\}$. $C = \{Y_1, Y_2, \cdots, Y_{13}\}$, Code Radius $m=2$, Code Size $|C| = 13$, Bit rate $R = \log_2 |C| / dimension = 1.85$ bit/sample, Lattice points on Pyramid : $C_s(0)=1$, $C_s(1)=4$, $C_s(2)=8$.

Let $C_r$ be the number of lattice points on the surface at a normalized distance $r$ from the origin. Values of $C_r$ have been calculated in[3] for popularly used lattice types. Define the codebook radius $m$ as the maximum normalized distance of lattice points in a codebook. Then, the total number of codewords, that is, the codebook size is $\sum_{k=0}^{m} C_k$. Table I shows the number of lattice points on and within the pyramid at r, namely, $C_r$ and $\sum_{k=0}^{r} C_k$[3]. The code book radius $m$ truncates the vector space into the granular and overload regions such as

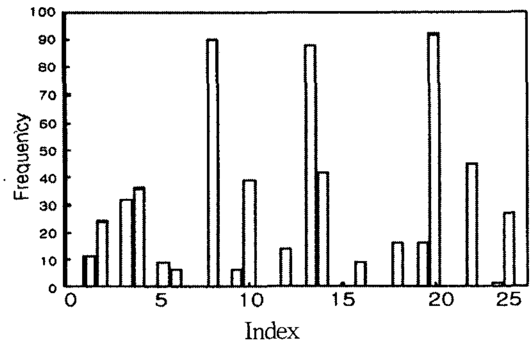
Granular region : $B^{gr}(cm) = \{X \in R^n : \|X\| \leq cm\}$

Overload region : $B^{ov}(cm) = \{X \in R^n : \|X\| > cm\}$

In case of one dimension, LVQ is an uniform




(a) Histogram of radius sequence




(b) Histogram of index sequence

그림 2. radius 열과 index 열의 빈도수 분포

Fig. 2. Histogram of the radius sequence and the index sequence. The input source is a Laplacian source with zero mean, unit variance. The used lattice type is $D4$, and the codebook radius is $m=6$; that is, the number of lattice codewords is 833[3]. The bit rate of sequence $\{Y_i\}$ is 0.43 bits/sample, and the corresponding scaling factor is $c=3.1$.

quantizer in which quantizer levels and quantizer step size are $m$ and $c$, respectively.

An example of a lattice quantization scheme for $l^1$ norm is shown in Figure 1. For given codebook radius $m$ and scaling factor $c$, a LVQ quantization procedure is described as follows.

Stage 1 : Lattice vector quantization(lossy coding)

**step 1.** Normalize each source vector $X$ so that the normalized vector $\hat{X} = X/c$.

**step 2.** If $\hat{X}$ is in the granular region, i.e, $\|\hat{X}\| \le m$, map $\hat{X}$ to the nearest lattice point ; else if $\hat{X}$ is in the overload region, i.e, $\|\hat{X}\| > m$, $\hat{X}$ is rescaled to the outermost surface enclosing the granular region, and then mapped to the nearest lattice point on the outermost surface. Say $\hat{X}$ is mapped to a lattice point $Y$, where the position vector of $Y$ is a lattice codeword.

3. LVQ Lossless coding

In the lossless coding stage, lattice codewords are identified by its norm and a label denoting its relative position on the surface specified by its norm[4]. For example, in Figure 1, the lattice points $Y_1$, $Y_2$, $Y_3, \cdots$, $Y_6$, $Y_7$ and $Y_8$, etc, are identified by $Y_{(0,0)}$, $Y_{(1,0)}$, $Y_{(1,1)}$, $\cdots$ , $Y_{(2,0)}$, $Y_{(2,1)}$ and $Y_{(2,2)}$, etc, respectively. That is, each codeword $Y_i$ can be enumerated into an integer pair $(r_i, l_i)$ such that

$$\{Y_i\} \xrightarrow{EnumerationAlgorithm} \{(r_i, l_i)\}$$

- $r_i$ : norm of the codeword of $Yi$, that is, $r_i = \|Y_i\|$.
- $l_i$ : index denoting $Yi$'s relative position on the surface at $r_i$.

We call $r$ as 'radius' and $l$ as 'index'. Enumeration algorithms can be found in[8][9].

Figure 2 shows the radius sequence and the index sequence distributions for a Laplacian source. A Laplacian source has the highest density at the origin and exponentially decreases from the origin. Hence, the source vectors are most likely to be quantized by the lattice point at the origin, $(r, l) = (0, 0)$. So, the radius sequence usually contains not only high frequencies of zeros but also long consecutive zeros between the non-zero values. Figure 2(a) also shows that $r = 0$ has a frequency at least 4 times that of $r = 2$ and over 32 times that of $r = 4$. To exploit the statistical redundancy, we typically apply *the combination of run-length* coding and Huffman coding to radius sequence. At the origin, there exists only one lattice point. Thus, at $r = 0$, the index value must be $l = 0$. This implies that the zero radius value already contains the information of its index value, i.e $l = 0$. So, the

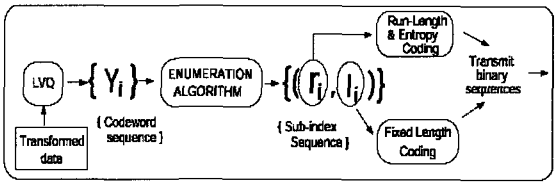



그림 3. LVQ 출력에 가해지는 무손실 코드화 기법
Fig. 3. Lossless coding scheme for LVQ output.

zero index values at $r = 0$ do not need to be transmitted, and Figure 2(b) excludes the zero indices at $r = 0$. On the contrary to radius sequence, as seen in Figure 2(b), no particular index has a significantly high frequency, nor does any long stream exist. This is because the source vectors on each pyramidal or equi-probable surface are uniformly distributed, and at each pyramidal surface the index values are uniformly distributed from 0 to $(C_{r-1})$[3][8]. This decorrelation of index sequence make run-length coding or entropy coding inefficient. Hence the index values are encoded with fixed-length binary bits. The index bit length of lattice points on the surface at $r$ is $\lceil \log_2 C_r \rceil$ bits to cover the index range from 0 to $(C_{r-1})$( $\lceil \cdot \rceil$ is the ceiling function.) Figure 3 illustrates the lossless coding scheme for LVQ outputs. The LVQ lossless coding procedure is also described as follows:

Stage 2 : Lossless coding

**step 1.** Enumerate lattice codeword $Y$ into integer pair $(r, l)$.

**step 2.** Do Run-length coding and Huffman(or Arithmetic) coding to radius sequence $\{r\}$.

**step 3.** Encode index sequence $\{l\}$ by fixed length binary bits in which bit length is $\lceil \log_2 C_r \rceil$ bits.

표 1. Plyramid 평면상과 내에 있는 격자 점들의 수

Table 1. Number of lattice points on or within Pyramids[3].

$D_4$ Lattice

| radius (r) | 0 | 2 | 4 | 6 | 8 | 10 | 12 |
|---|---|---|---|---|---|---|---|
| On Pyramid $C_s(r)$ | 1 | 32 | 192 | 608 | 1408 | 2720 | 4672 |
| Within Pyramid | 1 | 33 | 225 | 833 | 2241 | 4961 | 9633 |

$A_{16}$ Lattice

| radius (r) | 0 | 2 | 4 | 6 | 8 | 10 | 12 |
|---|---|---|---|---|---|---|---|
| On Pyramid $C_s(r)$ | 1 | 0 | 4320 | 61440 | 522720 | 2211840 | 8960640 |
| Within Pyramid | 1 | 1 | 4321 | 65761 | 588481 | 2800321 | 11760961 |

$Z^8$ Lattice

| radius (r) | 0 | 2 | 4 | 6 | 8 | 10 | 12 |
|---|---|---|---|---|---|---|---|
| On Pyramid $C_s(r)$ | 1 | 16 | 128 | 688 | 2816 | 9424 | 27008 |
| Within Pyramid | 1 | 17 | 145 | 833 | 3649 | 13073 | 40081 |

**step 4.** Transmit the variable bit stream of $\{r\}$ and the fixed length codes of $\{l\}$.

**Example 1** : The following example shows the lossless coding procedure of LVQ. The used lattice is $D4$.

1) Consider a the following LVQ output
$Y = \{(-2,0,0,0),(0,-1,-2,1),(0,1,-4,1),$
$(-2,0,-1,3),(-2,1,2,-7),(-1,1,1,5),$
$(-5,-2,-3,0),(2,0,2,0),(0,0,0,0),$
$(1,0,1,0)\}$

2) Applying the enumeration algorithm in[8] to Y :
$\{r\} = \{\|Y\|\} = \{2,4,6,6,12,8,10,4,0,2\}$
$\{l\} = \{30,36,40,420,1999,486,2434,148,0,20\}$

3) Applying Run-length coding and Huffman coding to $\{r\}$[10], the number of bits for encoding the radius sequence : $Bit_r = 35$ bits.

4) From Table I, on the pyramidal surface at $r = 2$, 32 lattice points exist. So, '30' is fixed coded with '11110'( $5bits = \lceil \log_2 32 \rceil$ ). 192 lattice points exist on the surface at $r = 4$, so '36' is encoded with '00100100'( $8bits = \lceil \log_2 192 \rceil$ ) and so on. Remember that $l = 0$ is not transmitted, and excluded in index encoding. With this fixed length coding, the number of index bits is $Bit_l = 82$ bits.

5) So, the total number of bits :
$TotalBits = Bit_r + Bit_l = 117$ $bit.$

## Ⅲ. Analysis of LVQ lossless Coding

In this section, we analyze the performance of LVQ lossless coding for a Laplacian source. Without loss of generality, we assume that the code book radius is $m$, and the lattice is scaled by $c$. Then, the norm of a lattice point is an integral multiple of the scaling factor ; that is, $\|Y\| = c \cdot k$where $k$ is an integer. In practical entropy-constrained quantizers, the codebook size is pre-decided, and the scaling factor must be controlled to determine quantization distortion and bit rate. As the scaling factor decreases, the quantization distortion becomes smaller because the lattice is more finely scaled. However, the bit rate becomes higher since the codeword probabilities become more uniform and the entropy rate increases. So, we derive the bit rate bound of LVQ lossless coding as a function of the scaling factor and compare the actual bit rate.

Let $\Pr(r)$ be the probability that lattice points on the surface at $r$ encode source vectors and $\Pr(r, l)$ be the probability that the lattice point at $(r, l)$ encodes source vectors. Then $\Pr(r) = \sum_{l=0}^{(C_r-1)} \Pr((r, l))$. For example, in Figure 1, $\Pr(r = c)$
$= \Pr(Y_2) + \Pr(y_3) + \Pr(Y_4) + \Pr(y_5)$
$= \Pr((r = c, l = 0)) + \Pr((r = c, l = 1))$

$+\Pr((r=c, l=2))+\Pr((r=c, l=3))$. Since the actual bit rate of radius sequence is not smaller than the entropy rate of the radius sequence, the bit rate bound of an n-dimensional LVQ output is

$$\begin{aligned}&\textit{Bit rate of } \{(r,l)\}\\&=\textit{Bit rate of } \{r\}+\textit{Bit rate of } \{l\}\\&\geq \textit{Entropy rate of } \{r\}+\textit{Fixed bit rate of } \{l\}\\&=\frac{1}{n}\sum_{k=0}^{m}\{-\Pr(ck)\cdot\log_2\Pr(ck)\\&\quad+\Pr(ck)\cdot\lceil\log_2 C_k\rceil\}\end{aligned}$$

The second term of last equation indicates the bit rate of index sequence on the surface at $r=ck$. We consider $Dn$ lattice as an example. $Dn$ lattice points exist on the pyramidal surfaces at only even radii, that is, $r=0,2,4,6,\cdots,m$[3]. So, $\Pr(r)$ for $Dn$ lattice is obtained from gamma function such as :

- At $r=0$(at the origin) :

  $\Pr(r=0)=\int_0^c g(r)\,dr=1-F(c)$.

- At $r=2ck$, $(k=1,2,\cdots,(m-2)/2)$(at granular region) :

  $\Pr(r=2ck)=\int_{c(2k-1)}^{c(2k+1)} g(r)\,dr=F(c(2k+1))-F(c(2k-1))$

- At $r\geq cm$(at overload region) :

  $\Pr(r=cm)=\int_{c(m-1)}^{\infty} g(r)\,dr=F(c(m-1))$.

where the gamma integral $F(r)$ is defined as

표 2. 식(3)의 비트율 경계와 LVR 코드화로 얻어진 실제 비트율의 비교

Table 2. Comparison of bit rate bound from Eq (3) and actually bit rate obtained by actual LVQ lossless coding.

| Scaling factor | 4.1 | 3.4 | 2.8 | 2.4 | 1.9 | 1.6 | 1.0 |
|---|---|---|---|---|---|---|---|
| Lower bound in Eq (3) (bits/sample) | 0.25 | 0.49 | 0.66 | 0.87 | 0.94 | 1.02 | 1.35 |
| Actual bit rate (bit/sample) | 0.25 | 0.50 | 0.75 | 1.00 | 1.25 | 1.50 | 2.00 |

$$\begin{aligned}F(r)&\equiv-\int g(r)\,dr\\&=-\int\frac{\lambda^n}{\Gamma(n)}r^{n-1}e^{-\lambda r}dr=e^{-\lambda}\sum_{k=0}^{n-1}\frac{\lambda^k}{k!}r^k\end{aligned}$$

Plugging each $\Pr(r)$ into Eq (2), we obtain the $Dn$ bit rate bound denoted as $B_L^{D_n}(c)$ :

$$\begin{aligned}&B_L^{D_n}(c)=\\&-\frac{1}{n}(1-F(c))\log_2(1-F(c))\\&-\frac{1}{n}\sum_{k=1}^{(m-2)/2}[(F(c(2k+1))-F(c(2k-1))\cdot\\&\{\log_2(F(c(2k+1))-F(c(2k-1)))-\\&\lceil\log_2 C_{2k}\rceil\}]\\&-\frac{1}{n}F(c(m-1))\cdot\{\log_2 F(c(m-1))+\\&\lceil\log_2 C_m)\rceil\}\end{aligned}$$

Table II compares the bit rate bound in Eq(3) and the actual bit rates for various scaling factors. The input source is the Laplacian source with zero mean, unit variance. The used lattice is $D4$, and the code book radius is $m=38$(that is, the codebook has 772161 lattice codewords). We must note that actual bit rate deviates more from theoretical bound as the bit rates increase. This implies that there is more possibility of improving a LVQ lossless coding at high bit rates. Table III analyzes the bit contributions of radius bit and index bit. The input source is also the Laplacian source used in Table II. As the bit rate increases, the portion of index bits becomes greater than that of radius bits. At bit rates more than 0.8 bits/sample, the portion of index bits dominates. We can explain the phenomenon as following : As the bit rate increases, or the scaling factor becomes smaller, the actual distance scaled by the scaling factor becomes smaller to bring more lattice points near to the origin where the source density is high. So, as bit rate increases the lattice points with large radii must encode source vectors more frequently. Furthermore, as seen in Table I, the range of corresponding index, i.e, $Cr$ exponentially increases with radius. Therefore, the index bit linearly increases with bit rate. However, the entropy rate of radius sequence increases logarithmically with the bit rate due to 'log' in entropy formulation. Hence the radius bit is much less sensitive to bit rate than the index bit. Consequently, the index bit portion hikes as the bit rate increases.

## Ⅳ. Enhanced LVQ Lossless Coding

In this section, we improve the LVQ lossless coding. In section 3, we observed that the performance of lossless coding deviates more from theoretical bound at higher bit rates and the index bit comes to dominate as bit rate increases. From this observation, we can conjecture that the LVQ lossless coding should be improved by reducing the index bits. To reduce the index bit, the large index ranges need to be reduced to smaller ones so that moderate index bit sizes can be obtained over entire bit rates. Therefore, we develop an algorithm $M$ that modifies the original radius and index sequences to operate as :

$$M\{(r_{org}, l_{org})\} = (r_{\text{mod}}, l_{\text{mod}})$$

such that Bit rate of $\{(r_{org}, l_{org})\}$ > Bit rate of $\{(r_{\text{mod}}, l_{\text{mod}})\}$ where $\{(r_{org}, l_{org})\}$, $\{(r_{\text{mod}}, l_{\text{mod}})\}$ are the original sequence and the modified sequence, respectively.

표 3. radius 열과 index 열에 의한 비트분포

Table 3. Bit contributions by the radius sequence and the index sequence.

| Scaling factor (c) | Bit Rate (bit/sample) | Radius Seq. | | Index Seq. | |
|---|---|---|---|---|---|
| | | Bit | Portion (%) | Bit | Portion (%) |
| 4.1 | 0.25 | 0.22 | 88 | 0.03 | 12 |
| 3.4 | 0.50 | 0.34 | 68 | 0.16 | 32 |
| 2.8 | 0.75 | 0.40 | 53 | 0.35 | 47 |
| 2.4 | 1.00 | 0.46 | 46 | 0.54 | 54 |
| 1.9 | 1.25 | 0.52 | 41 | 0.73 | 59 |
| 1.6 | 1.50 | 0.57 | 38 | 0.93 | 62 |
| 1.0 | 2.00 | 0.69 | 34 | 1.31 | 66 |

### 1. Lattice Codeword Enumeration with Set Partitioning

In view of signal modulation, the lattice points on the pyramidal surface at $r$ constructs $Cr$-AM signal constellation. For example, in Figure 1, lattice points at $r=2c$ constructs 8-AM constellation. So, we can adopt the Ungerboeck's set partitioning strategy to enumerate the lattice codewords. Figure 4 illustrates the principle of set partitioning at $r_{org}=r$. The principle is to reduce index ranges by partitioning the index signal set on a pyramidal surface into subsets of which sizes are smaller than the original index set size. We follow the dyadic partition. So, the number of those subsets is a power of 2. The dyadic division makes it possible to represent the subset indices with binary bits. We use the following definitions and notations.

**Definitions and Notations**

- $\mathbf{r}_{org}$ : Set of original radius values, that is $\mathbf{r}_{org}=\{0,1,\ldots,m\}$.
- $\mathbf{l}^{r}_{org}$ : Original index set of lattice points on the pyramidal surface at $r$, that is $\mathbf{l}^{r}_{org}$ = $\{0,1,\ldots,(C_{r-1})\}$.
- $2b_r$( $br$ : positive integer) : Number of partitioned index set on the pyramidal surface at $r$. If $r=0$, $b_r=0$ because only one lattice point exists at $r=0$.
- $\mathbf{l}^{r}_{i}$ $0\le i\le(2^{b_r-1})$ : The $i$th index subset partitioned from $\mathbf{l}^{r}_{org}$ . The elements of $\mathbf{l}^{r}_{i}$ are the indices

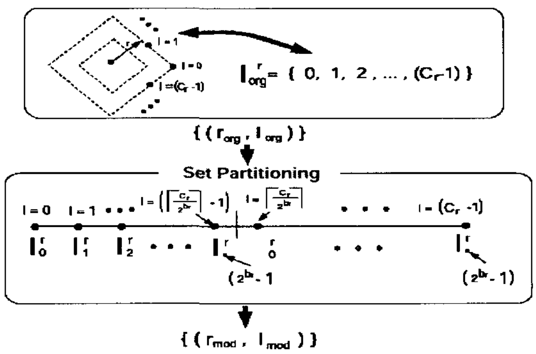



그림 4. 제안한 Set Partitioning 열거 알고리즘

Fig. 4. Set partitioning enumeration. The set of lattice points on the surface at $r$ is $\mathbf{l}^{r}_{org}$. The lattice points are partitioned to subset $\mathbf{l}^{r}_{i}$ where $0\le i\le(2^{b_r}-1)$.

of the $i$th lattice points in every partition. Therefore,

$$\mathbf{I}_i^r = \left\{ i, \left\lfloor \frac{C_r}{2^{b_r}} \right\rfloor + i, \left\lfloor \frac{C_r}{2^{b_r}} \right\rfloor *2+i, \left\lfloor \frac{C_r}{2^{b_r}} \right\rfloor *3+i, \left\lfloor \frac{C_r}{2^{b_r}} \right\rfloor *(2^{b_r}-1)+i \right\}$$

and $\mathbf{I}_{org}^r = \bigcup_{i=0}^{(2^{b_r}-1)} \mathbf{I}_i^r$

- $r_i^r$ : Modified radius value attributed to the partitioned set $\mathbf{I}_i^r$.
- $\mathbf{r}_{\text{mod}}^r$ : Set of modified radius values at $r_{org}=r$, that is $\mathbf{r}_{\text{mod}}^r = \{r_0^r, r_1^r, \cdots, r_{(2^{b_r}-1)}^r\}$. Therefore, the modified value set is $\mathbf{r}_{\text{mod}} = \bigcup_{r=0}^{m} \mathbf{r}_{\text{mod}}^r$.
- $\mathbf{I}_{\text{mod}}^r$ : The set of modified indices at $r_{org}=r$. The modified index values are obtained from $l_{\text{mod}} = \left\lfloor \frac{l_{org}}{2^{b_r}} \right\rfloor$. Thus, $\mathbf{I}_{\text{mod}}^r = \left\{0, 1, 2, \cdots, \left\lfloor \frac{C_r}{2^{b_r}} \right\rfloor \right\}$.

On the surface at $r$, the mapping algorithm $M$ works as follows :

$$\begin{pmatrix} r_{org}=r \\ \mathbf{I}_{org}^r \end{pmatrix} \overset{\text{Set Partitioning}}{\Rightarrow} \begin{Bmatrix} r_0^r, & r_1^r, & r_2^r, \cdots, & r_{(2^{b_r}-1)}^r \\ \mathbf{I}_0^r, & \mathbf{I}_1^r, & \mathbf{I}_2^r, \cdots, & \mathbf{I}_{(2^{b_r}-1)}^r \end{Bmatrix}$$

$$\overset{\text{Modified Sequence}}{\Rightarrow} \begin{pmatrix} \mathbf{r}_{\text{mod}}^r \\ \mathbf{I}_{\text{mod}}^r \end{pmatrix}$$

$C_r$ lattice points from small index to large index are cyclically assigned to $\mathbf{I}_0^r$, $\mathbf{I}_1^r$, $\cdots$, $\mathbf{I}_{(2^{b_r}-1)}^r$. So, the size of subsets are $\left\lfloor \frac{C_r}{2^{b_r}} \right\rfloor$. Then, the mapping algorithm attributes each set $\mathbf{I}_i^r$ with the modified radius value $r_i^r$. The modified radius value informs the relative position of a lattice point in a partition. We obtain the modified index values from $l_{\text{mod}} = \left\lfloor \frac{l_{org}}{2^{b_r}} \right\rfloor$. The modified index value indicates the partition cycle of a lattice point. The modified index values go from 0 to $(\left\lfloor \frac{C_r}{2^{b_r}} \right\rfloor - 1)$. Therefore, the bit length of the modified index values is $\left\lceil \log_2 \frac{C_r}{2^{b_r}} \right\rceil = (\lceil \log 2C_r \rceil - b_r)$ bits, which $b_r$ bits are saved over the bit length of original indices (i.e., $\lceil \log 2C_r \rceil$ bits).

The partitioning procedure generates $2^{b_r}$ modified radius values from one original value and reduce the index range to $\{0, 1, \cdots, (\left\lfloor \frac{C_r}{2^{b_r}} \right\rfloor - 1)\}$ from $\{0, 1, \cdots, (C_r-1)\}$. So, while the partitioning decreases the index bits, it generates more radius values and increases the entropy rate of radius sequence. Therefore, we should determine the modified index set sizes, or $b_r$ such that the index bit reduction will surpass the entropy rate increment on radius sequence. In the following, we analyze how the set partitioning affects the performance of LVQ lossless coding and then establish the rule deciding subset sizes. The following theorem proves that the partitioning does not change the theoretical bit rate bound.

**Theorem** : The index bit reduction by the set partitioning compensates the entropy rate increment on radius sequence. Therefore, the partitioning does not change bit rate bound of LVQ lossless coding.

*Proof* : Again, let $\Pr(r)$ be the probability that lattice points in $\mathbf{I}_{org}^r$ encode source vectors, and $\Pr(r_i)$ be the probability that lattice points in $\mathbf{I}_i^r$ encode source vectors. Then, $\Pr(r) = \sum_{i=0}^{(2^{b_r}-1)} \Pr(r_i)$. Probabilities that lattice points on the same pyramidal surface encode source vectors are equal. For example, in Figure 1, Pr(Y2) = Pr(Y3) = Pr(Y4) = Pr(Y5). Thus, $\Pr(r_i)$ is directly proportional to the size of the subset $\mathbf{I}_i^r$ over the size of $\mathbf{I}_{org}^r$, that is $\Pr(r_i) = \frac{\Pr(r)}{2^{b_r}}$. Then the theoretical bit rate bound of modified sequence can be obtained as

Bit rate bound of $\{(r_{\text{mod}}, l_{\text{mod}})\}$ =Entropy rate of $\{r_{\text{mod}}\}$ + Fixed rate of $\{l_{\text{mod}}\}$

$$= \sum_{r=0}^{m} \sum_{i=0}^{(2^{b_r}-1)} \left\{ -\Pr(r_i) \cdot \log_2 \Pr(r_i) + \Pr(r_i) \cdot \left\lceil \log_2 \frac{C_r}{2^{b_r}} \right\rceil \right\}$$

$$= \sum_{r=0}^{m} \sum_{i=0}^{(2^{b_r}-1)} \left\{ \frac{-\Pr(r_i)}{2^{b_r}} \cdot \log_2 \frac{\Pr(r_i)}{2^{b_r}} + \frac{\Pr(r_i)}{2^{b_r}} \cdot \left\lceil \log_2 \frac{C_r}{2^{b_r}} \right\rceil \right\}$$

since

$$\left\lceil \log_2 \frac{C_r}{2^{b_r}} \right\rceil = \lceil \log_2 C_r - b_r \rceil = \lceil \log_2 C_r \rceil - b_r$$

for integer $b_r$,

$$= \sum_{r=0}^{m} \{ -\Pr(r) \cdot \log_2 \Pr(r) + \Pr(r) \cdot \lceil \log_2 C_r \rceil \}$$

= Entropy rate of $\{r_{org}\}$ + Fixed rate of $\{l_{org}\}$

= Bit rate bound of $\{(r_{org}, l_{org})\}$.

In the third and the fourth equations, the entropy rate increment due to partitioning is compensated by the index bit reduction. This proves the theorem.

The above theorem implies that the subset set size does not change the theoretical bit rate. So, we decide the subset sizes from heuristic experiments. We restrict the index range to be larger than 16 and smaller than 512. A modified index range larger than 16, i.e $C_r > 16$, prevents too many modified radius values, and the index range smaller than 512, i.e $C_r < 512$, precludes too large index ranges. Thus, no partitioning is performed for the index range smaller than 16. We also augment $b_r$ by 1 at consecutive surfaces. It avoids too many radius values generated at one surface. So, if the original index bit size at current radius is larger than the bit size at previous radius, $b_r$ is increased by 1. Finally if the modified index range is larger than 512, we partition the original index range for the modified index range to be 512. Therefore, the rule deciding $b_r$ is :

$$b_r = \begin{cases} 0 & \text{if } C_r \le 16 \\ b_{(r-1)}+1 & \text{if } \lceil \log_2 C_r \rceil > \lceil \log_2 C_{(r-1)} \rceil \text{ and } \left\lceil \log_2 \frac{C_r}{512} \right\rceil \le b_{(r-1)}+1 \\ b_{(r-1)} & \text{if } \lceil \log_2 C_r \rceil \le \lceil \log_2 C_{(r-1)} \rceil \text{ and } \left\lceil \log_2 \frac{C_r}{512} \right\rceil \le b_{(r-1)}+1 \\ \left\lceil \log_2 \frac{C_r}{512} \right\rceil & \text{if } \left\lceil \log_2 \frac{C_r}{512} \right\rceil > b_{(r-1)}+1 \end{cases}$$

At $r=0$, $b_{(r=0)}=0$ because only one lattice point exists at the origin. From Table 1, in the case of $D_4$ lattice, $b_{(r=0)}=0$, $b_{(r=2)}=1$, $b_{(r=4)}=2$, $b_{(r=6)}=3$, $\cdots$ and again for $A_{16}$ lattice, $b_{(r=0)}=b_{(r=2)}=0$, $b_{(r=4)}=4$, $b_{(r=6)}=7$, $b_{(r=8)}=10$, $b_{(r=10)}=13$, $b_{(r=12)}=15$, $\cdots$.

2. Constructing Mapping Algorithm $M$

The lossless coding length coding applies run-length coding and entropy coding to radius sequence. Run-length coding produces two kinds of symbol sequences, 'Symbol-1' and 'Symbol-2'. 'Symbol-1' contains 'Run-length' values and 'Size' values. 'Run-length' is the number of consecutive-zero radius values preceding non-zero values, and 'Size' is the number of bits required to encode the amplitude of the non-zero radius values. 'Symbol-1' sequence is then entropy coded by Huffman coding or Arithmetic Coding. In 'Symbol-2', the amplitudes of the non-zero radius values are just represented by *two's complemented* binary number. The final bit stream combines the entropy coded variable bit stream and the two's complemented binary number bit stream. Consequently, we must minimize the bit length of the two's complementary radius value binary number. In other words, we must minimize the maximum absolute value of modified radii.

To minimize the maximum absolute value of modified radii, the modified radius values should be consecutive integers. For example, if there are 7 modified radius values, the most efficient modified radius set is $r_{\text{mod}} = \{-3,-2,-1,0,1,2,3\}$. So, if a current modified radius set is $r_{\text{mod}} = \{r_i\}$, the next modified radius values is generated as follows :

If $|\min\{r_i\}| \ge \max\{r_i\}$, $r_i^{next} = \max\{r_i\}+1$;
Else $r_i^{\text{next}} = \min\{r_i\}-1$;

For instance, if $r_{\text{mod}} = \{-2,-1,0,1,2\}$, $r_i^{\text{next}}=3$ since $|\min\{r_i\}=-2| = \max\{r_i\}=2$, again if $r_{\text{mod}} = \{-1,0,1,2\}$, $r_i^{\text{next}}=-2$ since $|\min\{r_i\}=-1| < \max\{r_i\} = 2$.

The modified radius values carry on the information about original radius $r_{org}$, subset size $b_r$ and the lattice point position in a partition $i$. In receiver side, we extract $r_{org}$, $b_r$ and $i$ from $r_{mod}$ and recover $lo$ from $l_{org}=2^{b_r} * l_{mod}+i$. Therefore, we need to save the table mapping $\{r_{org}\}$ to $\{r_{mod}\}$ in the receiver side. Since we know the mapping rule in advance, the table does not need to be transmitted so as not to increase side information.

표 4. $D_4$ 격자에 대한 Set Partitioning Mapping 표

Table 4. Set Partitioning Mapping Table for. $D_4$ lattice[3].

Encoding Table

| $r_{org}$ | $1^r_i$ | $r_{mod}$ | $l_{mod} = \lfloor l_{org}/2^{b_r} \rfloor$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 2 | $1^2_0=\{0,2,4,\cdots,30\}$ | 1 | $1^2_{mod}=\{0,1,2,3,\cdots,14,15\}$ |
| 2 | $1^2_1=\{1,3,5,\cdots,31\}$ | -1 | |
| 4 | $1^4_0=\{0,4,8,\cdots,188\}$ | 2 | $1^4_{mod}=\{0,1,2,3,\cdots,46,47\}$ |
| 4 | $1^4_1=\{1,5,9,\cdots,189\}$ | -2 | |
| 4 | $1^4_2=\{2,6,10,\cdots,190\}$ | 3 | |
| 4 | $1^4_3=\{3,7,11,\cdots,191\}$ | -3 | |
| 6 | $1^6_0=\{0,8,16,\cdots,600\}$ | 4 | $1^6_{mod}=\{0,1,2,3,\cdots,74,75\}$ |
| 6 | $1^6_1=\{1,9,17,\cdots,601\}$ | -4 | |
| 6 | $1^6_2=\{2,10,18,\cdots,602\}$ | 5 | |
| 6 | $1^6_3=\{3,11,19,\cdots,603\}$ | -5 | |
| 6 | $1^6_4=\{4,12,20,\cdots,604\}$ | 6 | |
| 6 | $1^6_5=\{5,13,21,\cdots,605\}$ | -6 | |
| 6 | $1^6_6=\{6,14,22,\cdots,606\}$ | 7 | |
| 6 | $1^6_7=\{7,15,23,\cdots,607\}$ | -7 | |

Decoding Table

| $r_{org}$ | $r_{mod}$ | $l_{org}$ |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 2 | $2*l_{mod}$ |
| 1 | 2 | $2*l_{mod}+1$ |
| 2 | 4 | $4*l_{mod}$ |
| 2 | 4 | $4*l_{mod}+1$ |
| 3 | 4 | $4*l_{mod}+2$ |
| 3 | 4 | $4*l_{mod}+3$ |
| 4 | 6 | $8*l_{mod}$ |
| 4 | 6 | $8*l_{mod}+1$ |
| 5 | 6 | $8*l_{mod}+2$ |
| 5 | 6 | $8*l_{mod}+3$ |
| 6 | 6 | $8*l_{mod}+4$ |
| 6 | 6 | $8*l_{mod}+5$ |
| 7 | 6 | $8*l_{mod}+6$ |
| 7 | 6 | $8*l_{mod}+7$ |

Table Ⅳ shows some examples of the $D_4$ lattice mapping table.(Note that $D_4$ lattice points do not exist on the surfaces at odd radii, i.e $r=1,3,5,\cdots$.) At $r_{org}=2$, there exist $C_r=32$ lattice points(see Table Ⅰ). So, $1^{r=2}_{org}=\{0,1,\cdots,31\}$. $1^r_{org}$ is partitioned into $1^r_0$, $1^r_1$, which implies $b_{r=2}=1$. The mapping algorithm then attribute $1^r_0$, $1^r_1$ with $r_0^{r=2}=1$, $r_1^{r=2}=-1$, respectively. Thus, $(r_{org}, l_{org})=(2,5)$ is encoded to $(r_{mod}, l_{mod})=(-1,2)$. In decoding, from $r_{mod}=-1$, we can extract $r_{org}=2$ and $b_r=1$. So, $l_{org}=2^{b_r} * l_{mod}+1=2*2+1$. The binary representation of $l_{org}=5$ at $r_{org}=2$ requires 5 bits since the size of $1^{r=2}_{org}$ is $C_r=32$. But the corresponding modified index $l_{mod}=2$ at $r_{mod}=-1$ can be represented with 4 bits because the size of $1^{r=2}_{mod}$ is 16. Likewise, the original sequence { $(r_{org}, l_{org})$} = { (4, 12), (4, 191), (6, 18), (6, 607) } is modified to { $(r_{mod}, l_m)$} = { (2, 3), (-3, 47), (5, 2), (-7, 75) }.

Example 2 : Consider the same output of Example 2:

1) From Example 1

$\{r_{org}\}=(2,4,6,6,12,8,10,4,0,2)$

$\{l_{org}\}=(30,36,40,420,1999,486,2434,148,0,20)$

2) Applying the mapping table to $(r_{org}, l_{org})$ :

$\{r_{mod}\}=(1,2,4,-5,27,11,-19,0,1)$

$\{l_{mod}\}=(15,9,5,52,79,132,47,52,0,10)$

3) Applying RLE and Huffman coding to $\{r_{mod}\}$, the bits for $\{r_{mod}\}$ is $Bit_{r_{mod}}=46$ bits

4) As seen in Table Ⅳ, the set size of $l_{mod}$ at $r_{mod}=1$ is 15. So, '15' is fixed coded to '1111' (4 bits = $\lceil \log_2 15 \rceil$ ). Again, the set size of $l_{mod}$ at $r_{mod}=2$ is 47. So, '9' is coded to '001001' (6 bits = $\lceil \log_2 47 \rceil$ ). The size of $l_{mod}$ at $r_{mod}=4$ is 75. So '5' is coded to '0000101' (7 bits = $\lceil \log_2 75 \rceil$ ), and so on. With this fixed coding, the number of bits by $l_{mod}$ sequence is $Bits_{l_{mod}}=57$ bits.

5) So, the total bits of modified sequence : $Total\ Bits = Bit_{r_{mod}} + Bit_{l_{mod}} = 103$ bits.

Hence, the mapping algorithm saves 14 bits over the conventional LVQ lossless Coding in Example 1 that does not use the mapping

## Ⅴ. Experimental Results

We applied the proposed LVQ lossless coding method to the Laplacian source used in Table Ⅲ. Table V shows improvement. The proposed lossless coding method does not touch quantization process. Thus, in Table V, at same scaling factors, the quantization distortions for both the proposed coding scheme and the conventional coding scheme are same, but their bit rates are different. Table VI also analyzes the performances of the proposed lossless coding method. Compared to the conventional coding scheme, the proposed coding scheme increases the radius bits, but decreases the index bits. Obviously, the reduction of index bit surpasses the increment in radius bit and so the overall bits produced by the proposed coding scheme must be smaller that those of the conventional coding scheme. Since the index sequences are of higher magnitude at higher bit rates, the proposed coding scheme saves more bits at higher bit rates.

We also applied the proposed lossless coding scheme to a wavelet based image coding. Coding results are given in Table Ⅶ. The optimal scaling factors for LVQs at each subband are decided by the procedure in[3]. The bit rates are obtained from the actual sizes of binary files, not from the entropy calculation. The coding results shows that the proposed lossless coding improves 15%~20% of coding performance at bit rates more than 0.25 bits/pixel. Since the proposed lossless coding scheme does not affect quantization process, the coding improvement by the proposed method must be achieved for any lattice quantization method.

표 5. 제안한 Mapping 방법을 이용한 LVQ 비손실 부호화의 Mapping을 사용하지 않은 표3에 있는 기존의 LVQ 비손실 부호화에 대한 부호화 성능향상

Table 5. Performance improvement of the LVQ ossless coding using the proposed mapping method over the conventional LVQ lossless coding in Table Ⅲ without the mapping procedure.

| Scaling factor | 4.1 | 3.4 | 2.8 | 2.4 | 1.9 | 1.6 | 1.0 |
|---|---|---|---|---|---|---|---|
| Proposed Method | 0.25 | 0.50 | 0.69 | 0.91 | 1.00 | 1.12 | 1.55 |
| Conventional Method | 0.25 | 0.50 | 0.75 | 1.00 | 1.25 | 1.50 | 2.00 |
| Saved bit rate | 0 | 0 | 0.06 | 0.09 | 0.25 | 0.38 | 0.45 |

표 6. 제안된 Mapping 방법을 사용한 LVQ 비손실 부호화의 성능분석

Table 6. Performance analysis of the LVQ lossless coding scheme using the proposed mapping method.

| Scaling factor (c) | Bit Rate (bit/sample) | Radius Seq. Bit | Radius Seq. Portion (%) | Index Seq. Bit | Index Seq. Portion (%) |
|---|---|---|---|---|---|
| 4.1 | 0.25 | 0.22 | 88 | 0.03 | 12 |
| 3.4 | 0.50 | 0.34 | 68 | 0.16 | 32 |
| 2.8 | 0.69 | 0.40 | 58 | 0.29 | 42 |
| 2.4 | 0.91 | 0.46 | 52 | 0.44 | 48 |
| 1.9 | 1.00 | 0.52 | 54 | 0.46 | 46 |
| 1.6 | 1.12 | 0.57 | 52 | 0.54 | 48 |
| 1.0 | 1.55 | 0.69 | 54 | 0.72 | 46 |

표 7. 기존의 LVQ 비손실 부호화와 제안된 Mapping 방법을 사용한 LVQ 비손실 부호화기와의 부호화 성능비교

Table 7. Coding performance comparisons of the conventional LVQ lossless coding and the LVQ lossless coding using the proposed mapping method.

| 8 bpp 512×512 'Lena' | | | 8 bpp 512×512 'Goldhill' | | |
|---|---|---|---|---|---|
| PSNR (dB) | Conventioanl (bits/pixel) | Proposed (bits/pixel) | PSNR (dB) | Conventioanl (bits/pixel) | Proposed (bits/pixel) |
| 32.92 | 0.21 | 0.20 | 29.52 | 0.20 | 0.20 |
| 33.83 | 0.28 | 0.25 | 30.31 | 0.26 | 0.25 |
| 34.46 | 0.34 | 0.30 | 30.83 | 0.32 | 0.30 |
| 35.13 | 0.39 | 0.34 | 31.59 | 0.38 | 0.35 |
| 35.74 | 0.43 | 0.37 | 31.96 | 0.46 | 0.41 |

Therefore, in Table Ⅶ, the coding improvement is important rather than the PSNR values. Figure 5 illustrates the perceptual performances.

## Ⅵ. Conclusion

We have analyzed the performance of LVQ lossless coding. Based on the analysis, we have observed that the index bits at moderate and high bit rates are too large and deteriorate the coding performance. To reduce the index bits, we have proposed a novel lattice enumeration algorithm. The algorithm adopts the set partitioning method[6] to partition large index sets into the subsets of which index sizes are smaller than the original index sizes. The reduced index ranges leads to reduction of the index bits. We have also proved that the set partitioning does not change the theoretical bit rate of the LVQ lossless coding. So, we have determined the hueristical criterion for deciding the proper sizes of partitioned index subsets.

Finally, we have shown that the proposed method improves 15%~20\% of coding performance at bit rates more than 0.25 bits/pixel for a wavelet based image coding and achieves more improvement at higher bit rates.

그림 5. 비트율에 따른 시각적 부호화 성능
Fig. 5. Perceptual performances : Original 'Lena' (upper left), PSNR=34.46 dB at 0.30 bpp (upper right), PSNR=32.37 dB at 0.18 bpp (lower left), PSNR=29.62 dB at 0.09 bpp (lower right).

## References


[1] J. Gibson and K. Sayood, "Lattice quantization", Advances in Electronics and Electron Physics, vol. 72, pp. 259~330, Jun. 1988.

[2] Z. M. Yusof and T.R. Fischer, "An Entropy-Coded Lattice Vector Quantizer for Transform and Subband Image Coding", IEEE Trans. on Image Processing, vol. 5, pp. 289~298, Feb. 1996.

[3] M. Barlaud and P. Solé and M. Antonini and P. Mathieu, "Pyramidal Lattice Vector Quantization for Multiscale Image Coding", IEEE Trans. on Image Processing, vol. 3, pp.367~380, Jul. 1994.

[4] J. M. Moureaux and M. Antonini and M. Barlaud, "Lattice vector quantization of image using a product code form and a new labelling method", SPIE, vol. 2308, no. 1, pp. 422~433, Sep. 1994.

[5] F. Chen and Z. Gao and J. Villasenor, "Lattice Vector Quantization of Generalized Gaussian Sources", IEEE Trans. on Inform. Theory, vol. 43, no. 1, pp. 92~103, Jan. 1997.

[6] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals", IEEE Trans. on Inform. Theory, vol. 28, no. 1, pp. 55~67, Jan. 1982.

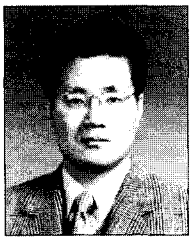
[7] K. A. Birney and T. R. Fischer, "On the Modeling of DCT and Subband Image Data for Compression", IEEE Trans. on Image Processing, vol. 4, no. 2, pp. 186~193, Feb. 1995.

[8] T. R. Fischer, "A Pyramid Vector Quantizer", IEEE Trans. on Inform. Theory, vol. 32, pp. 568~583, Jul. 1986.

[9] A. C. Hung and E.K. Tsern and T.H. Meng, "Error-resilient Pyramid Vector Quantization for Image Compression", IEEE Trans. on Image Processing, vol. 7, no. 10, pp. 1373~1386, Oct. 1998.

[10] G. K. Wallace, "The JPEG Still Picture Compression Standard", Communications of the ACM, vol. 34, no. 4, pp. 30~44, Apr. 1991.

## 저 자 소 개



金 元 河(正會員)

1985년 2월 : 연세대학교 전자공학과 학사. 1988년 5월 : 미 University of Wisconsin-Madison 전기공학 석사. 1997년 5월 : 미 University of Wiscon-Madison 전기공학 박사. 1996년 1월~9월 : 미 Motorola Software Enterprise at Schumbury, IL 소프트웨어 엔지니어. 1997년 8월~2000년 2월 : 미 Los Alamos 국립 연구소 연구원. 2000년 3월~현재 : 명지대학교 전자정보통신공학부 조교수. <주관심분야> 멀티미디어 데이터 압축 및 통신, Multi-Carrier 변조 시스템, Wavelet 변환 응용