1991年度 大韓醫用生體工學 秋季 學術大會 論文集 91/11

적응 변환코드를 이용한 영상신호 압축

○김 동 윤, 이 경 중, 윤 형 로
연세대학교 보건과학대학 의용공학과

An Adaptive Transform Code for Images

○Dong-Youn Kim, Kyung Joung Lee, Hyung Ro Yoon
Dept. of Medical Engineering, Yonsei Univ.

Abstract

There exists a transform trellis code that is optimal for stationary Gaussian sources and the squared-error distortion measure at all rates. In this paper, we train an asymptotically optimal version of such a code to obtain one which is matched better to the statistics of real world data. The training algorithm uses the M-algorithm to search the trellis codebook and the LBG-algorithm to update the trellis codebook. To adapt the codebook for the varying input data, we use two gain-adaptive methods. The gain-adaptive scheme 1, which normalizes input block data by its gain factor, is applied to images at rate 0.5 bits/pixel. When each block is encoded at the same rate, the nonstationarity among the block variances leads to a variation in the resulting distortion from one block to another. To alleviate the non-uniformity among the encoded image, we design four clusters from the block power, in which each cluster has its own trellis codebook and different rates. The rate of each cluster is assigned through requiring a constant distortion per-letter. This gain-adaptive scheme 2 produces good visual and measurable quality at low rates.

1. Introduction

Mazor and Pearlman[1] proved the existence of a transform trellis code whose mean squared-error performance approaches that of the rate-distortion bound exponentially fast with increasing constraint length. The corresponding coding theorem is valid for any discrete-time stationary Gaussian source with bounded continuous power spectrum and all coding rates. This coding scheme involves the concept of random coding and is optimal when the block size goes to infinity and will produce a good code with high probability for a sufficiently large block size. This code has been used with impressive results for encoding speech[2] and images[3].

To implement this coding scheme, we have to use a finite shift register with constraint length K and a block size N greater than K. The reproduction symbols on the trellis branch are drawn at random from the rate-distortion-theoretic optimal test channel probability distribution. Optimality of the codebook is guaranteed for infinite K, but for finite K, there is a high probability that the codewords selected in this way will not be representative of the distribution from which they are drawn. Moreover, for real-world sources which are inherently nonstationary, a random codebook selected through a stationary model may not be well matched. The transform trellis code used adaptations of random codebooks to overcome these difficulties[2,3,5]. These considerations provide motivation for studying methods for training of a transform trellis code with the actual data. In this paper, first we shall describe the training algorithm of the trained transform trellis code. Then each adaptive scheme will be described and applied to images at overall rates of 0.56 and 0.58 bits/pixel respectively. These results will be compared with those obtained by others at comparable rates.

2. The Transform Trellis Code

2.1. Transform Trellis Code

Let $Z_{i,j}$ be an N×N image, $i,j = 0,1,2,\ldots,N-1$. We segmemt $Z_{i,j}$ into a p×p subblocks and denote the mth subblock by $Z^m_{k,l}$, $m=1,2,3,\ldots,(N/p)^2$ and $k,l = 0, 1, 2, \ldots, p-1$. Let $\hat{Z}^m_{k,l}$ be a zero-mean subblock which is generated by subtracting its block mean μ_m from $Z^m_{k,l}$. This block mean will be reinserted after the inverse two dimensional Karhunen Loeve Transform(2D-KLT). For these zero-mean subblocks, the 2D-KLT will be applied to produce the uncorrelated transformed coefficients $U^m_{k,l}$ with variance $\lambda^m_{k,l}$. If we assume the distribution of each subblock is Gaussian, then the rate-distortion function for a Gaussian subblock with mean squared-error distortion measure is given parametrically[4]

$$D^m_\theta = \frac{1}{p^2}\sum_{k=0}^{p-1}\sum_{l=0}^{p-1}\min\{\theta_m, \lambda^m_{k,l}\} \quad (1)$$

$$R_m(D^m_\theta) = \frac{1}{p^2}\sum_{k=0}^{p-1}\sum_{l=0}^{p-1}\max\left\{0, \frac{1}{2}\log_2\frac{\lambda^m_{k,l}}{\theta_m}\right\} \quad (2)$$

where $R_m(D^m_\theta)$ is the optimal rate of m th block mean squared-error.

In our simulations, we used the two dimensional Discrete cosine Transform(2D-DCT) instead of 2D-KLT because of its fast algorithm and closeness in performance. The unitary 2D-DCT of $\hat{Z}^m_{k,l}$ is as follows

$$U^m_{u,v} = \sum_{k=0}^{p-1}\sum_{l=0}^{p-1} \hat{Z}^m_{k,l} T_u(k) T_v(l) \quad (3)$$

$$\hat{Z}^m_{k,l} = \sum_{u=0}^{p-1}\sum_{v=0}^{p-1} U^m_{u,v} T_k(u) T_l(v) \quad (4)$$

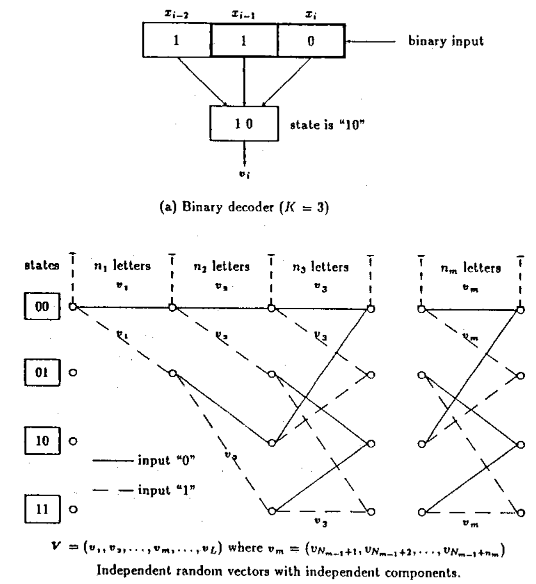
$$T_u(k) = \sqrt{2/p}\,\alpha(u)\cos\left\{\frac{\pi u(2k+1)}{2p}\right\} \quad (5)$$

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{2}} & ,u = 0 \\ 1 & ,u \neq 0 \end{cases}$$

2.2. The Structure of the Transform Trellis Code

The trellis structure is completely determined by the number of levels L, the branching factor q, and the shift register constraint length K, for a given desired rate R and the dimension of the transformed sequence N. The number of nodes equals to q^{K-1}, the number of states of the first K-1 stages of the register. The number of branches emanating from each node, or branching factor, is q, the number of different possibilities when the next transmission symbol enters the register. Fig.1 shows a trellis with branching factor q=2 and shift register length K=3. At level l of the trellis, each branch is populated with a vector v^m_l of n_l reproduction letters. These vectors and their lengths are obtained from Eq.(1) and (2). The detailed rate assignment procedure is in Ref.[5,8]. The optimal distribution function of v^m_l is[4]

$$P_{n_l}(v^m_l) = \prod_{a\in J_{N_m}} \frac{1}{\sqrt{2\pi(\lambda^m_a - \theta_m)}} \exp\left\{-\frac{(v^m_a)^2}{2(\lambda^m_a - \theta_m)}\right\} \quad (6)$$



(b) Trellis Diagram ($q = 2$, $K = 3$)

Fig. 1: Structure of a trellis code

2.3. Training algorithm

The initial codebook is constructed from the average spectrum of a training sequence and the reproduction letters are generated randomly from the optimal distribution function in Eq.(6) where the average training sequence spectrum approximates the eigenvalue spectrum. To train the trellis code, we use a search algorithm called the M-algorithm[6] to find partitions of codeword and the LBG-algorithm[7] to calculate the centroid of partitions. The trellis training algorithm consists of three parts: For a given training sequence and an initial codebook which is constructed from the average spectrum of the training sequence, first, find the minimum distortion path with M-algorithm. And update the total distortion. Second, if the decrease of the distortion ratio is small enough then quit. Otherwise go to next step. Third, from the path-maps of the training sequence, update each branch reproduction letters by replacing the centroid of the training sequence subvectors which have been encoded to that reproduction letters. Go to step 1.

The following is the flow chart of the training algorithm for a transform trellis code. To describe the training algorithm, we use the following parameters:

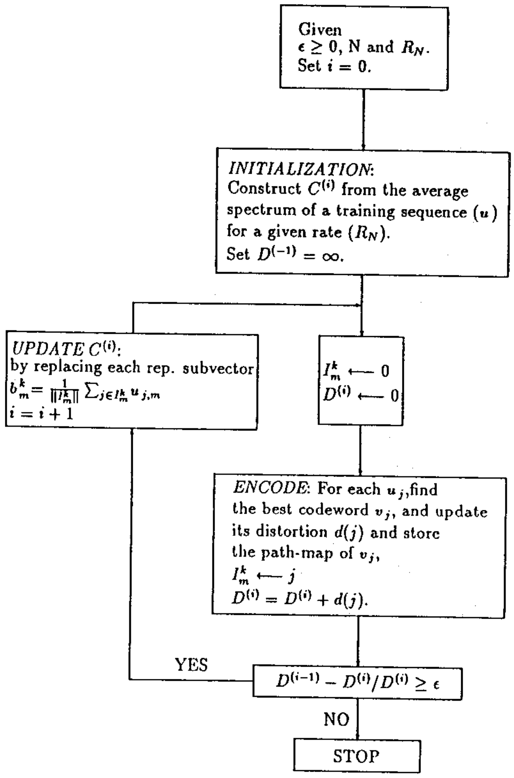


Fig.2: Flow chart of the training algorithm.

m Index of the trellis level

k Index of the trellis branch

j Index of block

$D^{(i)}$ Squared error distortion per block after ith iteration

$C^{(i)}$ Trellis codebook of after ith iteration

b_m^k Reproduction vector on mth level and kth branch

u_j A training vector of jth block

$u_{j,m}$ A subvector of u_j at mth level

v_j The reproduced vector of u_j

$v_{j,m}$ A subvector of v_j at mth level

I_m^k The set of indices of which have been assigned b_m^k

$\|I_m^k\|$ The cardinality of I_m^k

3. Simulations

We used seventeen images as a training sequence. Two 512×512 images (LENA, PEPPERS), which are not included in a training sequence, and one 480×512 (GIRL) from *inside training sequence are used as a test sequence.* As a performance measure, we used the peak-to-peak SNR(PSNR) which is defined by

$$PSNR = 10\log_{10}(P^2/D)$$

$$D_m = \frac{1}{p^2}\sum_{k=0}^{p-1}\sum_{l=0}^{p-1}(Z_{k,l}^m - Y_{k,l}^m)^2$$

$$D = \left(\frac{p}{N}\right)^2 \sum_{m=1}^{(\frac{N}{p})^2} D_m$$

where P is 255 for an 8-bit gray level(0,1,...,255) image. The mth original and reproduced subblock is denoted by $Z^m_{k,l}$ and $Y^m_{k,l}$, respectively. For all our simulations, we designed trellis codebook at the value of K=3 and q=8 searched at the value of M=16 and the block size p=16.

3.1. Gain-Adaptive Scheme 1

To cope with the widely varying dynamic range of input data, we normalized transformed coefficients by their block's gain factor. For this gain-adaptive trained transform trellis code, we designed the trellis codebook at rate 0.5 bits/pixel. Fig. 3 describes the block diagram of the gain-adaptive scheme 1. The transformed coefficients, $U^m_{k,l}$, are linearly arranged *as in the linear order of the normalized average* spectrum of the training sequence which has been used to construct the initial trellis codebook. The block gain factor, G_m, is defined as

$$G_m = \sqrt{\sum_{k=0}^{15}\sum_{l=0}^{15}\{\hat{Z}_{k,l}^m\}^2}$$

For the overall rate is 0.56 bits/pixel (0.5 bits/pixel for the codebook rates and 16 bits for block mean and gain factor), the achieved PSNR of GIRL from the inside training sequence, is 37.1 dB, and the achieved PSNR's of LENA and PEPPERS from the outside training sequence, are 33.4 dB and 34.1 dB, respectively. At low rates, when encoded with one codebook at the same rate, some blocks with low variances would benefit from this fixed rate assignment for each block and others with relatively high variances would not. This nonstationarity among the blocks leads to a variation in the resulting distortion from one block to another.

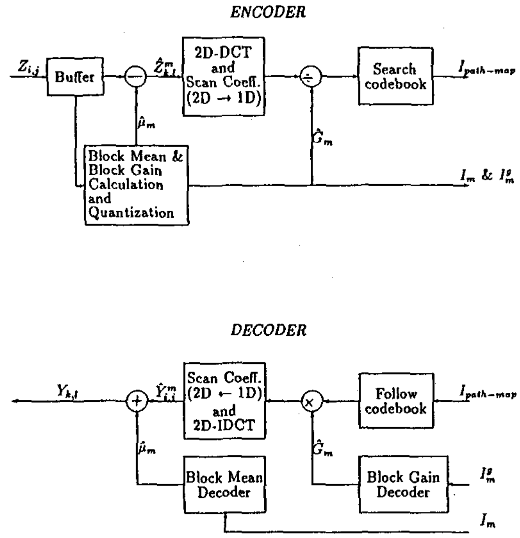


Figure 3: Block diagram of the GAIN-ADAPTIVE scheme 1

3.2. Gain-Adaptive Scheme 2

In gain-adaptive scheme 2, we classified the zero-mean subblocks based upon its block power, which is *considered to represent the activity level of each* block. The decision threshold of each cluster is determined such that each cluster has the same power.

In our simulations, we designed four clusters and the codebook of each cluster is designed for the normalized average spectrum of each cluster as in gain-adaptive scheme 1. Table 1. shows the assigned rate of each cluster and its decision threshold. The difference between the gain-adaptive scheme 1 and 2 is that the latter method uses different rate and structure of the trellis for subblocks of different classes.

Each input subblock is classified through its block power, so the receiver, which has the same classifier as in encoder, can select the cluster from the received gain factor. The designed average rate for this gain-adaptive scheme 2 is 0.45 bits/pixel for the training sequence. Because the side information (8 bits for block mean and 8 bits for gain factor) is 16/256 bits/pixel, the overall rate for the training sequence is 0.51 *bits/pixel. The resulting PSNR of GIRL (inside* training sequence) was 38.2 dB. When these cluster codebooks are applied to LENA and PEPPERS (outside training sequence), the PSNR's achieved were 33.8 dB and 34.3 dB, respectively, at the same overall rate of 0.58 bits/pixel. This rate discrepancy is due to the different block variance distributions of the inside and outside the training sequence. Although the improvements in dB over the previous adaptation method were minimal, the improvement in visual quality was indeed noticeable. The transform trellis code[3], which performed two passes on the input image, achieved 37.4 dB (PSNR) for the lower right quadrant of LENA at rate

0.56 bits/pixel. We compared these results with the performance of other coding schemes which perform one pass on the LENA(512×512) image. The PSNR of this gain-adaptive scheme 2 is 0.3 dB to 2.7 dB better than that of the sliding block entropy coding[10], pruned tree-structured vector quantization[11], and finite state vector quantization[12] at comparable rates.

Cluster (c)	Decision Threshold (dB)	*Prob*(c) %	Rate (bits/pixel)
1	53.00	82.75	0.3
2	56.5	9.95	1.1
3	58.8	4.69	1.125
4	∞	2.61	1.25

Table 1: Specification of each cluster

4. Summary and Conclusions

The trained transform trellis code was proposed and applied to encode images with gain adaptations. As a transform, a 16 x 16 two dimensional discrete cosine transform is used. To avoid the large distortion which occurs from the limited codeword choices and large variances of low coefficients during initial build-up of the trellis, we linearly order the two dimensional average variances of the training sequence in ascending order of magnitude. Then this linearly ordered average variance spectrum is normalized by dividing its power. The initial codebook is constructed from the normalized average spectrum at the value of K=3 and q=8 and searched for M, the minimum number of paths retained at each level, equals to 16. Gain-adaptive scheme 1 used one codebook with same rate for each subblock for normalized input data. When encoded at the same rate for each block, the nonstationarity of the image leads to a blocking effect. To alleviate this blocking effect, in gain-adaptive scheme 2, we constructed four clusters based on the block power. Each subblock is classified by the block power and each cluster has its own trellis codebook and scanning order. The receiver knows the cluster by decoding the received index of the gain factor. So, we need no side information to indicate the index of clusters. This gain-adaptive scheme 2 achieved good visual quality and the PSNR for the LENA(512×512) is 0.3 dB to 2.7 dB better than that of others at comparable rates.

5. References

1. B. Mazor and W. A. Pearlman, "A Trellis Code Construction and Coding Theorem," *IEEE Trans. Inform. Th.*, vol. IT-29, no. 6, pp. 924-930, Nov. 1983.
2. B. Mazor and W. A. Pearlman, "A Transform Trellis with Applications to Speech," *IEEE Trans. Comm.*, vol. COM-33, pp.1109-1116, Oct. 1985.
3. Farkash, D. S. Malah and W. A. Pearlman, "Transform Trellis Coding of Images at low Rates," *IEEE Trans. Comm.*, to be published.
4. T. Berger, Rate Distortion Theory, Englewood Cliffs, NJ:Prentice Hall, 1971.
5. R. Toy and W. A. Pearlman, "Backward Adaptation for Transform Trellis Coding of Speech," *Proceedings of the 1987 ICASSP*, pp.2209-2212, April 1987.
6. J. B. Anderson and J. B. Bodie, "Tree Encoding of Speech," *IEEE Trans. Inform. Th.*, vol. IT-21, no.4, pp. 379-387, July 1975.
7. Y. Linde, A. Buzo and R. M. Gray, "An Algorithm for Vector Quantization," *IEEE Trans. Comm.*, vol. COM-28, no.1, pp.84-95, Jan. 1980.
8. D. Y. Kim and W. A. Pearlman, "A Trained Transform Trellis for Stationary Gaussian Sources," *Proc. of the 1989 Conference On Information Sciences and Systems*, pp. 718-722, Baltimore, 1989.
9. J. H. Chen and A. Gersho, "Gain-Adaptive Vector Quantization with Application to Speech Coding," *IEEE Trans. Comm.*, vol. COM-35, no.9, pp.918-930, Sept.1987.
10. R. A. Cohen and J. W. Woods, "Sliding Block Entropy Coding of Images," *Proc.IEEE Int. Conf. Acoust.,Speech, Signal Processing*, vol.3 pp.1731-1734, May 1989.
11. E. A. Riskin, E. A. Daly and R. M. Gray, "Pruned Tree-structured Vector Quantization in Image Coding," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol.3 pp.1735-1738, May 1989.
12. T. Kim, "New Finite State Vector Quantizers for Images," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol.2 pp.1180-1183, April 1988.