SUBBAND VECTOR QUANTIZATION WITH TRELLIS CODING

Viktor MICHALČIN
Laboratory of Transfer Technology, Institute of Physical Science, Faculty of Science,
P. J. Šafárik University in Košice, Jesenná 5, 040 01 Košice, Slovak Republic,
tel.: +421 55 622 1128-9, e-mail: michalcin@vk.upjs.sk

ABSTRACT

In this paper, a subband coding system consisting a combination of analysis filter bank, vector quantizer and trellis coding is designed. For the analysis filter bank, which decompose an input signal spectrum to bands we used a design method of cosine modulated filter bank with the perfect reconstruction. The analysis filters are obtained by cosine modulation of a prototype filter with the transfer function H(z), which optimization is performed right through the two-channel lossless lattice structure. Next block in this coding system is the vector quantizer with a full search, which perform quantization in the subband domain. The final block of the designed subband coding system is the block of trellis coding. This uses a block structured codebook of the vector quantizer with an extended set of the quantization vectors in accordance with Ungerboeck's theory of clustering the set.

Keywords: *subband coding, vector quantization, trellis coding, prototype filter, cosine modulated filter bank subband system*

1. INTRODUCTION

The subband coding of signals belongs to one of effective compression methods by means of which the high quality of coding at low bit rates is achieved [1-4]. The subband coding system (SBCS) can be divided into two basic subsystems. The first one is the subsystem of analysis/synthesis [5-7], which contains the synthesis filter bank at the receiving side. Codecs [8-10] are the other subsystem, which contain all operations necessary for coding and decoding of subband signals obtained from the analysis/synthesis subsystem. Both parts must be designed very carefully because to imperfection of the implementation in both subsystems decreases a coded signal quality. Under a low bit rates, the characteristics of the analysis/synthesis system much sensitively affect a general distortion of the coded signal. In this paper is designed the subband coded system, that consists of a combination analysis filter bank, vector quantization and trellis coding as shown in Fig. 1.

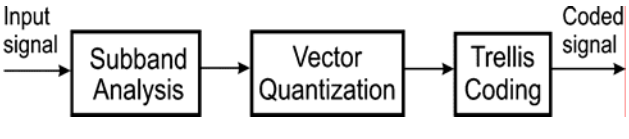


Fig. 1 Subband vector quantization with trellis coding

Next block of the coding system is the vector quantizer (VQ) with full search [11], [12] which realises quantization in the subband domain. Such VQ achieves higher efficiency of quantization the subband signals compared to scalar quantization. Final block of the designed subband coding system is the block of trellis coding. This block makes use of the structural codebook respectively reproduction alphabet of VQ with advanced set of quantization vectors, according to Ungerboeck's theory dividing set [13]. Then, in process of trellis coding [14], [15] the expanded number of quantization vectors is limited according to the required speed of transmission.

2. COSINE MODULATED FILTER BANKS WITH PERFECT RECONSTRUCTION

In general, the analysis filter bank can be expressed in matrix form [16] ($N = 2mK,\ m \geq 1$) as

$$A(z) = \begin{bmatrix} H_0(z) \\ H_1(z) \\ \vdots \\ H_{K-1}(z) \end{bmatrix} = \boldsymbol{C} \begin{bmatrix} G_0\left(-z^{2K}\right) \\ z^{-1}G_1\left(-z^{2K}\right) \\ \vdots \\ z^{-(2K-1)}G_{2K-1}\left(-z^{2K}\right) \end{bmatrix} \tag{1}$$

where K is the number of channels and $\mathbf{C}$ is a K x $2K$ cosine modulation matrix with elements

$$c_{k,\ell} = [\mathbf{C}]_{\mathbf{k},\ell} = $$
$$= 2\cos\left[(2k+1)\frac{\pi}{2K}\left(\ell - \frac{N-1}{2}\right) + (-1)^k \frac{\pi}{4}\right]$$

$$\text{for} \qquad 0 \leq k \leq K-1, \quad 0 \leq \ell \leq 2K-1 \tag{2}$$

Then implementation of the analysis filter bank described by eq. (1) is shown in Fig. 2, where functions $G_q(z)$ are the type 1 polyphase components [17] of a linear-phase prototype filter

$$H(z) = \sum_{q=0}^{2K-1} z^{-q} G_q(z^{2K}) \tag{3}$$

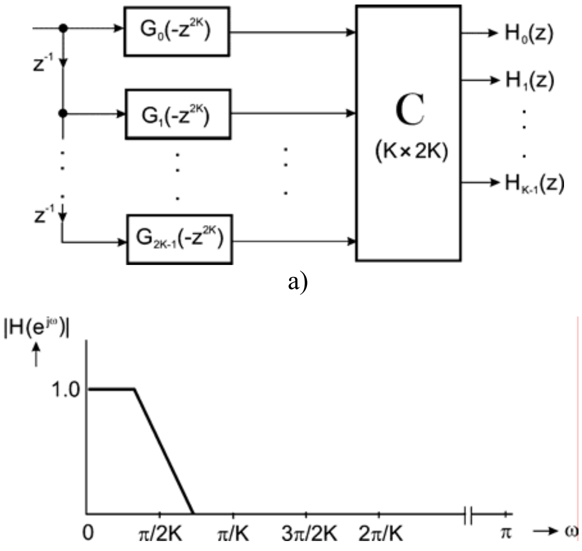
In cosine modulated filter banks, the analysis filters $H_k(z)$ and the synthesis filters $F_k(z)$ are obtained by the cosine modulation of low-pass prototype filter $H(z)$ as follows

$$h_k(n) = 2h(n)\cos\left[(2k+1)\frac{\pi}{2K}\left(n - \frac{N-1}{2}\right) + (-1)^k \frac{\pi}{4}\right]$$
$$0 \leq n \leq N-1 \tag{4}$$

$$f_k(n) = 2h(n)\cos\left[(2k+1)\frac{\pi}{2K}\left(n-\frac{N-1}{2}\right)-(-1)^k\frac{\pi}{4}\right]$$
$$0 \le n \le N-1 \qquad (5)$$

where $h_k(n)$ and $f_k(n)$ are the impulse responses of $H_k(z)$ and $F_k(z)$, respectively. From eq. (4) and (5), we can verify that the analysis and synthesis filters are related as

$$f_k(n) = h_k(N-1-n) \quad \text{and}$$
$$F_k(z) = z^{-(N-1)}\,\tilde{H}_k(z) \quad \text{for } 0 \le k \le K-1 \qquad (6)$$



a)

b)

Fig. 2 a) Polyphase implementation of the cosine-modulated analysis filter bank, b) The desired response of the prototype *H(z)*

The tilde accent on a function $H_k(z)$ means that $\tilde{H}_k(z) = H_k(z^{-1})$. The overall transfer function $T(z)$ of the analysis/synthesis system, has linear phase. However, the filters $H_k(z)$ and $F_k(z)$ do not have linear phase. Using type 1 polyphase decomposition, $A(z)$ can be expressed as

$$A(z) = \boldsymbol{E}(z^K)\boldsymbol{e}_K(z) \qquad (7)$$

where $\boldsymbol{E}(z)$ is the polyphase component matrix of the filter bank and $\boldsymbol{e}_K^T(z) = \begin{bmatrix} 1 & z^{-1} & \dots & z^{-(K-1)} \end{bmatrix}$. The losslessness of the polyphase component matrix is sufficient to ensure that the analysis/synthesis system satisfies perfect reconstruction. $\boldsymbol{E}(z)$ is lossless if and only if appropriate pairs of polyphase components of $H(z)$ are power complementary i. e.

$$\tilde{G}_k(z)\,G_k(z) + \tilde{G}_{K+k}(z)\,G_{K+k}(z) = \frac{1}{2K}$$
$$0 \le k \le K-1 \qquad (8)$$

Since the prototype $H(z)$ is linear phase and its length is $N = 2mK$, we have the following relation between the polyphase components of $H(z)$

$$G_k(z) = z^{-(m-1)}\tilde{G}_{2K-1-k}(z), \quad 0 \le k \le K-1 \qquad (9)$$

Each of pairs $\{G_k(z),\ G_{K+k}(z)\}$ can be designed by using a separable two– channel lossless lattice. For any K, the number of two–channel lossless lattice needed for design of the prototype filter $H(z)$ is $\lfloor K/2 \rfloor$ (where brackets are Gaussian's) because of symmetry given by eq.(9). The k-th lattice yields the power complementary pair $\{G_k(z),\ G_{K+k}(z)\}$. Its parameters are denoted as $\theta_{j,k}$, where the index j refers to the particular lattice section. The transfer function between the input of the lattice and the output of the p-th lattice section is denoted by a superscript p. The lattice transfer functions are initialized as

$$G_k^{(0)}(z) = c_{k,0} = \cos\ \theta_{k,0} \quad \text{and}$$
$$G_{K+k}^{(0)}(z) = s_{k,0} = \sin\ \theta_{k,0}\ ,$$
$$0 \le k \le \left\lfloor \frac{K}{2} \right\rfloor - 1 \qquad (10)$$

A typical four- multiplier implementation of the p-th lattice section of the k-th lattice has the parameters $c_{k,p} = \cos\ \theta_{k,p}$, $s_{k,p} = \sin\ \theta_{k,p}$. If the length of the prototype is $N = 2mK$, then each of $2K$ polyphase components has length m. This implies that each lattice has $(m - 1)$ sections, involving a total m unknown parameters $[\theta_{k,0},\ \theta_{k,1},\ \dots,\ \theta_{k,(m-1)}]$. Hence the total number of parameters to be optimized is $m\lfloor K/2 \rfloor$.

3. FULL SEARCH VECTOR QUANTIZATION

The basic quantization problem considered in the sequel is the following. Let $\boldsymbol{X}$ be a k-dimensional random vector, $\boldsymbol{Q}$ denotes a mapping of a vector quantizer and $\boldsymbol{Y} = \boldsymbol{Q}(\boldsymbol{X})$ its output generated by one quantization vector $\boldsymbol{b}_i$ from N-vector reproduction alphabet. The mapping $\boldsymbol{Q}$ is to be selected to minimize the mean squared error per dimension [12].

$$\sigma_q^2 = \frac{1}{v} E\left(\left\| \mathbf{X} - \mathbf{Q}(\mathbf{X}) \right\|^2\right) \qquad (11)$$

where E is statistical mean value.

Necessary conditions for an optimum VQ (OVQ) with full search of its reproduction vectors may be easily derived. In summary, associated with each quantizer output vector $\boldsymbol{b}_i$ is a nearest neighbour partition cell A_i, i = 1, ..., N, satisfying

$$A_i = \left\{ \boldsymbol{X} : \left\| \boldsymbol{X} - \boldsymbol{b}_i \right\|^2 < \left\| \boldsymbol{X} - \boldsymbol{b}_j \right\|^2,\ \ \forall\ j \ne i \right\} \qquad (12)$$

For each partition cell A_i, $\boldsymbol{b}_i$ is the conditional mean

$$\boldsymbol{b}_i = E\left(\mathbf{X} / \mathbf{X} \in A_i\right) \qquad (13)$$

The necessary conditions (12) and (13) provide the bases of the iterative LBG vector quantizer design algorithm. Since analytical or numerical computation of (13) is not feasible, a random number generator is used to produce a training sequence ($\boldsymbol{x}_n$, n = 1, ..., T), representing the distribution of random vector $\boldsymbol{X}$ (or, for a physical source of interest such as image, an appropriate training sequence is selected from available data). Given an initial set of output vectors *(**b**$_i$, i* = 1, ..., N*)*, (12) can be applied to the training sequence and the various $\boldsymbol{x}_n$ assigned to cells A_i of partition. Equation (13) can then be solved for a new set of quantizer output vectors. These output vectors are in general different from the initial values and can be used again in (12) with the training sequence to reassign each $\boldsymbol{x}_n$ to a new partition. Iterating between (12) and (13) provides a nonincreasing distortion, and the algorithm eventually converges to a locally optimum design. The basic idea of the LBG algorithm [11] is quite simple and can be summarized for an unknown distribution training sequence as follow:

1. Let *N* is number of quantizer output vectors, distortion threshold $\varepsilon \geq 0$, assume an initial *N*-vector reproduction alphabet B_0 and a training sequence $\boldsymbol{x}_n$, *n*=1, ..., T, and *r* is the number of iterations set to zero.
2. Given B_r = *(**b**$_i$, i* = 1, ..., N*)*, find the minimum distortion partition $P(B_r) = (A_i$, *i* = 1, ..., N*)* of the training sequence : $\boldsymbol{x}_n \in A_i$ *if* $d(\boldsymbol{x}_n , \boldsymbol{b}_i) \leq d(\boldsymbol{x}_n , \boldsymbol{b}_j)$, for all *j*, where *d* denotes a distance. Compute the time average distortion

$$\sigma_{qr}^2 = \sigma_q^2\left[B_r, P(B_r)\right] = \frac{1}{T}\sum_{n=1}^{T} \min_{b_i \in B_r} d(\boldsymbol{x}_n, \boldsymbol{b}_i) \qquad (14)$$

3. If $\left(\sigma_{qr-1}^2 - \sigma_{qr}^2\right)/\sigma_{qr}^2 \leq \varepsilon$, stop the iteration with the B_r final reproduction alphabet, otherwise continue.
4. Find the optimal reproduction alphabet

   B[P(B$_r$)] = *[* ***b***$_i$*(A$_i$)*, *i* = 1, ..., N*]* for *P(B$_r$)* where

$$\boldsymbol{b}_i(A_i) = \frac{1}{\|A_i\|}\sum_{n:\in A_i} \boldsymbol{x}_n \qquad (15)$$

5. Set $B_{r+1} = B[P(B_r)]$ and increment *r* to *r* + 1 and go to item 2.

In the above iterative algorithm, an initial reproduction alphabet B_0 was assumed in order to start the algorithm. There are a number of techniques to obtain the initial codebook. The simplest technique is to use the first widely spaced words from the training sequence. Another technique uses a splitting method where the centroid for the training sequence was calculated and split into two close vectors. These centroids or the reproduction vectors for the two partitions were then calculated. Each resulting vector was then split into two vectors and the above procedure was repeated until an N-vector initial reproduction alphabet was created. Splitting was performed by adding a fixed perturbation vector ***k*** to each vector ***b***$_i$ producing two new vectors ***b***$_i$+ ***k***, ***b***$_i$ – ***k***.

4. TRELLIS CODING FOR VECTOR QUANTIZATION

The basic idea behind trellis coding for vector quantization (TCVQ) is the following: Assume that we want to vector quantize a stationary vector source at the rate of n bits per vector. Performing quantization directly on this vector source would require OVQ with full search of 2^{Kn} reproduction vectors (codevectors), where *K* is a dimension. The OVQ may represent any source vector with any of available codevectors. The idea behind TCVQ [13] is to first quantize the source more finely with 2^{Kn+k} codevectors. Of course this would exceed the allocated rate, so we cannot have a free choice of symbols at all times. In our case we take *k* = 1 and the TCVQ has 2^{Kn+1} codevectors (i.e. twice that of the OVQ), but only a subset of size 2^{Kn} of these codevectors may be used to represent a source vector at any instance of time.

The vector codebook of 2^{Kn+1} codevectors is partitioned into subset of 2^{Kn-1} codevectors, each generating four sets. For example *n* = 2 bits, see Fig. 3. The subsets are designed such that each of them represents reproduction vectors of a coarser bit rate *(n*-1*)* quantizer. The four subsets are designated D_0, D_1, D_2 and D_3. Also, define $S_0 = D_0 \cup D_2$ and $S_1 = D_1 \cup D_3$, where S_0 and S_1 are known as supersets [14], [15].

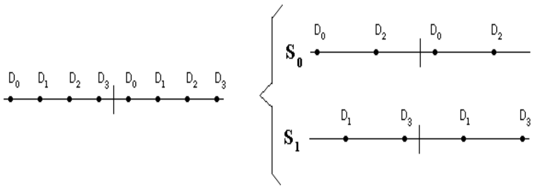


Fig. 3 Codebook of TCVQ and its supersets

Obviously, the rate constraint prohibits the specification of an arbitrary codevector out of 2^{Kn+1} codevectors. However, it is possible to exactly specify, with *n* bits, one element out of either S_0 or S_1. At each input vector, assuming we know which one of the superset to use, one bit can be used to determine the active subset, and *n*-1 bits to specify a codevector from the subset. The choice of superset is determined by the state of finite state machine, described by a suitable trellis. An example of such a trellis, with eight states, is given in Fig. 4. The subsets *{D$_0$, D$_1$, D$_2$, D$_3$}* are also used to label the branches of the trellis, so the same bit that specifies the subset (at a given state) also determines the next state of the trellis.

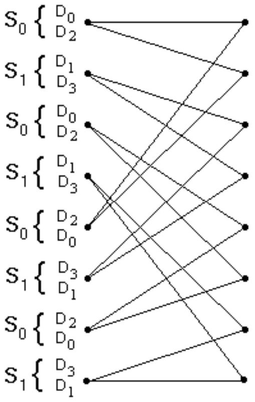


Fig. 4 Eight state trellis

Encoding is achieved by spending one bit per sample on specifying the path trough the trellis, while the remaining n-1 bits specify a codevector out of the active subset. We have more codevectors than a non optimal bitrate - n quantizer, because there is some freedom of choosing from codevectors of either S_0 or S_1. Of course this choice is not completely free: the decision made at each input vector is linked to decisions made at past and future input vectors, through the permissible paths of the trellis. But it is this additional flexibility that leads to the improved performance. The function of the TCVQ encoder is to find the best possible reproduction for any given sequence of source vectors. Specifically, for a search depth of L, the encoder looks for the sequence of codevectors on the L-state trellis that is at minimum Euclidean distance from the given sequence of L source vectors. Each sequence of codevectors on the trellis is specified by a path (a connected sequence of L branches) and the choice of codevectors from the subcodebooks associated with the branches. For a given sequence of source vectors, the encoder uses the well-known Viterbi algorithm to find the best sequence of codevectors. Then it releases all the LnK bits specifying this sequence at the end of the L-state trellis, which the decoder uses to generate the reproduction. The process is then repeated for the next L source vectors.

For design algorithm of TCVQ, the main goal of the partitioning step is to provide subsets of codevectors with maximal minimum distance within each subset. The overall set of codevectors is partitioned in several stages such that a binary tree of subsets of codevectors is generated. The two descendent nodes of any internal node of the tree form a partition of the subset at that node. The subsets at the leaves are assigned to the trellis branches according to the "mapping by set partitioning" method [18]. Given an initial OVQ codebook C of size $N=2^{Kn+1}$, the distances between all possible pairs of codevectors are calculated and listed in a nondecreasing order along with the corresponding pairs. This gives a table with N x $(N-1)/2$ entries, where the i-th entry corresponds to codevectors $\boldsymbol{b}_i$ and $\boldsymbol{b'}_i$ that are at distance $d_i = \|\boldsymbol{b}_i - \boldsymbol{b'}_i\|_2$. First $\boldsymbol{b}_0$ and $\boldsymbol{b'}_0$ are placed in the subsets S_0 and S_1, respectively, and the first entry is removed from the table. Then the following basic step is repeated as many times as necessary until the size of one of the subsets, S_0 or S_1, reaches $N/2$, at which point the remaining unassigned codevectors (if any) are added to the other subset. The basic step involves looking for the entry with the smallest index i with one (but not both) of the two codevectors already assigned to either S_0 or S_1, followed by adding the unassigned code vector to the other subset. All entries with both codevectors assigned are removed from the table. To partition the overall OVQ codebook into subsets, the algorithm is first applied to overall codebook to partition it into two subsets, followed by applying it to these two subsets to generate four subsets and so on. The performance of TCVQ can be improved if its decoder is modified. The design algorithm for TCVQ coder is locally optimal.

5. EXPERIMENTAL RESULTS

The subband vector quantizer with trellis coding (TCSVQ) was simulated on a personal computer for the input sequence of 320 000 source samples, from which we could make input vectors of dimensions K = 8 or 16. The input training sequence of samples was generated by Gaussian Markovian random discrete process with zero-mean value, unit-variance and correlation coefficient $\rho = 0{,}95$.

The subband system analysis of TCSVQ has been proposed by the K-channel cosine modulated filter bank. Then the procedure for design of its prototype filter (length N = $2mK$) involves the initialization of the parameters of the $\lfloor K/2 \rfloor$ lattices and the optimization of these parameters [17]. Initialization of the parameters is carried out as follows

$$\theta_{k,p} = \begin{cases} \dfrac{\pi}{4}\,, \quad p = 0\,, & 0 \le k \le \left\lfloor \dfrac{K}{2} \right\rfloor - 1 \\ \dfrac{\pi}{2}\,, \quad 1 \le p \le (m-1), & 0 \le k \le \left\lfloor \dfrac{K}{2} \right\rfloor - 1 \end{cases} \tag{16}$$

Having initialized all $m\lfloor K/2 \rfloor$ parameters, they are then optimized using function "fmins" [19] to minimize the objective function

$$\phi = \int_{(\pi/2K)+\delta}^{\pi} \left| H\left(e^{j\omega}\right)\right|^2 d\omega \tag{17}$$

where δ < π/2K. Using ϕ, the problem involves the minimization of the stopband energy yielding a minimum energy solution.

After optimization according to eq. (16) and eq. (17), we have obtained the desired prototypes filters for 8 and 16 channels. The impulse responses coefficients of the prototype filters $H(z)$ for K = 8, m = 2, δ = 0.001, with N = 32 and K = 16, m = 2, δ = 0.001, with N = 64 are given in the Tab. 1 and Tab. 2. The impulse responses coefficients for N = 32 *and* N = 64 are symmetrical, therefore only one-half of coefficients are in these tables.

Table 1 Impulse response coefficients of prototype for N = 32 of a 8-channel modulated PR filter bank

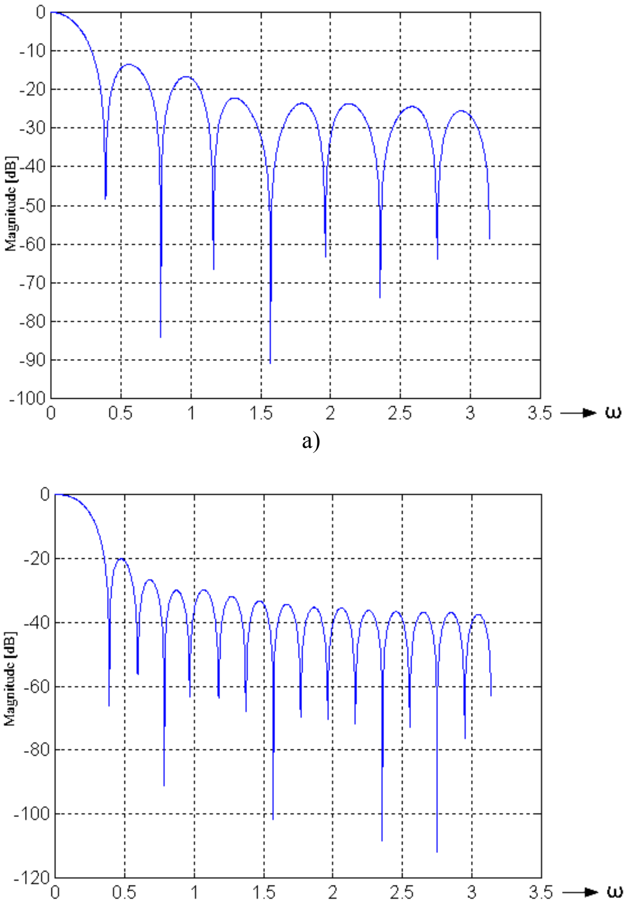
n	0	1	2	3
h(n)	-0.149026	-0.126144	-0.085469	-0.031739
n	4	5	6	7
h(n)	0.034676	0.112369	0.204445	0.305997
n	8	9	10	11
h(n)	0.411710	0.509722	0.599325	0.674428
n	12	13	14	15
h(n)	0.736843	0.787958	0.826120	0.845371

For each of the two designs, the magnitude responses of the prototype are plotted in Fig. 5a and Fig. 5b.

As can be seen from the designed filters (Tab. 1, Tab. 2), length of the impulse responses of the prototype filter increases with channels number K. In comparison, the magnitude response of the prototype for N = 64, K = 16 has better stopband attenuation then this one for N = 32, K = 8 as shown in Fig. 5a and 5b.

Table 2 Impulse response coefficients of prototype for N = 64 of a 16-channel modulated PR filter bank

n	0	1	2	3
h(n)	-0.000016	0.000000	-0.000017	0.000001
n	4	5	6	7
h(n)	-0.000001	0.000017	0.000000	0.000016
n	8	9	10	11
h(n)	0.000484	-0.000011	-0.000248	0.000307
N	12	13	14	15
h(n)	0.000309	-0.000253	-0.000011	0.000495
n	16	17	18	19
h(n)	-0.045275	-0.011039	0.029469	0.015640
n	20	21	22	23
h(n)	-0.015538	-0.028713	0.011266	0.047331
n	24	25	26	27
h(n)	0.682077	0.700498	0.709574	0.723018
n	28	29	30	31
h(n)	0.690477	0.703429	0.713480	0.728341



a)

b)

Fig. 5 Magnitude responses of the prototype a) N = 32 , K = 8, b) N = 64, K = 16

Vector quantization in subband domain was carried out by using trellis coding. Its initial codebooks have been obtained by the full search optimization and consequently were partitioned according trellis algorithm. The depth of trellis was set down on L=10 and number of trellis state S = 4. The simulation results of TCSVQ, given by the signal to noise ratio (SNR) in decibels [dB] are presented in Tab. 3 and Tab. 4 for different K and n.

It is evident from these tables, that at a given dimension K the SNR of TCSVQ increases with the bit rate n. Also, these results proof gradually increasing of SNR along with dimension of vector space at the same bit rate. For K = 16 and n = 0,25 bit/sample SNR of TCSVQ is about 0,53dB better, for n = 0,38 bit/sample about 0,72dB better and for n = 0,5 bit/sample about 1,29dB better then TCSVQ for K = 8.

Table 3 The simulation results of TCSVQ for dimension equal 8 and different bit rate n

K = 8		SNR [dB]							
n' bit/vector		1	2	3	4	5	6	7	8
n bit/sample		0,12	0,25	0,38	0,5	0,62	0,75	0,88	1
	TCSVQ	X	6,23	8,52	10,1	11,4	12,5	13,5	14,6
	PVQ	2,85	5,16	7,61	9,31	10,5	11,6	12,7	13,6

Table 4 The simulation results of TCSVQ for dimension equal 16 and different bit rate n

K = 16		SNR [dB]							
n' bit/vector		1	2	3	4	5	6	7	8
n bit/sample		0,06	0,12	0,19	0,25	0,31	0,38	0,44	0,5
	TCSVQ	X	5,07	5,49	6,76	8,18	9,24	10,51	11,39
	PVQ	1,29	3,47	4,57	5,81	7,26	8,34	9,19	10,18

The obtained results were compared, too with those ones of primary vector quantization (PVQ) directly in signal domain. TCSVQ has better results than PVQ in all cases, approximately about 1 dB. When compared only results of PVQ for different v and the same bit rate, we can see that for K = 16 and n = 0,25 bit/sample SNR of PVQ is about 0,65dB better, for n = 0,38 bit/sample about 0,73dB better and for n = 0,5 bit/sample about 0,87dB better then PVQ for K = 8. That means, these results of SNR do not increase gradually as for TCSVQ in dependence on dimension at the same bit rate.

6. CONCLUSION

In this paper, the subband coding system consisting of the combination analysis filter bank, vector quantizator and trellis coding is presented. Foremost were to be designed by means of the optimization on a personal computer impulse responses of the prototype filters with transfer function $H(z)$ for K = 8, m = 2, δ = 0.001, with N = 32 and K = 16, m = 2, δ = 0.001, with N = 64. The optimization was done directly on the lattice parameters. This enables us to optimize the prototype response, while it is guaranteed that the modulated filter bank will satisfy the PR property. Further the hierarchical structure of the lattice, which makes it flexible to increase the length of the prototype filter, was exploited in the design of procedure. By means of those were next calculated the impulse responses $h_k(n)$ of cosine-modulated filter banks for 8 a 16 channels.

Consequently has been carried out the simulation of the subband vector quatization with trellis coding for the input consequence of 320 000 samples of Gaussian Markovian random discrete process with zero-mean value, unit-variance and correlation coefficient $\rho = 0,95$. The results of this simulation have been evaluated by means of the SNR for different K and n. From the experimental results be evident that SNR of TCSVQ increases both with a bit rate and with a dimension of vector space K. The obtained results were compared, too with those ones of primary vector quantization PVQ directly in signal domain. TCSVQ achieved better results than PVQ in all dimensions as well bit rates approximately about 1 dB. It may be applied as a good coding method for purpose of increasing its performance and so we get the efficient coding system which may be important part of digital processing and communication systems.

REFERENCES

[1] FLIEGE, N. J.: *Multirate Digital Signal Processing: Multirate Systems, Filter Banks, Wavelets*, John Willy & Sons, Chichester, 1994.

[2] MIHALÍK, J. – ZAVACKÝ, J. – DZIVÝ, J.: *Perfect Reconstruction 2DQMF Bank for Subband Image Coding*, Journal of Electrical Engineering, vol. 47, no. 7–8, pp. 195–201, 1996.

[3] JEE, I. – HADDAD, R. A.: *Modelling and Analysis of Vector-Quantized M-Channel Subband Codecs*, International Conference on Acoustics, Speech, and Signal Processing, vol. 2, pp. 1320–1323, May, 1995.

[4] ZAVACKÝ, J. – MIHALÍK, J. – GLADIŠOVÁ, I.: *Implementation of Discrete Wavelet Transform in Standard JPEG-2000*, Electronic Horizon, vol. 63, no. 3–4, pp. 5–9, 2007 .

[5] AKANSU, A, N. – HADDAD, R. A.: *Multi resolution Signal Decomposition (Transforms, Subbands and Wavelets)*, Academic Press, USA, 2nd ed., 2001.

[6] VAIDYANATHAN, P. P.: *Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial*, Proceedings of IEEE 78, no. 1, pp. 56–93, 1990.

[7] ZAVACKÝ, J. – MIHALÍK, J.: *Algorithm for Calculation of Wavelets by Using Quadrature Mirror Filter Bank*, Acta Electrotechnica et Informatica, vol. 5, no. 1, pp. 42–49, 2005.

[8] VAZE, R. – RAJAN, B. S.: *On Space–Time Trellis Codes Achieving Optimal Diversity Multiplexing Tradeoff*, IEEE Trans. Inf. Theory, vol. 52, no. 11, pp. 5060–5067, Nov. 2006.

[9] BARO, S. – BAUCH, G. – HANSMANN, A.: *Improved Codes for Space-Time Trellis-Coded Modulation*, IEEE Commun. Lett., vol. 4, pp. 20–22, Jan. 2000.

[10] GLADIŠOVÁ, I. – MIHALÍK, J. – ZAVACKÝ, J.: *Lossless Compression of Image by Using State Binary Arithmetic Coding its Bit Planes*, Electronic Horizon, vol. 62, no. 1–2, pp. 18–21, 2006.

[11] GERSHO, A. – GRAY, R. M.: *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992.

[12] MIHALÍK, J.– GLADIŠOVÁ, I.– MICHALČIN, V.: *Two Layer Vector Quantization of Images*, Radioengineering, vol. 10, no. 2, pp. 14–19, 2001.

[13] UNGERBOECK, G.: *Trellis – Coded Modulation with Redundant Signal Sets – Part II: State of the Art*, IEEE Commun. Mag., vol. 25, pp. 12–21, 1987.

[14] NARAYANAN, T. A. – RAJAN, B. S.: *A General Construction of Space-Time Trellis Codes for PSK Signal Sets*, In Proc. IEEE GLOBECOM, pp. 1978–1983, 2003.

[15] WANG, H. S. – MOAVERI, N.: *Trellis Coded Vector Quantization*, IEEE Trans. on Commun., vol. 40, no. 8, pp. 1273–1276, 1992.

[16] ZAVACKÝ, J. – MIHALÍK, J.: *An Algorithm for Design of Cosine Modulated Filter Banks with Perfect Reconstruction*, Journal of Electrical Engineering, vol. 50, no. 5–6, pp. 122–126, 1999.

[17] KOILPILLAI, R. D. – VAIDYANATHA, P. P.: *Cosine-Modulated FIR Filter Banks Satisfying Perfect Reconstruction*, IEEE Trans. on Signal Processing , vol. 40, no. 4, pp. 770–783, 1992.

[18] GLADIŠOVÁ, I. – MIHALÍK, J.: *Trellis Coding for Vector Quantization*, Proc. 5th Intl. Scien. Conf. „Digital Signal Processing and Multimedia Communications“, Košice, pp. 31–33, Nov. 2001.

[19] *Matlab On–Line Documentation, Optimalization Toolbox*. Mathworks, Natick, MA, 2002.

Received July 8, 2010, accepted January 17, 2011

BIOGRAPHY

Viktor Michalčin was born in Ukraine. He obtained MSc. Degree in Electronics and Multimedia Telecommunication at Technical University of Košice in 2000. Since 2000 he joined with same university of Košice, where he received his PhD Degree in 2005. Today he is a member of Laboratory of Transfer Technology at University of Pavol Jozef Šafárik University at Košice and works as a developer of video tools for videoconference system EVO. His research interests are focused on vector quantization of signals and video compression algorithms.