Ph.D. Dissertation Summary

New Lattice Vector Quantization Algorithms and Their Applications to Video and Speech Coding

László Lois

Supervised by
Dr. Antal Pálinszki

University of Technology and Economics Budapest
Department of Telecommunications
Budapest

2003

Introduction

The encoding algorithms for different sources such as image, video, audio and speech play an important role to reduce the transmission and storage cost. The early systems were based only sampling and scalar quantization. The idea of scalar quantization is similar to the one described above. We categorize similar values together and give them a name or an 'index' or a 'label'. When we want to send these values we send the index rather than the whole value or vector itself. This idea can be applied to audio, video data whenever the receiver of the information is human. This idea is easily generalized to vector quantization, only the word 'value' has to be replaced with 'vector'. The scalar quantization can also be replaced with vector quantization in the most of the applications.

The vector quantization has several advantages over the scalar quantization: block coding, the number of bits per one coordinate could be fractional, the distortion measure could be also multi-dimensional etc. But the vector quantization could be disadvantageous since finding the closest codepoint in the codebook could require high computational complexity at higher bit rates and the design of the codebook is also difficult. To solve these problems several vector quantization schemes are developed resulting smaller codebook search complexity and easier design, but these simplifications usually lead to worse performance.

There are other methods that could achieve or approach the advantages of the vector quantization i.g. the transform coding or lattice vector quantization. But these methods usually have regular structure hence they must be further complete with supplementary algorithm to handle a specific sources.

Research objectives

A key contribution of the work present here is to invent new encoding systems for multi-dimensional digital signal compressing using lattices [1] to improve the performance and the applicability of the vector quantization (VQ) [2].

On the field of lattice vector quantization (LVQ) [3] I developed new counting, quantizing and labeling algorithms for several constrained lattices such as the cubic lattice and lattice defined by $\bigcup_{i=0}^{N-1}\left(\underline{b}_i + s \cdot D_K\right)$ which is the union of the cosets of the scaled root lattice D_K and L of binary translation vectors where the D_K lattice is the set of K-dimensional integer coordinate vectors having even L_1 norm and the integer scaling factor s is larger than 1.

The new proposed algorithms can handle the following constrains:

- the range of coordinate values is limited
- the algorithm should avoid those vectors which have one or more zero coordinate values.

Several labeling, quantizing and counting algorithms exist [3][4][5][6][7][8], but they could not take these constraints into consideration. The new algorithms are listed below:

- new lattice quantizing algorithm for nonzero coordinate lattice vectors which can handle the cubic lattice and the lattice defined by $\bigcup_{i=0}^{N-1}\left(\underline{b}_i + s \cdot D_K\right)$
- new counting formula for cubic lattice pyramids with limited coordinate range where the limitation is also contains the omission of special coordinate values such as the zero
- new counting formulas and new pyramidal labeling algorithm for lattice $\bigcup_{i=0}^{N-1}\left(\underline{r}_i + s \cdot D_K\right)$ with limited coordinate range or the omission of zero

The new lattice quantizing and labeling algorithms are used to transcode MPEG-1 video streams to lower bit rates by re-quantizing the AC coefficients. The new transcoder uses a new ternary symbol significance map encoder to arrange the significant AC coefficients into vectors where the significance is defined by a threshold. The new algorithm uses pyramidal multi-stage LVQ with the 16 dimension Barnes-Wall lattice. The labeling algorithm checks whether the lattice vector has only nonzero coordinates or not, and this decision is encoded with a zero flag bit and the nonzero or general labeling algorithm is used according to the flag bit. The radius of the lattice vectors, the scaling factors of the multi-stage quantizer and the zero flag bits are encoded by an adaptive arithmetic encoder.

I showed that the PSNR of the output of the new multi-stage Pyramidal Lattice Vector Quantization scheme is tipically 0.5 dB-1.0 dB better than the PSNR of the output of the scalar quantization based MPEG-1 transcoder.

On the field of LSF speech parameter encoding I developed a new transform coding schemes for quantization of the speech LSF parameters. The first algorithm is an improved hybrid transform coding VQ scheme for quantization of the transformed LSF parameters with weighted Euclidean distance (WED) measure. In the second algorithm, the VQ is replaced by a multi-scale LVQ (MSLVQ). I showed that in the encoder the transform-prediction-MSLVQ scheme is better than the prediction-transform-MSLVQ scheme. Furthermore, I developed some new methods such as the packed indexing, coordinate shrinking and a new analysis-by-synthesis based distance measure.

Based upon the experience, I showed that the coding quality of the new PMSLVQ scheme is at least 0.1 dB better in Log Spectral Distortion sense compared to the old PMSLVQ of [12] and WED based VQ scheme at a bitrate of 18..24 bits/frame which result shows significant improvement.

New Results

Thesis 1: New Lattice Labeling, Counting and Quantizing Algorithms [K2][K5]

The new counting formulas, quantizing and labeling (indexing) methods lead to the family of lattices called D_K^{++}, defined as the union of cosets of the root lattice $s \cdot D_K$ and L of binary translation vectors where the D_K lattice is the set of K-dimensional integer coordinate vectors having even L_1 norm and the integer scaling factor s is larger than 1. The new counting formulas and indexing methods are able to count, quantize and label a subset of a lattice pyramid, where:

- the range is limited in each vector coordinate
- the 0 is not allowed in each vector coordinate (nonzero coordinate vector)

Thesis 1.1: New Lattice Quantizing Algorithm for Nonzero Coordinate Lattice Vectors

The quantization algorithm for the root lattice D_K and the lattice defined by $\bigcup_{i=0}^{N-1}\left(\underline{r}_i + s \cdot D_K\right)$ is developed by Conway and Sloane [4]. These quantization algorithms find the closest lattice point. These algorithms are further developed to find the closest lattice point which has no zero coordinate.

The modified quantization algorithm of the lattice D_K for an arbitrary $\underline{x} \in R^K$ is the following:

- For each coordinate of $\underline{x}$ say $x(i)$ find the nearest odd number $o(i)$ and the nearest even number $e(i)$.
    - If $e(i) = 0$ then let $e(i) = 2$ when $0 \leq x(i)$ or let $e(i) = -2$ for negative $x(i)$.
    - Let $Q(x(i))$ be the nearest of $o(i)$ and $e(i)$ and $B(x(i))$ the farest of $o(i)$ and $e(i)$.
- Compute the radius (L_1 norm) of the resulting vector.
- If the radius is odd:
    - Find the coordinate k which minimizes $(B(x(k))- x(k))^2 - (Q(x(k))- x(k))^2$.
    - Let $Q(x(k))= B(x(k))$
- The result is $Q(\underline{x})$

The modified quantization algorithm of lattice $\bigcup_{i=0}^{N-1}\left(\underline{r}_i + s \cdot D_K\right)$ for an arbitrary $\underline{x} \in R^K$ is based on the quantization algorithm of Conway and Sloane. In this case, the closest lattice points of all cosets are determined, but the internal quantization for the lattice D_K is modified in the coset c as follows:

- For each coordinate of $\underline{x}$ find the nearest odd number $o(i)$ and the nearest even number $e(i)$. If $e(i)= 0$ and the binary translation vector $r_c(i) = 0$ then change $e(i)$ to 2 or -2 according to the sign of $x(i)$. Otherwise $e(i)$ remains unchanged even it's value is 0.
- Compute the radius (L_1 norm) of the resulting vector.
- If the radius is odd:
  - Find the coordinate k which minimizes $(B(x(k))- x(k))^2 - (Q(x(k))- x(k))^2$.
  - Let $Q(x(k))= B(x(k))$
- The nearest point of $\underline{x}$ in the lattice D_K is $Q(\underline{x})$ in the coset c and the candidate vector of the coset c is $s \cdot Q(\underline{x}) + \underline{r_c}$

Thesis 1.2: New Counting Formula for Cubic Lattice Pyramids with Limited Coordinate Range

In this case, the limited coordinate range means that in each coordinate the range has a negative or zero lower limit and a positive upper limit.

Let denote $_{a,j}$ the amplitude of the possible lowest value, and $i_{f,j}$ the possible highest value in coordinate j. Let denote $N(d,m)$ the number of d-dimensional vectors of radius m, where m and d are integers and $d > 0$ and $m \geq 0$.

I proved that

$$N(d,m)=\sum_{i=-\min\{m,i_a\}}^{\min\{m,i_f\}} N(d-1,m-|i|) \qquad (1)$$

I also showed that this formula can be simplified as follows:

$$\begin{aligned}
&N(d,0) = 1\\
&N(d,m) = 0 \quad \text{if } ((d=1) \text{ and } (\max\{i_{a,1},i_{f,1}\}<m)) \text{ or } (m<0)\\
&N(1,m) = 1 \quad \text{if } \min\{i_{a,1},i_{f,1}\}< m \leq \max\{i_{a,1},i_{f,1}\}\\
&N(1,m) = 2 \quad \text{if } m \leq \min\{i_{a,1},i_{f,1}\}\\
&N(d,m) = N(d,m-1) + N(d-1,m) + N(d-1,m-1) - N(d-1,m-1-i_{a,d}) - N(d-1,m-1-i_{f,d}) \quad \text{if } d > 1
\end{aligned} \qquad (2)$$

Furthermore, I showed when the lower limit is zero i.e. nonnegative coordinate vectors are considered with positive coordinate limit, the formula is the following:

$$\begin{aligned}
&N(d,0) = 1\\
&N(d,m) = 0 \quad \text{if } ((d=1) \text{ and } (i_{f,1} < m)) \text{ or } (m<0)\\
&N(1,m) = 1 \quad \text{if } m \leq i_{f,1}\\
&N(d,m) = N(d,m-1) + N(d-1,m) - N(d-1,m-1-i_{f,d}) \quad \text{if } d > 1
\end{aligned} \qquad (3)$$

I also proved that the number of the nonzero coordinate cubic lattice points on a pyramid can be evaluated as

$$
\begin{aligned}
&N(\mathrm{d},m) = 0 && \text{if } ((d{=}1) \text{ and } (\max\{i_{a,1},i_{f,1}\}<m)) \text{ or } (m<0) \\
&N(1,m) = 1 && \text{if } \min\{i_{a,1},i_{f,1}\}< m \leq \max\{i_{a,1},i_{f,1}\} \\
&N(1,m) = 2 && \text{if } m \leq \min\{i_{a,1},i_{f,1}\} \\
&N(d,m) = N(d,m\text{-}1) + 2\cdot N(d\text{-}1,m\text{-}1) - N(d\text{-}1,m\text{-}1\text{-}i_{a,d}) - N(d\text{-}1,m\text{-}1\text{-}i_{f,d}) && \text{if } d > 1
\end{aligned}
\tag{4}
$$

Let $N_n(d,m)$ denote the number of nonnegative cubic lattice points. Let the coordinate limit be symmetric i.e. $-(i_{a,j}-1) \leq x(\mathrm{i}) \leq i_{f,j}-1$ and $i_{a,j}=i_{f,j}$ for an arbitrary lattice vector $\underline{x}$. I also showed that

$$N_{nonzero}(d, m) = 2^{\mathrm{d}} \cdot N_n(d, m - d) \tag{5}$$

where the counting formula for $N_n(d,m)$ is in equation (3).

Thesis 1.3: New Pyramidal Labeling Algorithm for Lattice $\bigcup_{i=0}^{N-1}\left(\underline{r}_i + s\cdot D_K\right)$

The input vector of the labeling algorithm called $\underline{x}$ is a vector of radius M, which is calculated by the fast quantizing algorithm of Conway and Sloane [4], which quantizer algorithm also determines the index of the coset leader binary codeword $\underline{r}_b$. The main steps of the labeling algorithm are the following:

- Splitting:

  In order to trace the labeling problem back to that in [3] the algorithm splits $\underline{x}$ according to $\underline{r}_b$:
  - all coordinates over zeros in binary vector $\underline{r}_b$ are collected into a K_0-dimensional vector $\underline{c}_0$,
  - and similarly all coordinates over ones in $\underline{r}_b$ are collected into a K_1-dimensional vector $\underline{c}_1$,
  where K_0 and K_1 are the numbers of zeros and ones in $\underline{r}_b$, respectively. Hence

$$r(\underline{c}_0)+ r(\underline{c}_1)= r(\underline{c})=m \tag{6}$$

- Transform of the even coordinate vector $\underline{c}_0$:

  This vector contains even integer values hence the siginificant LSB bits can be eliminated as follows:

$$\underline{u}_{0,i} = \frac{\underline{c}_{0,i}}{2}, \tag{7}$$

- Transform of the odd coordinate vector $\underline{c}_1$:

  This vector contains non-zero coordinates only. If an arbitrary non-zero integer coordinate is in the range [-A,B], it could be represented as a sign bit and a value in the range [1,max(A,B)] or in the range [0,max(A,B)-1] by a simply subtraction with 1. With this approach, the K_1-

dimensional vector $\underline{c}_1$ is transformed into K_1 sign bits and a K_1-dimensional vector $\underline{u}_1$ in the following way:

$$\underline{u}_{1,i} = \frac{(-1)^{s_i} \cdot c_{1,i} - 1}{2} \tag{8}$$

I proved that one sign bit is redundant for lattice $\bigcup_{i=0}^{N-1} (\underline{r}_i + 2 \cdot D_K)$ hence one bit in the K_1 length sign bit vector could be eliminated. Let $\underline{s}$ denote the K_1-dimensional sign bit vector.

The radius of each resulting vector:

$$r_0 = r(\underline{u}_0) = \frac{r(\underline{c}_0)}{2} \qquad r_1 = r(\underline{u}_1) = \frac{r(\underline{c}_1) - K_1}{2} \qquad r_0 + r_1 = \frac{m - K_1}{2} \tag{9}$$

- Labeling of the two subvectors:
  - The vector $\underline{u}_1$ is encoded with the non-negative pyramidal lattice encoder.

    $index_1$= nonnegative_labeling_algorithm($\underline{u}_0$)

  - If the original input vector is a non-zero vector, then vector $\underline{u}_0$ is also a non-zero vector, thus the non-zero labeling algorithm is used, otherwise the well-known labeling algorithm of Fisher [5] is adopted.

    $index_0$= general_labeling_algorithm ($\underline{u}_1$)

- Multiplexing: finally, the algorithm has to produce one codeword from $\underline{s}$, $index_0$ and $index_1$.

a) $$code_0 = \sum_{i=0}^{r_0 - 1} N_{general}(K_0, i) \cdot N_{nonnegative}\left(K_1, \frac{m - K_1}{2} - i\right) \tag{10}$$

b) $$code_1 = code_0 + index_0 + N_{general}(K_0, r_0) \cdot index_1$$

c) $$code_2 = 2^{K_1 - 1} \cdot code_1 + \left(\underline{s} \,\&\, \left(2^{K_1 - 1} - 1\right)\right)$$

d) $$label = code_2 + \sum_{a=0}^{b-1} N_{\underline{r}_a}(m)$$

where $N_{\underline{r}}(m)$ is denoted here the number of lattice points of coset of $\underline{r}$ binary codeword and the radius of the vectors is m.

The result of the labeling algorithm is found in *label*. The decoding of the label is the inverse of the encoding. The main steps of the label decoding algorithm are the following:

a) Determine b as follows:

$$for(b=0;\ label \geq \sum_{a=0}^{b} N_{\underline{r}_a}(m);++b);$$

b) The value of b determines $\underline{r}_b$, K_0 és K_1, and the splitting method is also known. The value of $code_2$ is the following:

$$code_2 = label - \sum_{a=0}^{b-1} N_{\underline{r}_a}(m)$$

c) Determine the sign bit vector $\underline{s}$ and $code_1$:

$$s = code_2 \;\&\; \left(2^{K_1-1} - 1\right) \qquad \text{and} \qquad code_1 = \frac{code_2}{2^{K_1-1}}$$

d) Determine the r_0 and compute $code_0$ from r_0 as follows:

$$for(r_0{=}0;\, label \geq \sum_{i=0}^{r_0-1} N_{general}(K_0, i) \cdot N_{nonnegative}\left(K_1, \frac{m-K_1}{2} - i\right);\, {+}{+}r_0);$$

$$code_0 = \sum_{i=0}^{r_0-1} N_{general}(K_0, i) \cdot N_{nonnegative}\left(K_1, \frac{m-K_1}{2} - i\right)$$

e) Determine the label of $\underline{u}0$ and $\underline{u}1$ and decode them:

$$index_1 = \frac{code_1 - code_0}{N_{general}(K_0, r_0)} \Rightarrow \underline{u}_1 = nonnegative\_label\_decoder(index_1, r_1)$$

$$index_0 = (code_1 - code_0) \bmod N_{general}(K_0, r_0) \Rightarrow \underline{u}_0 = general\_label\_decoder(index_0, r_0)$$

f) The split vectors $\underline{u}_0$ and $\underline{u}_1$ are merged according to the splitting.

g) Set the unknown bit of $\underline{s}$ to 0 and compose the vector. If the radius of the result does not equals to m, then set the unknown bit of $\underline{s}$ to 1 and compose the vector again.

Thesis 1.4: New Counting Formulas for Lattice $\bigcup_{i=0}^{N-1}\left(\underline{r}_i + 2 \cdot D_K\right)$

Notice that the number of lattice points on a pyramid with a given radius in lattice $\bigcup_{i=0}^{N-1}\left(\underline{r}_i + 2 \cdot D_K\right)$ can be calculated by calculating the number of possible triplets ($\underline{u}_0$, $\underline{u}_1$, $\underline{s}$) in Thesis 1.3.

Let $N_{\underline{r},\underline{s}}(m)$ denote the number of lattice points belong to the binary translation vector $\underline{r}$ where the sign bits of the odd coordinates equal to $\underline{s}$.

In this case

$$N(m) = \sum_{i=0}^{N-1} N_{\underline{r}_i}(m) \qquad \text{and} \qquad N_{\underline{r}}(m) = \sum_{s=0}^{2^{K_1(\underline{r})-1}-1} N_{\underline{r},\underline{s}}(m) \tag{11}$$

I proved that $N_{\underline{r},\underline{s}}(m)$ can be calculated as follows

1) Determine K_0 and K_1 from $\underline{r}$. Since one sign bit is redundant:
   $dim(\underline{s}) = K_1\text{-}1$.
2) *Transform to the cubic lattice*: the coordinate limit of $\underline{c}$ should be also transformed according to the transformation of $\underline{c}_0$ and $\underline{c}_1$ as described in (7) and (8), respectively.

- *Number of sub-codes*: according to equation (9) when the radius of $\underline{u}_0$ is known, then radius of $\underline{u}_1$ equals to $\frac{m-K_1}{2}$-$r(u_0)$, hence

$$N_{\underline{r},s}(m) = \sum_{i=0}^{\frac{m-K_1}{2}} N_{general}(K_0, i) \cdot N_{nonnegative}\left(K_1, \frac{m-K_1}{2} - i\right) \tag{12}$$

Finally, from equation (11) and (12):

$$N(m) = \sum_{\underline{r} \in \{\underline{r}_0, \underline{r}_1, \ldots, \underline{r}_{N-1}\}} \sum_{s=0}^{2^{K_1(\underline{r})-1}} \sum_{i=0}^{\frac{m-K_1}{2}} N_{general}(K_0, i) \cdot N_{nonnegative}\left(K_1, \frac{m-K_1}{2} - i\right) \tag{13}$$

Furthemore, I proved when there are no coordinate limits or the coordinate limits are symmetric, then the sign bit vector has no role and (13) can be simplified as follows:

$$N(m) = \sum_{\underline{r} \in \{\underline{r}_0, \underline{r}_1, \ldots, \underline{r}_{N-1}\}} 2^{K_1(\underline{r})-1} \cdot \sum_{i=0}^{\frac{m-K_1}{2}} N_{general}(K_0, i) \cdot N_{nonnegative}\left(K_1, \frac{m-K_1}{2} - i\right) \tag{14}$$

I showed that to exclude the lattice vectors having one or more zero coordinates, the equation (13) should be rewritten as follows:

$$N(m) = \sum_{\underline{r} \in \{\underline{r}_0, \underline{r}_1, \ldots, \underline{r}_{N-1}\}} \sum_{s=0}^{2^{K_1(\underline{r})-1}} \sum_{i=0}^{\frac{m-K_1}{2}} N_{nonzero}(K_0, i) \cdot N_{nonnegative}\left(K_1, \frac{m-K_1}{2} - i\right) \tag{15}$$

Finally, the equation (14) can be rewritten for the case of nonzero coordinate lattice points as follows:

$$N(m) = \sum_{\underline{r} \in \{\underline{r}_0, \underline{r}_1, \ldots, \underline{r}_{N-1}\}} 2^{K_1(\underline{r})-1} \cdot \sum_{i=0}^{\frac{m-K_1}{2}} N_{nonzero}(K_0, i) \cdot N_{nonnegative}\left(K_1, \frac{m-K_1}{2} - i\right) \tag{16}$$

Thesis 2: Transcoding of MPEG Video Using the New Lattice Vector Quantization Algorithms [K2][K5]

Transcoding of MPEG video streams to lower bit-rates plays an important role in digital video communications. For instance, imagine a number of clients connected to a video server via transmission links of different capacity in a video on demand system. In the simplest case, all the video films are available on the server at all the necessary bit-rates. However, a more reasonable solution is to store each film in one instance only and to transcode it when needed. Transcoding can be done efficiently either in the pixel and or in the frequency domain.

It is known that the amount of overhead (motion vectors, headers) and that of DC difference bits included in an MPEG stream are almost independent of the bit-rate [9][10]. Therefore, nearly all of these data are preserved in the transcoded bit-stream, only the AC coefficients are subject to vector quantization. The Macro Block (MB) headers may be modified since LVQ is able to zero out DCT blocks, changing the Coded Block Pattern or MB type in this way.

In this work the GOP (Group Of Pictures) remains unchanged, the LVQ (Lattice Vector Quantizer) is applied to MPEG quantized AC coefficients, which are compensated in P and B pictures to avoid drift after reconstruction. For Variable Length Coding (VLC) several adaptive arithmetic encoders are used to achieve the highest compression.

Thesis 2.1: New Significance Map Encoder to Arrange the Significant AC Coefficients into Vectors

The input of LVQ is a vector but the AC coefficients are scalar values. To produce the input vector of the LVQ, the AC coefficients must first be arranged into vectors. The arrangement could be

- Static arrangement: the position of an AC coefficient determines the position in the arranged vector
- Dynamic arrangement: each AC coefficient is labeled as significant or non-significant, and only the significant AC coefficients are scanned and arranged into a vector. Samples with absolute value below the significance threshold are excluded. The decision on significance is encoded as a significance map. The significance map contains a binary value for each AC coefficient which indicates if the coefficient is quantized (significant) or set to zero.

I showed that the dynamic arrangement produces much less vectors hence this method is developed further. The decision on significance is encoded by the significance map encoder and this information is also used to arrange the coefficients into vectors.

I developed a new significance map encoder which works as follows:

- The scanning of the coefficients is horizontal block-by-block, the coefficients of the same position within the neighboring blocks get side-by-side. Samples with absolute value below the significance threshold are excluded. The horizontal scanning arranges the samples of similar statistics, since the index within the block is the same. The scanning begins at the top-left block, and the index is selected in zigzag order. The Y, Cr and Cb components are scanned separately.
- The binary significance map is encoded with ternary symbols as follows:
  - 0 = Zero (non-significant) coefficient.
  - 1 = Non-zero coefficient, last in the block (this block is skipped from now).
  - 2 = Non-zero coefficient, but not the last.
- The ternary symbols are encoded by using an adaptive arithmetic encoder.
- A completely zero MPEG block is marked in CBP (Coded Block Pattern) hence it is skipped by the significance map encoder. Similarly, a completely zero MPEG macroblock is signaled in the MPEG bit stream hence it is skipped by the significance map encoder

Thesis 2.2: New Quantizing and Indexing Methods for Nonzero Coefficient Vectors of AC Coefficients

In this experiments the significance threshold t is chosen as the multiple of the scaling factor λ of the LVQ and the rate/distortion diagrams are evaluated for several λ and $c = \frac{t}{\lambda}$.

I showed that the optimal values of c are between 0.75 and 1 when the zero values are allowed in each coordinate of the lattice vectors. In this case the lattice quantization algorithm of Conway and Sloane is used.

I also observed that the optimal values of c are higher when we quantize to the nonzero coordinate lattice vectors, the optimal value of c is between 1.25 and 1.5 in this case. The quantization algorithm of this case was the new nonzero lattice quantizer algorithm described in Thesis 1.

Although the nonzero quantization is proved inefficient, I studied that the nonzero lattice indexing method is worth to use even 1 bit/vector is necessary to sign that the lattice vector index describes a general or a nonzero lattice point. This flag bit is called zero flag bit. I observed that compared to a general lattice indexing algorithm the overall bitrate is lower when the lattice indexing algorithm sends a zero flag bit to sign if the lattice vector is nonzero or general and the lattice indexing is done according to this information. By distinguishing the vectors with the zero flag bit the bit-rate decreases, since non-zero coordinate lattice pyramid contains fewer points than general lattice pyramid having the same radius.

Thesis 2.3: Multi-stage Pyramidal Lattice Vector Quantization of the Vectors of AC Coefficients by Using the New Lattice Labeling and Quantizing Algorithms

In higher vector dimensions, the length of codeword (the output of the labeling algorithm) of the quantized vector could be higher than the arithmetic precision of the computer (e.g. in 16 dimension the size of the largest possible lattice pyramid can be over 100 bits). Taking the integer arithmetic capability of the computer into consideration, the encoder uses an integer scaling factor to keep the size of the largest possible lattice pyramid below a value depending on the CPU (32 bits in our system).

Using this scaling factor, the vectors having larger radius are quantized coarser, hence the lattice codebook is piecewise uniform. The integer scaling factor s is encoded by an arithmetic encoder. When the scaling factor is larger than 1, multi-stage vector quantization is used and the quantization of the error of the stage is repeated until the scaling factor becomes 1.

I examined the pyramidal and spherical lattice labeling of the vectors and the simple Huffman coding of the vector coefficients. I observed that the coding gain of the pyramidal lattice labeling algorithm was about 7% at $\lambda = 1$, but it exceeds 15% when $\lambda \geq 1$.

I developed several versions of labeling algorithm to allow or deny the zero coefficients and take the coordinate limit into account. These labeling algorithms are entropy coding methods hence produces the same distortion for the same data. The versions are listed below:

Labeling Algorithm 1: No coordinate limit information is sent. Hence the coordinate limit information is neglected at the first stage, but it is calculated from the scaling factor of the previous stage when multi-stage quantization if performed. The only side information is the zero flag bits which is sent to the decoder hence both the nonzero and general indexing algorithms are used.

Labeling Algorithm 2: One coordinate limit for each zigzag position.

This algorithm is based on *Labeling Algorithm 1*, the only difference that one coordinate limit is sent for each zigzag position of the blocks of the picture. This information is the maximum of the amplitude of the positive and negative coordinate limit. This coordinate limit information is used at the first stage, but the coordinate limits of the following stages are calculated from the scaling factor of the previous stage when multi-stage quantization if performed. The zero flag bits are also used and sent as a side information.

Labeling Algorithm 3: Positive and negative limit for each zigzag position.

This algorithm is based on *Labeling Algorithm* 2, the only difference that both of the positive and the negative coordinate limit are sent for each zigzag position of the blocks of the picture. This coordinate limit information is used at the first stage, the coordinate limits of the following stages are calculated from the scaling factor of the previous stage when multi-stage quantization if performed. The zero flag bits are also used and sent as a side information.

Based upon the experience the *Labeling Algorithm 3* produced the worst results. Generally the *Labeling Algorithm 1* was the better i.e. this algorithm produced the smallest number of bits. The *Labeling Algorithm 2* seldom outperforms the *Labeling Algorithm 1* and in these cases the difference was not significant. Hence I establish that the *Labeling Algorithm 1* performs better than the other labeling algorithms.

The *Labeling Algorithm 1* is also compared to the previously known pyramidal lattice labeling algorithm (no zero flag bits, no limits at the higher stages of the multi-stage vector quantizer). I experienced that the gain of this method is 4-6% compared to the previously known pyramidal lattice labeling algorithm.

The summary of the overall multi-stage lattice vector quantizer is the following:

- Quantization:
  - 16 dimension Barnes-Wall lattice is used.
  - Multi-stage lattice vector quantization, the 0 values are allowed during the quantization.
  - Significance threshold $t = \lambda$ i.e. $c = 1$.
- Sent information of one stage:
  - Radius of the vector: encoded by an adaptive arithmetic encoder.
  - Pyramidal label: produced by the *Labeling Algorithm 1*.
  - Scaling factor s: encoded by an adaptive arithmetic encoder.
  - Zero flag bit: encoded by an adaptive arithmetic encoder.
- Significance map encoding: ternary alphabet encoded by an adaptive arithmetic encoder.
- Only the AC coefficients are encoded, the other part of the MPEG bitstream remains unchanged.

I showed that the PSNR of the output of the new multi-stage Pyramidal Lattice Vector Quantization scheme is tipically 0.5 dB-1.0 dB better than the PSNR of the output of the scalar quantization based MPEG-1 transcoder. For videos with long GOP, the PSNR of the proposed algorithm could be more than 2 dB higher than the PSNR of the scalar quantization based MPEG-1 transcoder which is significant improvement in quality.

Thesis 3: New Transform Coding Schemes for Quantization of Speech LSF Parameters [F1][K3]

There has been a number of LSF (Line Spectral Frequency) speech parameter quantizers proposed in engineering literature since 1985. As a result, lots of scalar and vector quantization schemes have been studied to quantize the LSF. The current state of the art LSF encoders use Weighted Euclidean Distance (WED) and split vector quantization. The Log Spectral Distortion (LSD) has been accepted as the best objective measure for speech coding [12-17].

One of the most important property of the LSF parameters is known as the strong correlation between the LSF parameters of adjacent frames (inter-frame correlation) and neighboring LSF parameters within the same frame (intra-frame correlation). The traditional utilization of the inter-frame correlation is the predictive coding, the intra-frame correlation can be exploited by vector quantization and/or transform coding. In contrast to the vector quantization, the known hybrid transform coding-vector quantization scheme has a huge disadvantage that the WED measure should be evaluated in the LSF domain hence the inverse transform should be applied for each codebook vector of the vector quantizer which significantly increases the coding complexity while the coding quality slightly improves.

The first part of this thesis introduces a new hybrid transform coding and vector quantization scheme which improve the coding quality. The second part of this thesis describe a new hybrid transform coding and lattice vector quantizer algorithm using Karhunen-Loeve or Discrete Cosine Transform and Predictive Multi-Scale Lattice Vector Quantization (PMSLVQ) [12] which improve the coding quality significant while the complexity remains low. The last part extends this PMSLVQ scheme with a new LSD-based distortion measure which can be further improve the coding quality but also gears up the complexity.

Thesis 3.1: Improved Hybrid Transform Coding and Vector Quantization Scheme for Quantization of the Transformed LSF Parameters with Weighted Euclidean Distance Measure [F1][K3]

In order to reduce the extremely high computation complexity of the vector quantization even at low bit rates (up to 24 bits/LSF vector), the input vector is segmented into subvectors and the segments then can be quantized independently.

In a hybrid transform coding and vector quantization scheme the encoded vectors are first transformed and then quantized by the vector quantizer. When the distance measure is the Euclidean distance and the transform is orthogonal and linear, the Euclidean distance can also be used in the transform domain and no inverse transform is needed.

Let $\underline{\omega}$ and $\underline{\hat{\omega}}$ denote the original and the decoded LSF vectors. The weighted Euclidean distance is applied to the whole original and reconstructed LSF vectors rather than subvectors as follows

$$WED_{LSF}\left(\underline{\omega},\underline{\hat{\omega}}\right)=\sum_{i=1}^{D} w_i\left(\omega_i-\hat{\omega}_i\right)^2 \tag{3.1}$$

where w_i the i-th weight computed from $\underline{\omega}$ and D is the dimension of the LSF vectors.

I proved that in case of split vector quantization in the transform domain the segments could not be quantized independently any more since the weights of the weighted Euclidean distance is defined in the LSF domain.

To solve this problem I developed the following algorithm. Let the transformed LSF column vector $\underline{u}$ is split into segments $\underline{u}_1, \underline{u}_2, \ldots \underline{u}_S$, so that $\underline{u} = [\underline{u}_1, \underline{u}_2, \ldots \underline{u}_S]$ and hence $\dim(\underline{u})=\sum_{i=1}^{S}\dim(\underline{u}_i)$. To encode the input LSF vector $\underline{\omega}$, first the weight vector is computed from $\underline{\omega}$. Then $\underline{\omega}$ is transformed into the transform domain by using the transform matrix $\underline{\underline{T}}$, hence $\underline{u}=\underline{\underline{T}}\cdot\underline{\omega}$. To compute the distortion between the segment $\underline{u}_s$ and the quantized (or candidate) subvector $\underline{v}_s$, these vectors must be extended into D dimensional vectors to transform them back to the LSF domain since the WED could only be evaluated there. Due to the significance order of the KL coefficients, the subvector which consists of the lower-index transformed coefficients is more important than those which have higher-index transformed coefficients. Thus, the subvector $\underline{u}_s$ and $\underline{v}_s$ are padded with zero from bottom and padded with the $\underline{u}_k$ and $\underline{u}_k$ from top, respectively, where $0 < k < s$. The distortion is then computed as follows:

$$WED(\underline{u}_s, \underline{v}_s) = WED_{LSF}(\underline{\underline{T}}^{-1}[\, \underline{u}_1, \underline{u}_2,\ldots, \underline{u}_s, 0,\ldots, 0\,], \underline{\underline{T}}^{-1}[\, \underline{v}_1, \underline{v}_2,\ldots, \underline{v}_s, 0,\ldots, 0\,]) \tag{3.2}$$

For the comparison purpose, the known formula for evaluating the WED of subvector $\underline{u}_s$ and $\underline{v}_s$ is the following:

$$WED(\underline{u}_s, \underline{v}_s) = WED_{LSF}(\underline{\underline{T}}^{-1}[\, 0,\ldots,0, \underline{u}_s, 0,\ldots, 0\,], \underline{\underline{T}}^{-1}[0,\ldots,0, \underline{v}_s, 0,\ldots, 0\,]) \tag{3.3}$$

Based on the experimental results, the introduction of the Karhunen-Loeve transform improves the quality of the reconstructed LSF frames. The introduction of the WED_{LSF} described in equation (3.1) without transformation also improves the coding quality in LSD. These two results agree with the previous results reported in the literature.

I showed that the degree of improvement of the introduction of the WED in the scheme with no transform can be achieved in the transform coding scheme by introducing the proposed WED described in equation (3.2) resulting the best scheme out of the mentioned schemes in contrast to the previously known WED_{old} described in equation (3.3) which introduce slightly or no improvement in the coding quality.

Thesis 3.2: New New Hybrid Transform Coding - PMSLVQ Scheme for LSF Quantization

The PMSLVQ scheme is developed in [12]. This scheme uses inter-frame prediction and multi-scale lattice vector quantization. The linear transform could be inserted in this scheme in two positions at the encoder side:

- before the prediction
- between the prediction and the quantization

The linear transforms used in the experiments were the Karhunen-Loeve Transform (KLT) and the Discrete Cosine Transform (DCT), while the predictor was an autoregressive predictor. The predictor used in this study has the form:

$$\omega_t^P = \sum_{i=1}^{N} p_i \cdot \hat{\omega}_{t-i} \qquad (3.4)$$

where ω_t^P is the predicted value, $\hat{\omega}_k$ the previous quantized values, N the predictor order and p_i the predictor coefficients. In this study the value of N is set to 4. To encode the 10 dimensional LSF vectors the lattice $D_{10}^{+} = D_{10} \cup \left((½,½,\ldots,½) + D_{10}\right)$ is used.

Based upon the experience, I showed that the introduction of the KLT or DCT results more than 0.1 dB improvement in Log Spectral Distortion sense which is significant improvement. I experienced that the transformation- prediction scheme performed better than the prediction-transformation scheme in Log Spectral Distortion sense but the different was not significant.

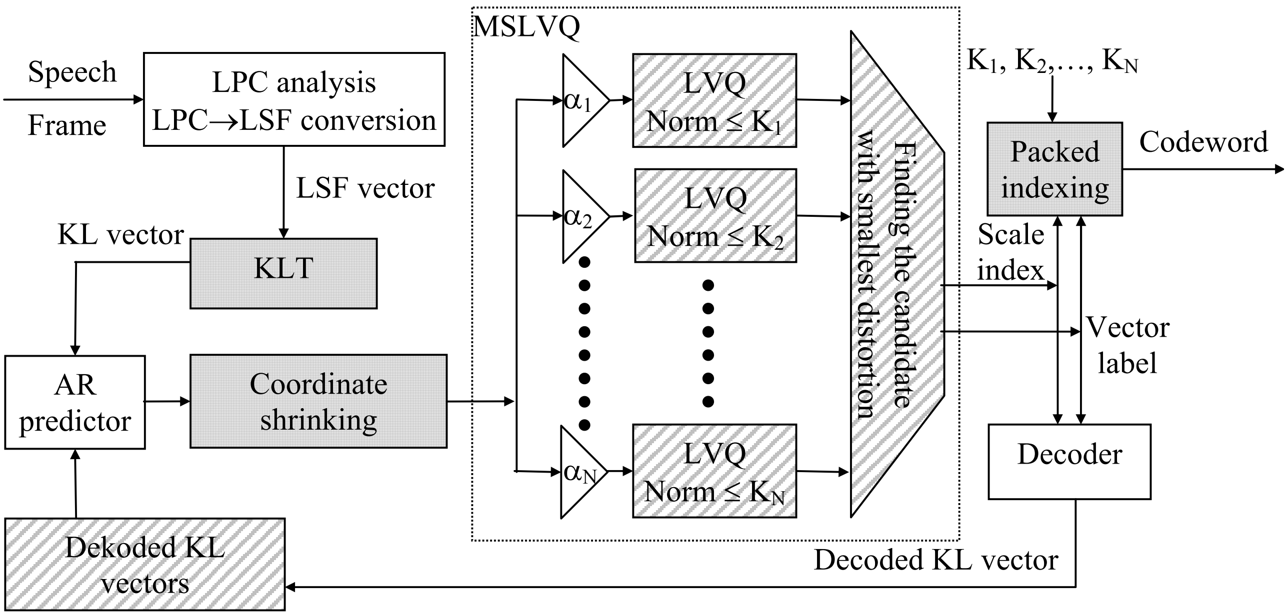

Figure 1. The new KLT-PMSLVQ scheme for transform coding of LSF vectors (the developed parts are shaded gray, the significantly modified parts are striped)

Thesis 3.3: New Hybrid Transform Coding - PMSLVQ Scheme for LSF Quantization with Weighted Euclidean and New Nonlinear Distance Measure

I developed some improvement in the basic hybrid transform coding - PMSLVQ scheme. The new scheme is shown in Figure 1.

Let denote N the number of scales in the MSLVQ. The main improvements of the scheme are the following:

- Coordinate shrinking:

  The input LSF vector is transformed to a KL vector, and then the predictor produces the prediction error of the current KL vector by using the previously decoded KL vectors. The prediction error signal is then quantized. Since the importance of the coordinates in the prediction vector is statistically different, my new algorithm exploits this by multiplying the values of coordinates with shrinking factors. The shrinking factors had been determined empirically on the training data set.

- Different truncation level for each scale and packed indexing:

  The Multi-Scale Lattice Quantizer introduced in [12] works with spherical or pyramidal truncation with the same truncation level at each scale. The truncation level is the highest possible norm of the lattice vector where limitation of the L_1 norm leads to pyramidal truncation and the limitation of the L_2 norm is the spherical truncation of the lattice. The encoder of [12] determines the scale s and the lattice label L at scale s and the resulting codeword is the product code of s and L.

  I proved that the product code is not an optimal bit allocation for and MSLVQ based on lattice D_{10}^{+} for 20 and 22 bits/frame. To improve the efficiency of the bit allocation, I developed a new method for MSLVQ indexing called packed indexing. The packed indexing has the following properties:

  - Let K_i denote the number of lattice points within the scale i in the multi-scale LVQ where $0 \leq i < N$. In the proposed packed indexing algorithm, K_i and K_j could be different when $i \neq j$.
  - Let $S_i = \sum_{k=0}^{i} K_k$, where $0 \leq i < N$ and $S_{-1}=0$
  - Let the index of the resulting scale is n, and the label of the lattice vector is i_{LVQ}, where $0 \leq n < N$ and $0 \leq i_{LVQ} < K_n$
  - Let the packed index i.e. the resulting codeword of the MSLVQ is the following:

    $$packed\_index(n,i) = S_{n-1} + i_{LVQ}$$

  - The decoding algorithm of the packed index is the following:
    - Find the scale index n ($0 \leq n < N$), which satisfies

$$S_{n-1} \leq packed\_index < S_n$$

- If the decoder has determined the value of n, the label of the LVQ can be calculated as follows:

$$i_{LVQ} = packed\_index - S_{n-1}$$

- If the decoder knows i_{LVQ}, it can determine the lattice vector by using the lattice label decoding algorithm.

- New distance measure using analysis by synthesis:

  The nearest lattice vector of $D_{10}^{+} = D_{10} \cup \left((½,½,\ldots,½) + D_{10}\right)$ is determined as follows:

  1. Find the nearest vector in D_{10} on every scale taking the truncation level into account.
  2. Find the nearest vector in (½,½,…,½)+D_{10} on every scale taking the truncation level into account.

  - The result of the upper 2 steps is 2·N of lattice vectors. The quantization error values of the 2·N lattice vector quantizers are scaled with the scaling factor of the corresponding MLVQ scale and the resulting vectors are arranged according to the scaled quantization error.
  - For the lower A of the scaled quantization error values the corresponding decoded LSF vectors are decoded (synthesis) and the final distortion measure is:
    - WED, where A=2·N since the complexity of this distortion measure is small
    - LSD, where A=2,…,2·N

    The proposed method has two new important properties on the field of lattice vector quantization:

    - Every scale produces two candidate vectors for the final decision, not only one.
    - This scheme is able to use the LSD distortion measure as the final decision rule due to the relatively small number of candidate vectors (at least 2·N).

Based upon the experience, I showed that the the coding quality of the new PMSLVQ scheme with transform coding, coordinate shrinking, packed indexing and the new distance measure is 0.1 dB better in Log Spectral Distortion sense compared to the old PMSLVQ algorithm of [12] at a bitrate of 18..24 bits/frame. This result shows significant improvement. I experienced that the KLT performed better than the DCT at bitrates of 20..24 bits/frame, but at very low bit rates (18..19 bits/frame) the DCT outperforms the KLT in Log Spectral Distortion sense. The very low bit rate leads to high quantization error and the KL transform is no more optimal. The spherical truncation of the lattice produced better result in general, but at several bit-rates the pyramidal truncation was the better.

List of publications

Journal Papers

[F1] H. L. Vu, L. Lois (2000), „Efficient Distance Measure for Quantization of LSF and Its Karhunen-Loeve Transformed Parameters”, *IEEE Trans. on Speech and Audio Processing*, Vol. 8, No. 6, pp. 744-746, Nov. 2000.

[F2] L. Lois, H.L. Vu (1999), „Spectral Coding of Speech LSF Parameters Using Karhunen-Loeve Transform”, *IEICE Trans. Fundamentals, Special Section on Information Theory And Its Applications*, Vol. E82-A, No. 10, pp. 2138-2146, October 1999.

[F3] H. L. Vu, L. Lois (1999), „A New General Distance Measure for Quantization of LSF and Its Transformed Coefficients”, *IEICE Trans. Fundamentals, Special Section on Digital Signal Processing*, Vol. E82-A, No. 8, pp. 1493-1500, August 1999

Conference Papers

[K1] L. Lois (2002), „Interaktív műsorszórós alkalmazások megvalósítása a digitális videó műsorszórásban”, *Proceedings of the 10th Conference and Exhibition on Television and Audio Technologies*, Budapest, Hungary, pp. 95-104, May 2002

[K2] L. Lois (2000), „Rácskvantálás alkalmazása a video transzkódolásban”, *Proceedings of the 9th Conference and Exhibition on Television and Audio Technologies*, Budapest, Hungary, pp. 171-181, May 2000

[K3] H. L. Vu, L. Lois (2000), „Optimal Weighting Function for Quantization of LSF and Its Karhunen-Loeve Transformed Parameters”, *Proceedings of the 9th Conference and Exhibition on Television and Audio Technologies*, Budapest, Hungary, pp. 183-191, May 2000

[K4] L. Lois (1999), "Variable Length Coding of Transformed LSF Parameters", *Proceedings of ESCA EUROSPEECH’99 Conference*, Budapest, Hungary, Vol. 3. pp. 1455-1458, 1999

[K5] L. Lois, S. Bozóki (1998), „Transcoding Of MPEG Video Using Lattice Vector Quantization”, *Proceedings of the IEEE International Conference on Image Processing*, Chicago, Illinois, USA, Vol. 2. pp. 341-345, October 1998

[K6] L. Lois, H. L. Vu (1998), „Transcoding Alapú Kommunikáció a Videó- és Beszédkódolásban", *Proceedings of the 8th Conference and Exhibition on Television and Audio Technologies*, Budapest, Hungary, pp. 32-39, May 1998.

[K7] H. L. Vu, L. Lois (1998), „A New General Distance Measure for Quantization of LSF and Their Transformed Coefficients", *Proceedings of IEEE Int. Conf. On Acoustics, Speech And Signal Processing*, pp. 45-48, 1998.

[K8] H. L. Vu, L. Lois (1997), „Spectral Sensitivity of LSP Parameters and Their Transformed Coefficients", *Proceedings of the European Conf. Speech Comm. Technology, EUROSPEECH*, Rhodos, Turkey, pp. 1251-1254, September 1997.

[K9] H. L. Vu, L. Lois (1997), „Optimal Transformation of LSP Parameters Using Neural Network", *Proceedings of IEEE Int. Conf. On Acoustics, Speech And Signal Processing*, pp. 1339-1342, 1997.

[K10] L. Lois, H. L. Vu (1996), „Video and Speech Compression at Low Bit Rate", *Proceedings of the IEEE International Conference on Multimedia Technology and Digital Telecommunication Services*, Budapest, Hungary, pp. 89-96, October 1996.

References

[1] J.H. Conway, N.J.A. Sloane, ”Sphere packing, Lattices and Groups”, Springer Verlag, 1993

[2] N .S. Jayant and P. Noll, "Digital Coding of Waveforms: Principles and Applications toSpeech and Video." Prentice-Hall, 1984.

[3] J.H. Conway, N.J.A. Sloane, ”A fast encoding method for lattice codes and quantizers", *IEEE Trans. Inform. Theory*, pp. 820-824, Nov. 1983

[4] J.H. Conway, N.J.A. Sloane, ”Fast quantizing and decoding algorithms for lattice quantizers and codes”, *IEEE Trans. on Information Theory*, pp. 227-232, 1982

[5] T. R. Fischer, ”A pyramid vector quantizer”, *IEEE Trans. on Information Theory*, pp. 568-583, July 1986

[6] P. Rault, C. Guillemot, “Lattice vector quantization with reduced or without look-up table”, *SPIE Int. Conf. on Visual Communication and Image Processing*, San Jose, CA, Jan. 1998, vol. 3309, pp. 851-862

[7] J.M. Moureaux, P. Loyer, M. Antonini, “Efficient indexing method for Zn and Dn lattice quantizers”, *Proc. of IEEE Int. Conf. on Image Processing*, 1996, Vol. 3. pp. 447-450

[8] J.M. Moureaux, P. Loyer, M. Antonini, “Low complexity indexing method for Zn and Dn lattice quantizers”, *IEEE Transactions on Communications*, 1998, pp. 1602-1609

[9] G. Keesman et al.: “Transcoding of MPEG-2 bitstreams”, *Signal Processing: Image Communications*, Vol. 8, Sept. 1996, pp. 481-500.

[10] P. A. A. Assuncao and M. Ghanbari: “Transcoding of MPEG-2 video in the frequency domain”, *Proc. of IEEE Int. Conf. On Acoustics, Speech And Signal Processing*, 1997, Vol 4

[11] F. Itakura, "Line Spectrum Representation of Linear Predictive Coefficients of Speech Signals", *Journal Acoust. BoG. America*, Vol. 57, pp. S35(A), 1975.

[12] Vasilache, M. Vasilache, I. Tabus, “Predicitive multiple-scale lattice VQ for LSF quantization”, *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Phoenix, May 1999, pp. 657-660

[13] K.K. Paliwal and B.S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 bits/frame", *IEEE Trans. Speech Audio Processing*, Vol.1, No.1, pp. 3-14, January 1993.

[14] R. Laroia, N. Phamdo, N. Farvardin, “Robust and Efficient Quantization of Speech LSP Parameters Using Structured Vector Quantizers”, *Proc. of IEEE Int. Conf. On Acoustics, Speech And Signal Processing*, 1991, pp. 641-644

[15] D. Chang, S. Ann, C. W. Lee, “A Classified Vector Quantization of LSF Parameters”, *Signal Processing*, 1997, pp. 1347-1350

[16] W.R. Gardner, B.D.Rao, „Theoretical analysis of the high-rate vector quantization of LPC parameters”, *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 367-381, 1995

[17] M. Skoglund, J. Skoglund, „On non-linear utilization of intervector dependency in vector quantization”, *Proc. IEEE Int. Conf. On Acoustics, Speech And Signal Processing*, Seattle, USA, 1998