Trellis Coding For Vector Quantization

Ing.Iveta GLADIŠOVÁ, PhD.- Prof.Ing. Ján MIHALÍK, PhD.*
FEI TU Košice, Park Komenského13, 041 20 Košice, SR
Tel.: +0421-55-6022940; +0421-55-6022854

Abstract - A new source coding technique based on the idea of trellis coded modulation is trellis coded quantization. We present a trellis coded quantization and its vector generalization, then experimental results showing its performance on the identically independent distribution Gaussian and Laplacian sources. Trellis coding for vector quantization (TCVQ) yields a lower distortion then optimum vector quantization (OVQ) for a given bit rate at the cost of an increase in implementation complexity.

Index terms – trellis, vector quantization

1 Introduction

Trellis coded quantization (TCQ) [1], [4] is a fast effective method of quantizing random variables. Trellis coding exploits correlations between variables and it can use non-rectangular quantizer cells that give it quantization efficiencies not attainable by scalar quantizers. The central ideas of TCQ grew out of the ground – breaking work of Ungerboek [5] in trellis coded modulation. TCQ has proved to be an attractive source coding technique with good rate-distortion performance and yet low implementation complexity. While TCQ’s performance is in many case close to the theoretical rate-distortion bound, an improvement is always possible by generalizing its structure to the vector case. This comes at the cost of an increase in complexity due to vector codebook searching.

TCQ of a given rate assigns the output levels [7] of a scalar quantizer with twice that rate to the branches of a trellis according to the “mapping by set partitioning” method of trellis coded modulation. This is done in such a way that the minimum Euclidean distance among the output levels assigned to the same branch is maximized. While this is rather straightforward in the scalar case, it is not a simple task in the vector case and multidimensional space. An algorithm of TCVQ [6] is described in another part of the paper which partitions an optimum vector quantizer codebook into a given number of subsets such that the desired property of maximizing the minimum intrasubset Euclidean distances may be realized here as well. The codebook used in TCVQ plays an important role in the coder’s performance.

We present experimental results showing the performance of TCVQ’s of different bit rate and dimensions on the identically independent distribution Gaussian and Laplacian sources. Comparisons are made with OVQ [2].

2 Trellis coding for vector quantization

The basic idea behind TCVQ is the following: Assume that we want to vector quantize a stationary, memoryless uniform source at the rate of n bits per vector. Performing quantization directly on this uniform source would require an OVQ with $2^{vn}$ reproduction vectors (codevectors), where $v$ is a dimension. The OVQ may represent any source vector with any of available codevectors. The idea behind TCVQ is to first quantize the source more finely, with $2^{vn+k}$ 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 the TCVQ has \( 2^{vn+1} \) codevectors (i.e. twice that of the OVQ), but only a subset of size \( 2^{vn} \) of these codevectors may be used to represent a source vector at any instance of time.

The vector codebook of \( 2^{vn+1} \) codevectors is partitioned into subset of \( 2^{vn-1} \) codevectors, each generating four sets. For example \( n=2 \) bits, see Fig.1. 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.

![Fig.1 Codebook of TCVQ and its supersets](image)

Fig. 1 Codebook of TCVQ and its supersets

Obviously, the rate constraint prohibits the specification of an arbitrary codevector out of \( 2^{vn+1} \) 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.2. 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.2. Eight state trellis.](image)

Fig. 2. Eight state trellis.

Encoding is achieved by spending one bit per sample on specifying the path through 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 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 [6].

Given an initial OVQ codebook $C$ of size $N=2^{vn+1}$, 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 $N(N-1)/2$ entries, where the $i$th entry corresponds to codevectors $b_i$ and $b'_i$ that are at distance $d_i = \|b_i - b'_i\|$. First $b_0$ and $b'_0$ 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 codevector 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.

4 Experimental results and conclusions

The designed algorithm of TCVQ was simulated on a personal computer for the sequence of 360,000 source samples, from which we could make input vectors of dimension $v = 2, 4, 8, 12, 16$, while the following may be selected: bit rate $n' = 1 \div 8$ bit/vector or $n$ bit/sample, number of trellis state $S = 4, 8, 16, 32, 64$ and different search depth $L$. The input training sequence of samples was generated by identically independent distributed Gaussian (Gpdf) and Laplacian (Lpdf) random variables with zero-mean value and unit-variance. The initial VQ codebooks was used codebooks of OVQ, which were partitioned according this algorithm. The algorithm performance is characterized by the signal noise ratio (SNR) in decibe [dB].

The simulation results of TCVQ are presented in Tab.1 for different $v$ and $n$. The obtained results were compared with those of OVQ. It is evident from the tables that at a given dimension $v$ the SNR of TCVQ increases with the bit rate $n$. Also these results proof increase of SNR along with dimension of space at the same bit rate. In Tab.2 are presented SNR results for different parameters $S$ and $L$ for one of selected dimension and probability density function. From this table we can see, that ANR increase with number of trellis state $S$ and also with search depth $L$, but then it changes only slightly. It is caused by limitation of this search depth.

This paper deals with the design of an algorithm of trellis coding for vector quantization. The main advantage of this method is that few computational costs ensue compared with classical vector quantizer methods. The principle of TCVQ was explained and the obtained results of simulation designed algorithm were finally evaluated. This algorithm uses and provides good quantization results at low bit rate. It may be applied as a good coding method for purpose of increasing its performance and
The work was supported by the Grant Agency of Ministry of Education of Slovak republic under GAV - 1/7387/20

References


