정상 가우시안 소오스와 음성 신호용 변환 격자 코드에 대한 훈련 알고리즘 개발

A Training Algorithm for the Transform Trellis Code with Applications to Stationary Gaussian Sources and Speech

김 동 윤* 박 용 서** 황 금 찬*** William A. Pearlman****

(Dong-Youn Kim,* Yong-Seo Park,** Keum-Chan Whang*** William A. Pearlman****)

요 약

변환 격자 코드는 모든 레이트에서 정상 가우시안 소오스와 자승 오차 왜곡에 대해 최적코드이다. 본 논문은 실제 데이타의 통계적 특성에 잘 맞는 코드를 얻기 위해 점근적으로 최적인 변환 격자 코드를 훈련시켰다. 훈련 알고리즘은 격자 코드북을 탐색하기위한 M알고리즘과 코드북을 새롭게하기 위한 LBG 알고리즘을 사용했다.

훈련된 변환 격자 코드의 성능을 조사하기 위해서 상관 계수가 0.9인 1차 AR 가우시안 소오스와 실제 음성 데이타를 사용하였다. 1차 AR 소오스에서, 훈련에 사용되지 않은 데이타에 대한 SNR은 레이트에 따라 샤논의 정보량 왜곡 함수에 의한 SNR보다 0.6에서 1.4dB 낮았으나, 이것은 같은 계산량을 사용한 다른 코딩 결과들보다 우수 했다. 실제 음성 데이타는 레이트 1.0 bits/sample에서 코딩을 했으며, 보다 좋은 성능을 얻기 위해 윈도우 함수와 이득 적응을 사용했다.

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.

We investigate the trained transform trellis coding scheme for the first-order AR(autoregressive) Gaussian source whose correlation coefficient is 0.9 and actual speech sentences. For the first-order AR source, the achieved SNR for the test sequence is from 0.6 to 1.4 dB less than the maximum achievable SNR as given by Shannon's rate-distortion function for this source, depending on the rate, and surpasses all previous known results for this source. For actual speech data, to achieve improved performance, we use window functions and gain adaptation at rate 1.0 bits/sample.

*Dept. of Medical Engineering, Yonsei University
**Dept. of Electrical Engineering, Kyungwon University
***Dept. of Electrical Engineering, Yonsei University
****Dept. of Electrical Computer and System Engineering, Rensselaer Polytechnic Institute, U.S.A. .
접수일자 : 1991. 10. 10.

1. Introduction

The existence of a transform trellis code whos

mean squared-error performance approaches that of the rate-distortion bound exponentially fast with increasing constraint length was proved by Mazor and Pearlman. The corresponding coding theorem is valid for any discrete-time stationary Gaussian source with bounded continuous power spectrum and all coding rates [1-2]. This coding scheme involves the concept of random coding and is optimal when block size N 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. The trellis branch symbols are drawn at random from the rate-distortion theoretic optimal test channel probability distribution. Optimality of the codebook is guaranteed for infinite N, but for finite N, there is a finite 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 can not be well matched. To overcome these difficulties, the transform trellis code used adaptations of a random codebook. These adaptations are concentrated on the estimation of the spectrum which plays an important role in designing the trellis codebook. This spectral estimation has to be done every few blocks or clusters which has similar spectral characteristics and the corresponding codebook is generated from this estimated spectrum [2,3]. In our work, we trained an asymptotically optimal version of transform trellis code to obtain one which is matched better to statistics of real world data. After a set of training data is used to design the codebook, the codebook remains fixed.

In this paper, we present the first training algorithm for a transform trellis codebook which uses the M algorithm[4] to search a trellis and the LBG algorithm[5] to update the trellis codebook. The remainder of this paper is organized as follows. In Chapter 2, we introduce the transform trellis code and present its structure and the procedure of the rate assignment which minimize the error between the optimal rate and the coding rate. Then the training algorithm of the transform trellis code is described. Next, in Chapter 3, the first-order AR Gaussian source (correlation coefficient is 0.9) is used to compare the performance of the trained transform trellis code with others. The results are also compared with the rate-distortion bound at rate from 0.07 to 1.0 bits/sample. In Chapter 4, we apply this proposed scheme to encode speech data with window functions and gain adaptation. The window functions are used to suppress the blocking effects, which produce annoying noise in transform coding : a rectangular window is used as an analysis window before the transform and a trapezoidal window is used as a synthesis window after the inverse transform. Gain adaptation is used to adapt this codebook to the widely varying input power. Finally in Chapter 5, we present the summary and conclusion of this paper.

2. The Trained Transform Trellis Code

In this chapter, we first describe the transform trellis code which is proved by Mazor and Pearlman [1,2] as an optimal source code for stationary Gaussian sources with a squared-error distortion measure. Then we present the code construction procedure and our training algorithm of the transform trellis code.

2.1 Transform Trellis Code

Let the input sequence z be N samples from a zero-mean Gaussian source. Let Γ be a unitary transform (Karhunen-Loeve Transform) that

decorrelates $\underline{z}$ through $\underline{u}=\Gamma^T\underline{z}$, where $\Lambda_{\underline{u}}=\Gamma^T\Phi_{\underline{z}}\Gamma=[\lambda_i\delta_{ij}]$ is the (diagonal) covariance matrix of $\underline{u}$, which is also Gaussian, and $\{\lambda_i\}^N$ is the set of eigenvalues associated with covariance matrix of $\underline{z}$, $(\Phi_{\underline{z}})$. From these λ_i's, we can determine the trellis code. The encoder searches the trellis codebook for the transformed coefficients to find the minimum distortion path and then sends the corresponding q-level path-map, $\underline{x}$ to the channel. If we assume the channel is noiseless. Then the decoder, which stores the replica of the codebook, uniquely determines the codewords, $\underline{v}$, from the received path-map, $\underline{x}$. Finally, by taking the inverse transform, we can get the time-domain reproduced sequence, $\underline{y}$. Fig. 1 shows the block diagram of this transform trellis coding system. Since Γ is an invertable mapping and preserves the mutual information, and since it is a unitary transform, it preserves the squared-error [1]. Therefore it enables the solution of the rate-distortion function[7] in the transform domain. For a given desired rate R, we can find the rate-distortion parameter θ

$$D_\theta=\frac{1}{N}\sum_{i=1}^{N}\min(\theta,\lambda_i) \tag{1}$$

$$R_N(D_\theta)=\frac{1}{N}\sum_{i=1}^{N}r_i \tag{2}$$

$$r_i=\max\left(0,\frac{1}{2}\log_2\frac{\lambda_i}{\theta}\right) \tag{3}$$

If $\theta<\min\{\lambda_i\}$, then $N_c=N$, where N_c is the number of letters which have received non-zero rate. If $\min\{\lambda_i\}\le\theta<\max\{\lambda_i\}$, then $N_c<N$. The remaining $N-N_c$ letters which receive zero rate are set to zero. If $\theta\ge\max\{\lambda_i\}$ then R is zero.

Now, the optimal distribution function, $P_N(\underline{v})$, of the transformed reproduction sequences, $\underline{v}$, is essential in the construction of random codes. This distribution is Gaussian with independent, identically distributed (Gaussian) components. In defining γ as the set of N_c indices ($N_c\le N$) for which $\lambda_i\ge\theta$, the optimal distribution function $P_N(\underline{v})$ is explicitly

$$P_N(\underline{v})=\prod_{i\in\gamma}\frac{1}{\sqrt{2\pi(\lambda_i-\theta)}}\exp\left\{-\frac{V_i^2}{2(\lambda_i-\theta)}\right\}\prod_{i\notin\gamma}\delta(V_i) \tag{4}$$

That is, the coefficients with indices in γ have variances $\lambda_i-\theta$, whereas those with indices not in γ have zero variances. The latter set can be optimally encoded by mapping each member to zero[2].

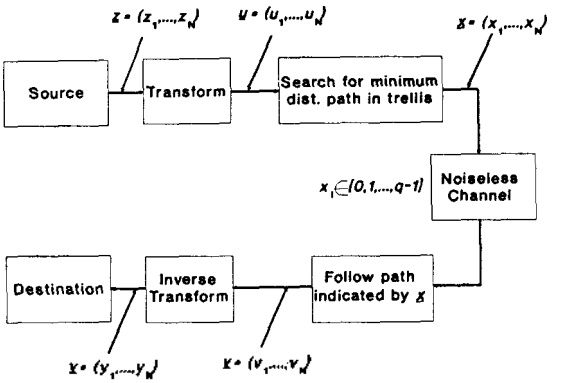


Fig. 1 Trellis coding system.

2.2 The Structure of a Transform Trellis

The trellis structure is completely determined by the trellis depth L, the branching factor q, and the shift register constraint length K, for a given desired rate R and the block size N [1],[2]. 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. The connection between nodes at successive depths corresponds to the transitions between states when a new symbol enters the register. Fig. 2 shows a trellis with branching factor q=2 and shift register length $K=3$. At depth m of the trellis, each branch is

populated with a vector $\underline{v}_m$ of n_m reproduction letters. These letters will be randomly chosen from the distribution of $P_N(\underline{y})$ as in egn.(4).

The selections of q and n_m, the number of reproduction letters at each level m_i are governed by the requirement that the level rate $\log_2 q / n_m$ super-approximate the rate-distortion function rates r_i of the letters at level m_i To accomodate the maximum rate, the branching factor q must then satisfy

$$\log_2 q \geq \max\{r_i\} \tag{5}$$

The trained transform trellis coding scheme needs a fixed codebook whose size is $N_c q^K$. If q is chosen to satisfy egn.(5) it results in too large a codebook. We therefore relax the restrictions of the optimal rate assignment and choose the largest q that satisfies $\log_2 q \leq \max\{r_i\}$ for a given codebook size. We used $K=3$ and q=8. For the given q, we select the largest n_m number of letters on m-th level branch) and L from the following equations(6).

$$r_{th} \cdot \frac{\log_2 q}{n_m} \geq r_i \text{(for all } i \text{ at level m)} \tag{6}$$

$$\frac{L-1}{N} \log_2 q < R \leq \frac{L}{N} \log_2 q \tag{7}$$

The threshold rate, r_{th}, is one in theory. For $r_{th}=1$, a trellis code has too many levels, so the assigned rate $R=(L \log_2 q)/N$ is higher than the desired rate R. So, we get the n_m by adujusting r_{th} until the L satisfies egn.(7).

2.3 A Training Algorithm of the Transform Trellis Code

To implement the transform trellis coding scheme, we have to determine a finite shift register length and block size. The trellis branch symbols are drawn at random from the rate distortion theoretic optimal test channel probability

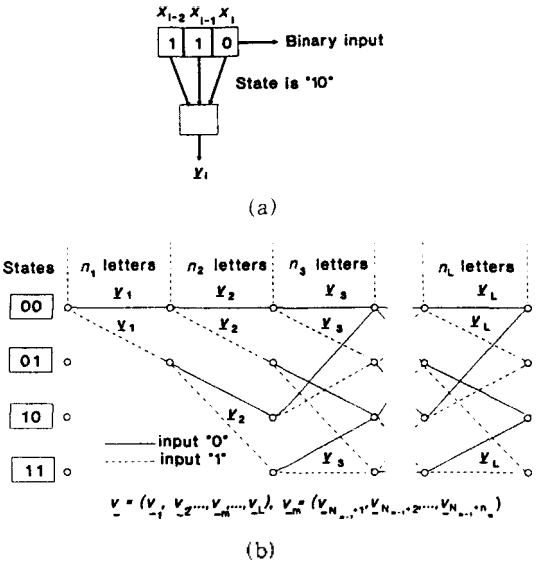


Fig. 2 (a) Binary decoder($K=3$).
(b) Trellis diagram(q=2, $K=3$).

distribution. Optimality of the codebook is guaranteed for infinite N, but for finite N, there is a finite probability that the codewords selected in this way will not be typical of the distribution from which they are drawn. Moreover, for real-world sources which are inherently non-stationary, a random codebook selected through a stationary model can not be well matched. These considerations provide motivation for studying methods for training of a transform trellis code.

We now describe an algorithm for training a trellis which takes account of the commonality of branches in different codewords. We are given $\{\underline{u}_j : j=0, 1, \cdots, n-1\}$, a collection of transform training vectors which is the training sequence, the initial codebook is constructed from the average spectrum calculated from these training vectors in order to approximate the eigenvalue spectrum of the source. The number of letters per branch, $n_m(m=1, 2, \cdots, L)$, and the branching factor, q, are determined by minimizing the difference between the optimal rate and the coding rate as in section 2.2. From egn.(4), we can obtain the prob-

ability distribution, $P_{nm}(\underline{v}_m)$, of a reproduction subvector, $\underline{v}_m$, on the m-th level of the trellis, which is essential to generate the reproduction vectors on the m-th level of the trellis. This distribution is Gaussian with independent (Gaussian) components. All reproduction subvectors, $\underline{b}_m^k$ ($k=1, 2, \cdots, q^K$), on the branches of the trellis at level m, are drawn independently and at random from the distribution function $P_{nm}(\underline{v}_m)$, given by

$$P_{nm}(\underline{v}_m)=\prod_{a\in J_{Nm}}\frac{1}{\sqrt{2\pi(\lambda_a-\theta)}}\exp\left\{-\frac{va^2}{2(\lambda a-\theta)}\right\}$$

where $J_{Nm}=\{N_{m-1}+1, N_{m-2}+2, \cdots, N_{m-1}+n_m\}$ and $N_{m-1}=\sum_{i=1}^{m-1} n_i$. The m-th reproduction subvector, $\underline{v}_m$, has n_m letters with index set J_{Nm}.

We can rewrite $P_N(\underline{v})$ in eqn. (4) as follows

$$P_N(\underline{v})=\prod_{m=1}^{L} P_{nm}(v_m)\prod_{a\notin\gamma}\delta(va)$$

For encoding the j-th block of the transformed source sequence $\underline{u}_j$, we search the trellis for the reproduction word, $\underline{v}_j=\{\underline{v}_{j,1}, \underline{v}_{j,2}, \cdots, \underline{v}_{j,L}\}$, which produces the minimum average distortion. To search a trellis, we need a search function. In our simulations, we used the *M* algorithm which finds at each level of the trellis the best *M* states and extends them to the next level. This process is repeated until the end of the trellis is reached, and the best of the resulting *M* path-maps, $\underline{x}_j=(x_{j,1}, x_{j,2}, \cdots, x_{j,L})$, is transmitted to the decoder.

The m-th path-map symbol of the j-th block, $x_{j,m}$, is uniquely associated with a branch at level m of the trellis and the reproduction subvector, $\underline{b}_m^k$, for some k. The decoder can generate the reproduced sequence $\underline{v}_j$ from the received path-map, $\underline{x}_j$, by walking through the trellis. The m-th level reproduced subvector in v_j, $v_{j,m}$, is identical to one of the reproduction vectors in m-th level, $\underline{b}_m^k$, according to the path-map. After encoding each input block, we know its minimum average distortion and path-map. The resulting distortion and path-map of each block are stored to update the codebook.

After encoding all the training vectors, each reproduction subvector at m-th level and k-th branch, $\underline{b}_m^k$, is updated by centroid of the input training subvectors $\underline{u}_{j,m}$ which have been encoded to $\underline{b}_m^k$. Now, we have a new trellis codebook. For this new codebook, again we apply the same encoding procedure for all the training vectors. Then compare the distortion ratio between the previous and present iteration. If the ratio is small enough, halt. Otherwise, update the trellis codebook by replacing each $\underline{b}_m^k$ in the trellis by the centroid of the $\underline{u}_{j,m}$ which have been encoded to $\underline{b}_m^k$. As in the LBG algorithm with unknown probability distributions, we can improve the performance of the trellis codebook by updating each reproduction subvector from the decoder and then use the new codebook for the training sequence. If there are some reproduction subvectors which are not used, then leave them unchanged.

The following is the detailed training algorithm for a transform trellis code. To describe this training algorithm, we use the following parameters :

m	Index of the trellis level
k	Index of the trellis branch
j	Index of block
n	Maxmum unmber of blocks in a training sequence
L	Total number of levels in trellis
K	Trellis constraint length
q	Number of branches per node
$D(i)$	Squared error distortion per block after i-th iteration
$C(i)$	Trellis codebook of after i-th iteration

$\underline{b}_m^k$ Reproduction vector on m-th level and k-th branch

$\underline{u}_j$ A training vector of j-th block

$\underline{u}_{j,m}$ A subvector of $\underline{u}_m$ at m-th level

$\underline{v}_j$ The reproduced vector of $\underline{u}_j$

$\underline{v}_{j,m}$ A subvector of $\underline{v}_j$ at m-th level

I_m^k The set of indices of which have been assigned $\underline{b}_m^k$

$\|I_m^k\|$ The number of indices in I_m^k

M The number of paths retained at each stage in progressing through the trellis until the best path is selected at the end

f() A partition function which maps $\underline{u}_j$ to $\underline{v}_j$

• Design Algorithm

1. Initialization : Given a distortion threshold $\varepsilon > 0$, a training sequence $\{\underline{u}_{j,m} : j=0, 1,\ldots, n-1 : m=1,\ldots, L\}$, M and a desired rate R_N, construct an initial codebook $C^{(0)}=\{\underline{b}_m^k : m=1,\ldots, L : k=1,\ldots,q^K\}$ from the average spectrum of the training sequence. Set $i=0$ and $D^{(-1)}=\infty$.

2. Given $C^{(i)}$, find the minimum average distortion path of the trellis codebook using the M algorithm which is a partition function of $C^{(i)}$, $f(\underline{u}_j)=\underline{v}_j$. Now, the path of the j-th block is determined. The reproduction vector of the j-th block at the m-th level, $\underline{v}_{j,m}$, is associated with one of m-th level reproduction subvectors, $\underline{b}_m^k$, according to the path-map of j-th block. The block index j is now inserted into $\{I_m^k\}$, the set of block indices associated $\underline{b}_m^k$. Repeat this procedure for all blocks of the training sequence. Compute an average distortion of the i-th iteration $D^{(i)}$:

$$D^{(i)}=\frac{1}{n}\sum_{j=0}^{n-1}\sum_{m=1}^{L}\|\underline{u}_{j,m}-\underline{v}_{j,m}\|^2$$

3. If $(D^{(i-1)}-D^{(i)})/D^{(i)}\le\varepsilon$, then halt with $C^{(i)}$ as the final codebook, otherwise go to 4.

4. Update trellis codebook $C^{(i)}$ by replacing each reproduction vector $\underline{b}_m^k$ by the centroid of the training sequence subvectors $\underline{u}_{j,m}$ which have been encoded to $\underline{b}_m^k$,

$$\underline{b}_m^k=\frac{1}{\|I_m^k\|}\sum_{j\in I_m^k}\underline{u}_{j,m}$$

If the set of indices I_m^k is empty, then do not change reproduction vector $\underline{b}_m^k$. Replace $i=i+1$ and go to 2.

3. Encoding AR Gaussian Data

In our simulation, we generate the first-order autorgressive Gaussian sequence where the correlation coefficient $a_1=0.9$,

$$Z_n=a_1Z_{n-1}+G_n$$

and G_n is a zero mean, unit variance, independent and identically distributed Gaussian source. This source is useful as a mathematical model for some real data sources and its information theoretic optimal performance bounds as described by the distortion rate function are known.

We constructed the transform trellis codebook from an average spectrum of a training sequence. The trellis code is designed for $K=3$, $N=128$, and its corresponding codebook size is Nxq^K. The number of multiplications per letter S, is determined by

$$S=(M\sum_{m=1}^{L}n_mq)/N$$

where M refers to the parameter M in the M algorithm. There is no initial build-up of the trellis from the all-zero state, large distortions inevitably occur from the limited codeword choice during this phase. Instead, a full state trellis is started with full search and the initial state path-map is supplied as overhead. The overhead for $K=3$, $q=8$ and $N=128$ is $\lceil(K-1)\log_2 q\rceil/N$,

which in our case is 0.05 bits/sample. Also, we used the discrete cosine transform (DCT) [9] instead of the Karhunen-Loeve transform (KLT), because of its fast computation and closeness in perfomance to the KLT for the finite-order Marcov process.

To test the perfomance of this trained transform trellis scheme, we used a first-order autoregressive Gaussian source (ρ_1=0.9) with 5000 blocks for the training sequence and 1000 blocks for the test sequence. Each block is 128 samples.

In Table 1, we compare the achieved SNR and its theoretical SNR (rate-distortion bound), for the various rates including the overhead of 0.05 bits/sample. We used a full search for the first several levels which have not received sufficient rates, and the M algorithm for the rest of levels. The achieved SNR for the test sequence is from 0.6 to 1.4 dB less than the theoretical SNR, depending on the rate.

In Table 2, we compare the perfomance of our coding scheme and others [10]-[12] for the test sequence of same source at a rate of 1 bits/sample with a similar number of multiplications per sample. The SNR of the full-state transform trellis code is from 0.15 dB to 0.9 dB higher than that of any other and it took fewer calculations and larger dimension. For the full-state trellis, we searched the beginning five levels exclusively and then extended the eight best paths at each level. For the trellis which starts from zero state, we chose M=10 to maintain a number of multiplications per sample, S, similar to that of the full-state trellis. For the same structure of the trellis, SNR of the trained trellis code is from 0.34 dB to 1.2 dB higher than that of the untrained trellis code.

Table 1. Performance of the trained transform trellis code for the test sequence of an AR 1 source (ρ_1=0.9)

Rate	S	SNR(dB) (Experimental)	SNR(dB) (Theoretical)
0.12	22.5	4.2	4.8
0.16	28.5	5.2	5.8
0.30	35.0	7.0	7.9
0.56	61.0	9.5	10.4
0.75	72.5	10.6	11.7
1.06	81.5	12.1	13.5

Table 2. Performance of the trained transform trellis code and others for the test sequence of an AR 1 source (ρ_1=0.9) at rate 1.0 bits/sample.

Dimension	Test(dB)	Method	S	Domain
4	11.73	CH-FSVQ		Time
6	11.5	VTE		Time
7	11.2	VQ	128	Time
6	11.96	PTWC	96	Time
7	11.56	NPTWC	128	Time
128	9.7	TTC	80	Transform
128	10.9	TTTC	80	Transform
128	11.77	TTC(*)	82	Transform
128	12.11	TTTC(*)	82	Transform

CH-FSVQ : conditional histogram finite state vector quantization[10]
VTE : vector trellis encoding(labeled-states)[10]
VQ : vector quantization[11]
PTWC : predictive trellis waveform code[12]
NPTWC : non-predictive trellis waveform code[12]
TTC : transform trellis code
TTTC : trained transform trellis code
S : a number of multiplication per sample
(*) : a trellis codebook which starts from full state

4. Encoding Speech Data

All the speech data used in this simulation is collected from the local FM station in U.S.A. and other. Each sample is low-pass filtered to 3.2 kHz and sampled at 8 kHz. To train the trellis, we used 6938 blocks of 128 samples collected from 16 men and 14 women. To test the trained transform trellis code, we used 1948 blocks collected from 5 men and 3 women which were not used in a training sequence. Table 3 shows the contents and the size of the test sequence. The trellis code is designed for K=3, q=8, and N=128.

For objective measurement, we use both the long-term signal-to-noise ratio(SNR) and the segmental signal-to-noise ratio (SEGSNR) criteria. The long-term SNR is the ratio of the total energy of the source to the total energy of the encoding error ; whereas, the SEGSNR(j) is the average SNR computed for each block.

In section 4.1, we apply the trained transform trellis code to actual speech signals at a rate of 0 bits /sample. For this non-adaptive scheme, the reproduced speech signals produced the "clicking" noise due to blocking effects which are typical in transform coding. In section 4.2, we introduce adaptive schemes to achieve good perceptual quality and high SNR. In section 4.3, the performance of the non-adaptive and that of the adaptive coding scheme are compared.

Table 3. The number of samples and the contents of the sequence.

Data	Samples	Contents
man1	30208	... so easy. Subscribe to WMHT one of the
man2	32000	Your membership helps make the progress possible it's easy to think ...
man3	32000	It takes a start broadcast such as this to unlock one such ...
man4	32000	Reagan's comments came during his weekly radio address he says renewed aids to contras...
man5	17729	Thieves who rob friends deserve jail.
woman1	32000	It's always nice to know that you're having a part in the programs that goes on here.
woman2	32000	And to sweeten in the pot still a little bit we have a fantastic ...
woman3	41344	It's now twenty minutes passed ten o'clock and we'll continue music ...

4.1 A Trained Transform Trellis Code

We applied the trained transform trellis code without adaptations (BASIC scheme) to the speech data at rate 1.01 bits /sample.

The overall rate R is defined as

$$R=\frac{B_{path}+B_{side}}{N-N_{over}}$$

$$B_{path}=\sum_{m=1}^{L} r_m n_m$$

$$=\sum_{m=1}^{L} (\log_2 q / n_m) n_m$$

$$=L \log_2 q$$

where B_{path} is the number of bits for the path-map, B_{side} is the number of bits for the side information, N is the number of samples in one block and N_{over} is the number of overlapped samples between blocks.

4.2 An Adaptive Trained Transform Trellis code

The transform trellis code is optimal when the block size, N, and branching factor, q, goes to infinity. The implementation of such a code requires finite values of N and q. In speech coding, we usually assume that speech is quasi-stationary, i.e., stationary for short time segments. Clearly this contradicts the stationary assumption needed for the optimal transform trellis code. Also, the quasi-stationarity of speech implies that N can not be too large if we desire the speech signal to be quasi-stationary. Thus, we see that in practical implementation some difficulties will be encountered.

This subsection presents our attempts to overcome these difficulties. First, we present the system block diagram of the adaptive version of the trained transform trellis code. Then, the window functions are introduced to remove the blocking effects which occurs in transform coding. Finally, gain adaptation is presented to adapt the codebook to the widely varying input signal power.

4.2.1 System Block Diagram

Fig. 3. describes the block diagram of the adaptive scheme. The adaptive scheme used window functions and gain adaptation. Let z_j be j-th

block of a time-domain sequence from the output of source $\underline{z}$ and consists of 128 samples. First, we multiply $\underline{z}_j$ by the analysis window, then take the DCT to produce the approximately uncorrelated coefficients $\underline{u}_j$. The analysis window contains 128 samples, and four samples are overlapped with both neighboring blocks as in Fig. 4. We used a buffer to calculate the j-th block gain factor, σ_j, and quantized σ_j to $\hat{\sigma}_j$ using a 8-bit scalar quantizer. The index of the quantized block gain factor ($I_{\hat{\sigma}_j}$) will be sent as a side information to receiver. Then coefficients in $\underline{u}_j$ are normalized by its quantized block gain factor ($\hat{\sigma}_j$). For these normalized coefficients, the trained transform trellis codebook is searched to find the minimum distortion path with the M algorithm. So, the path-map ($I_{\text{path-map}}$), and the index of the quantized block gain factor ($I_{\hat{\sigma}_j}$) are sent through the channel. If we assume the channel is noiseless, then the decoder, which stores a replica of the codebook, uniquely determines the codeword from the received path-map ($I_{\text{path-map}}$). We can get the reproduced coefficients $\underline{v}_j$ by multiplying the codeword by the quantized block gain factor $\hat{\sigma}_j$. By taking the Inverse Discrete Cosine Transform of the reproduced coefficients ($\underline{v}_j$), we can get the reproduced signal. Finally, we can get the reproduced output signal ($\underline{y}_j$) from the synthesis window which has a trapezoidal shape where four samples are overlapped from adjacent blocks as in Fig. 4.

4.2.2 Window Functions

The DCT is closely related to the $2N$ point DFT [13] and the blocking effect is less than that of DFT. Thus, the DCT produces less noticeable boundary effects than the DFT. This is an additional advantage of the DCT besides the fact that it is "close" in performance to the KLT. But, the proper choice of the window functions in the analysis/synthesis involving the DCT can be instrumental in further reducing the boundary effects [13].

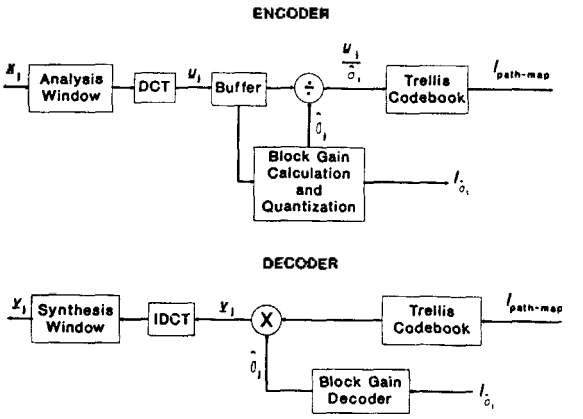


Fig. 3 Block diagram of the adaptive trained transform trellis code.

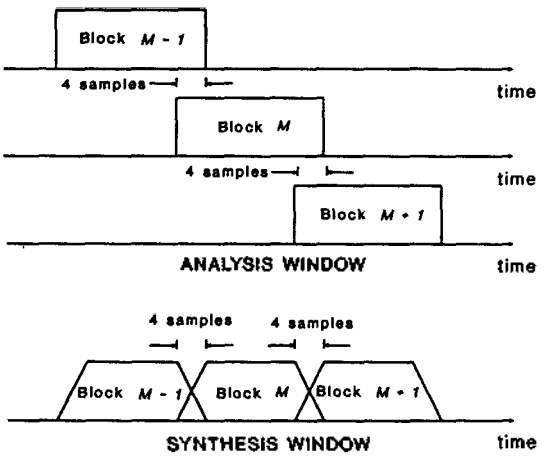


Fig. 4 Analysis and Synthesis window.

In our system, we use an overlapped rectangular window as the analysis window which was placed before the DCT in the encoder, and an overlapped trapezoidal window, which has been found to be very useful for low bit-rate coding [13], as the synthesis window which is located in the decoder after the inverse DCT(WINDOWED scheme). Fig. 4 illustrates the analysis and synthesis windows. By allowing a small(4 samples/block) overlap between the successive blocks being coded, a significant reduction of blocking effect noise is achieved without signifi

cantly lowering the number of bits available for encoding each block. Clearly, if this number of bits is significantly reduced, the overall incease in quantization noise can offset whatever blocking effects reduction is achieved by the overlapped processing.

4.2.3 Gain Adaptation

The basic idea of the gain adaptation is to adapt the codewords in the codebook according to the widely changing input power of speech data. Then no matter how the input level varies, the norm, or rms energy level, of quantization error vectors will tend to be proportional to norm of signal vectors. Hence, the resulting signal-to-noise ratio will be constant for all input levels. This implies that the ideal gain-adaptation can accommodate input signals with large dynamic range. In out experiment, we normalized the input data and keep the codebook fixed (GAIN-ADAPTIVE scheme).

For the transformed input sequence $\underline{u}_j$ which has N samples, we calculate the j-th block gain factor (σ_j) from the buffered one block data. The block gain factor is defined as follows :

$$\sigma_j = \sqrt{\sum_{i=1}^{N} u_{j,i}^2}$$

where $u_{j,i}$ is the i-th coefficient of the j-th block of the transformed data. Then we used a scalar quantizer to quantize σ_j and the quantized $\hat{\sigma}_j$ is used to normalize $u_{j,i}$. This normalized input ($u_{j,i}/\hat{\sigma}_j$) is encoded by a trained trellis codebook.

4.3 Simulation Results

For the BASIC scheme, the overall rate, R, is 129/128 bits/sample (for the value of the B_{path}=129, B_{side}=0, N=128 and N_{over}=0). Perceptually, the reproduced speech contained a lot of "clicking" noise, but it was intelligible. When coded at low rate, this "clicking" noise due to the blocking effects, is typical in transform coding. Table 4 displays the SEGSNR and SNR of each speaker in the outside training sequence. The objective quality among different speakers varied widely, but the perceptual quality differences were much less pronounced.

Table 4. Performance of the BASIC scheme at rate 1.01 bits/sample.

Data	SEGSNR(dB)	SNR(dB)
man1	13.55	14.52
man2	12.72	13.37
man3	9.57	10.92
man4	8.19	7.68
man5	9.30	6.52
woman1	13.48	13.22
woman2	9.92	10.78
woman3	9.99	14.37
AVERAGE	10.84	11.42

When applying the BASIC scheme to speech coding, we can hear the "clicking" noise due to the bolcking effect. To remove this undesirable noise, we used the WINDOWED scheme. The number of overlapped samples between blocks is four. The resulting rate of the WINDOWED scheme is 1.04 bits/sampole (R=(129)/(128-4) for the value of the B_{path}=129, B_{side}=0, N=128 and N_{over}=4). Table 5 shows the SEGSNR and SNR of each speaker. The average SNR and SEGSNR of the WINDOWED scheme is 11.5dB respectively. The objective quality of the WIDOWED scheme is close to that of the BASIC scheme, but we almost can not hear the "clicking" noise from the reproduced speech. However, artifacts consisting of some clicks and warbling are present. The rate of the WINDOWED scheme is 0.03 bits/sample higher than that of the BASIC scheme because of the four sample overlap of blocks.

In speech coding, the goal is to generate the least audible noise possible. To achieve this goal, we used gain adaptation to adapt the codebook for widely varying input power and window functions for the trained transform trellis code (GAIN-ADAPTIVE scheme). We used an 8 bit scalar quantizer to quantize the block gain factor. The overall rate for this gain-adaptive trained transform coding is 1.1 bits/sample (R=(129+8)/(128-4) for the value of the B_{path}=129, B_{side}=8, N=128 and N_{over}=4). The achieved rate of the GAIN-ADAPTIVE scheme is 0.09 and 0.06 bits/sampde higher than that of the BASIC scheme and the WINDOWED scheme. With this increased rate, the average SEGSNR and the SNR of the GAIN-ADAPTIVE scheme is 12.36 dB and 11.82 dB respectively. As expected, the GAIN-ADAPTIVE scheme achieved the best objective and perceptual quality with relatively small increase in rate. Table 6 displays the performance of each speech data in the test sequence. The resulting average SEGSNR and SNR of the GAIN-ADAPTIVE scheme is 1.52 dB and 0.4 dB better than that of the WINDOWED scheme. The clicks and warbling are infrequent and barely noticeable.

Table 5. Performance of the WINDOWED scheme at rate 1.04 bits/sample.

Data	SEGSNR(dB)	SNR(dB)
man1	13.48	14.56
man2	12.61	13.55
man3	9.39	11.43
man4	7.93	7.71
man5	9.19	6.45
woman1	13.60	13.42
woman2	9.75	10.62
woman3	9.91	14.23
AVERAGE	10.73	11.50

Table 6. Performance of the GAIN-ADAPTIVE scheme at rate 1.1 bits/sample.

Data	SEGSNR(dB)	SNR(dB)
man1	15.22	14.76
man2	13.69	13.64
man3	12.69	14.14
man4	8.90	7.82
man5	10.22	7.74
woman1	14.29	12.32
woman2	12.04	9.96
woman3	11.80	14.19
AVERAGE	12.36	11.82

5. Summary and Conclusion

We have proposed a new training algorithm for the transform trellis code which uses the M algorithm to search the trellis codebook and the LBG algorithm to update the trellis codebook. Our trellis codebook is constructed for K=3, q=8 and N=128, and its corresponding codebook size is $N_{cb}q^K$. To compensate for the large distortions, due to the finite K and q, occuring at the initial build-up of the trellis, we designed the trellis codebook which starts from full state and searched the first several levels exhaustively which have not received sufficient rates. We used a first-order autoregressive Gaussian source (ρ=0.9) to test the performance of this trained transform trellis coding scheme with others. The achieved SNR of the trained transform trellis code is higher than that of any other [10]~[12] with fewer calculations and large dimension. For the same structure of the trellis, the achieved SNR of the trained transform trellis code is from 0.34 to 1.2 dB higher than that of the untrained transform trellis code. The comparision between the trained and the random trellis code is shown in Table 7 where the full-state trellis is denoted by (*). Then, in the applications of speech coding, we used 6938 blocks of 128 samples collected from 16 men and 14 women as a training sequence, and

1948 blocks from 5 men and 3 women which were not in the training sequence, are used as a test sequence. When applying the proposed scheme to actual speech signals without adaptations (BASIC scheme), we can hear the "click" noise due to blocking effects. This is an annoying noise to listeners. To remove this undesirable noise, we used a rectangular window before the DCT in the encoder and a trapezoidal window after the inverse DCT in the decoder(WINDOWED scheme). With window functions, we could achieve a good perceptual quality. But the SEGSNR of the WINDOWED scheme is only 0.08dB more than that of the BASIC scheme. The improvement of the WINDOWED scheme is not so successful in terms of SEGSNR.

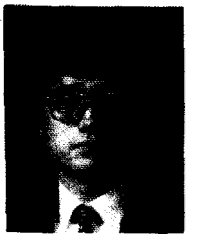
To achieve further improvement, we used window functions and gain adaptation (GAIN-ADAPTIVE scheme). Gain adaptation is employed to adapt the codebook for widely varying input power. Each block transform coefficient is normalized by its block gain factor and the normalized coefficients are used to encode. We can get the reproduced transformed coefficients by multiplying the block gain factor to the decoded coefficients from the received path-map. The block gain factor has to be sent as a side information. The performance of each scheme is shown in Table 8. As we have expected, this GAIN-ADAPTIVE scheme produced the highest SEGSNR, SNR and a good perceptual quality with the increase of a small overthead.

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, November 1983.
2. ——, "A Transform Trellis Code with Applications to Speech," IEEE Trans. Comm., Vol.COM-33, pp. 1109-1116, October 1985.
3. S.Farkash, W.A. Pearlman and D.Malah, "Transform Trellis Coding of Images at low Rates with Blocking Effect Removal," EUSIPCO, Grenoble, France, September 1988.
4. 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.
5. Y. Linde, A. Buzo and R.M. Gray, "An Algorithm for Vector Quantization," IEEE Trans. Comm., Vol. COM-28, No.1, pp.84-95, January 1980.
6. R.Toy and W.A. Pearlman, "Backward Adaptation for Transform Trellis Coding of Speech," Proceedings of the 1987 ICASSP, Vol.4, pp.2209-2212, April 1987.
7. T. Berger, Rate Distortion Theory, Englewood Cliffs, NJ : Prentice Hall, 1971.
8. G.D. Forney Jr., "The Viterbi Algorithm," Proceedings of the IEEE, Vol.61, No.3, pp.268-278, March 1973.
9. N. Ahmed, T. Natarajan and K.R. Rao, Orthogonal Transforms for Digital Signal Processing, New York : Springer-Verlag, 1975.
10. C.D. Bei and R.M. Gray, "Simulation of Vector Trellis Encoding Systems," IEEE Trans. Comm. Vol.COM-34, No.3, pp.214-218, March 1986.
11. R.M. Gray, "Vector Quantization," IEEE ASSP Magazine, pp.4-27, April 1984.
12. E. Ayanoglu and R.M. Gray, "The Design of Predictive Trellis Waveform Coders Using the Generalized Lloyd Algorithm," IEEE Trans. Comm., Vol, COM-34, No.11, pp.1073-1080, November 1986.
13. J.M. Tribolet and R.E. Crochiere, "Frequency Domain Coding of Speech," IEEE Trans. on ASSP, Vol.ASSP-27, No.5, pp.512-530, October 1979.

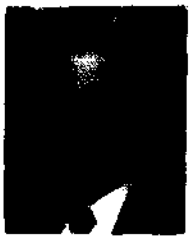
▲**Dong-Youn Kim** received the B.S. and M.S. degrees in electrical engineering from the Yonsei University in 1981 and 1983, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, U.S.A., in 1990.

Since March 1991 he has been with the Department of Biomedical Engineering at Yonsei University, where he is currently an Assistant Professor. His research interests are in digital signal processing, and speech and image coding, and medical image processing.



▲**Yong Seo Park** was born in Seoul, Korea on January 22, 1958. He received the B. E. degree in 1982, the M.E. degree in 1984, and the Ph.D. degree in 1988 from the department of electrical engineering of Yonsei University, Seoul, Korea.

Since 1989, he has been a member of the faculty of Kyungwon University, Seoungnam, Korea where he is now an Assistant Professor. His current research concerns spread spectrum, multiuser communications systems, spread spectrum for mobile communications, and signal processing.



▲**Keum-Chan Whang** received the G.S. degree in electrical engineering from the Yonsei University in 1967, and the M.S. and the Ph.D. degrees in electrical engineering from Polytechnic Institute of New York, NY, U.S.A., in 1975 and 1979.

Since September 1980 he has been with the Department of Electrical Engineering at Yonsei University, where he is currently an Professor. His research interests are in spread-spectrum communications and analog signal processing

▲**Willian A. Pearlman** received the S.B. and S.M. degrees in electrical engineering from the Massachusetts Institute of Technology in 1963 and the Ph.D. degree in electrical engineering from Stanford University in 1974.

From 1963 to 1965 and 1971 to 1974 he was employed by Lockheed Missiles and Space Company in Sunnyvale, CA, where he analyzed and designed satellite communications and telemetry systems and served as a consultant on problems in optics and communications. Between 1965 and 1971 he was employed by GTE-Sylvania in Mountain View, CA, where he developed techniques of detection and parameter estimation for spread-spectrum communications, radar, and ECM systems. In 1974, he left industry and spent five years as an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Wisconsin-Madison and then joined the Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute, where he is currently a Professor. During the 1985-1986 academic year, he was on sabbatical leave as a Visiting Associate Professor and Lady Davis Scholar in the Department of Electrical Engineering at the Technion-Israel Institute of Technology, Haifa, Israel. His research interests are in information theory and source coding, image and speech coding, and image processing.

Dr. Pearlman is a member of the Optical Society of America, SPIE, Sigma Xi, and Tau Beta Pi.