[10] W. A. Pearlman, "Sliding-block and random source coding with constrained size reproduction alphabets," *IEEE Trans. Commun.*, vol. COM-30, pp. 1859–1967, Aug. 1982.
[11] A. Chekima, "Coding bounds for memoryless analog sources with constrained reproduction alphabets," Ph.D. dissertation, Electrical, Computer, and Systems Eng. Dept., Rensselaer Polytechnic Institute, Troy, NY, May 1984.
[12] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform Theory*, vol. IT-28, pp. 129–137, Mar. 1982.
[13] J. Max "Quantizing for minimum distortion," *IRE Trans. Inform. Theory*, vol. IT-6, pp. 7–12, Mar. 1960.
[14] W. C. Adams, Jr., and C. E. Giesler, "Quantizing characteristics for signals having Laplacian amplitude probability density function," *IEEE Trans. Commun.*, vol. COM-26, pp. 1295–1297, Aug. 1978.
[15] W. A. Pearlman and G. H. Senge, "Optimal quantization of the Rayleigh probability distribution," *IEEE Trans. Commun.*, vol. COM-27, pp. 1101–1112, Jan. 1979.
[16] R. E. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 460–473, July 1972.

Near Merging of Paths in Suboptimal Tree Searching

HSUEH-MING HANG AND JOHN W. WOODS, MEMBER, IEEE

Abstract—**The near merging of paths in tree searching is explored. This near merging can degrade coding performance when a suboptimal search algorithm is used. The problem is first identified and then a feasible solution is presented. Some examples from image source coding using the (M, L) algorithm are given.**

I. Introduction

Because of their easy implementation and near optimum performance, instrumentable tree codes have been the object of rather extensive study. The encoding of one-dimensional discrete memoryless sources and speech signals was successfully carried out by Anderson and his co-workers [1]–[4]. Along the same line, Modestino *et al.* obtained very good results on two-dimensional image coding [5], [6] using a regenerative search scheme [7] to be described below. The conventional one-dimensional search scheme is nonregenerative; this saves considerable computation but is afflicted by a new phenomenon that we call "near merging."

It was generally believed that tree coding performance might be improved by increasing the search length. As will be shown below, this assertion is not true if the encoder is a stable autoregressive moving-average (ARMA) filter and the search is suboptimal. Furthermore, if a conventional (M, L) search is used, increasing the search length can degrade coding performance significantly. This degradation comes from the near merging of paths which will be identified and discussed in Section III. Section II summarizes the basic definitions of tree coding and the (M, L) algorithm. A nonregenerative image coding example which demonstrates the effect of near merging and the means to avoid it is shown in Section IV.

II. Tree Coding

A *tree code* is a code whose words are generated by a regular tree structure having r symbols on each branch and b branches out of each node. The codewords are formed by concatenating the symbols encountered along each path in the tree. If the paths are all L branches long, we say the tree has *depth* or *delay* L. A tree code of depth L with b branches per node and r symbols per branch contains codewords of length $n = Lr$ and rate $R = (\log b)/r$ nats per source symbol [9, ch. 6]. The range of the reproducing symbols can be either discrete or continuous. A sampled image will be represented by a two-dimensional array $s(m, n)$ whose value corresponds to the gray level of pixel (m, n), so the continuous case is of interest here.

It was proved by Jelinek that for a discrete-time memoryless source and single-letter fidelity criteria, the $R(D)$ performance limit can be approached arbitrarily closely by tree codes of sufficiently large tree depth [8], [9 ch. 6]. However the theorem is proved using randomly generated tree codes, which are not realistically instrumentable. The implementation problem for tree coding can be solved by using a structured path generator such as an ARMA filter. The conventional trellis or convolutional encoder is a special case of an ARMA filter, which has only an MA portion and operates on a finite range of input values. We can also use an ARMA filter together with real number operations to encode continuous speech or image waveforms [3]–[5].

Various code search techniques have been invented for tree encoding [10]. As one would expect, the exhaustive search scheme, which compares every possible path in the tree with the source signal, has the best performance. However, the computation associated with an exhaustive search grows exponentially with search depth and, hence, it is not instrumentable in practice. Many other search algorithms reduce the computations substantially; for example, the stack algorithm [2], the 2-cycle algorithm [11], and the (M, L) algorithm [1]. The (M, L) algorithm is of value because of its reasonably good performance and moderately strong properties in all respects [2], [10]. Therefore, a modified version of the (M, L) algorithm is used in this correspondence.

The (M, L) algorithm is normally applied using a nonregenerative search. A *nonregenerative search* algorithm is characterized by not regrowing the code tree after each branch release. The new branches are just concatenated onto the old tree. In contrast, a *regenerative search* algorithm regrows the entire code tree, back to the released branch, at each new data point.

The operation of the conventional nonregenerative (M, L) search is as follows. The encoder views all the branches it will ever view at a given tree-level before advancing any further into the tree. The M best paths at each level are kept according to their associated cumulative distortions. Then one selects the next best M branches stemming from the current retained branches. After depth L is reached, the level-one branch of the best surviving path is released. Then the new branches are extended from the surviving paths whose root branch is the released branch. The tree thus advances to a new tree level, and the level-two branch of the tree becomes the new released branch.

The regenerative (M, L) search still follows the (M, L) algorithm rules in the growing of the tree, i.e., M paths are retained at each tree-level and the decision is made at the depth L. However, the entire code tree has to be reconstructed repetitively for every released branch in the regenerative search. If the search is exhaustive, the nonregenerative scheme will give the same results as the regenerative one; otherwise it may not.

III. Near-Merging of Paths

A general ARMA filter can be described by a recursive difference equation,

$$s(k) = \sum_{i=1}^{I} c_i(k) s(k-i) + \sum_{j=1}^{J} d_j(k) u(k-j). \qquad (1)$$

It is well-known that the finite discrete-valued convolutional encoder displays the *path merging phenomenon*, that is the paths in the tree can merge after a certain delay [12], [13]. This phenomenon takes place because the input to the encoder is a

Manuscript received February 25, 1983; revised August 16, 1983.
The authors are with the Electrical, Computer, and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY 12181.

0018-9448/84/0500-0567$01.00 ©1984 IEEE

finite field. Consequently, the number of possible branches at each tree level is fixed, and the tree representation can be reduced to the trellis representation. However, if a real-valued ARMA filter is under consideration, the values of the encoder states are real and distinct for nontrivial cases, and thus the number of possible states becomes infinite. Nevertheless, a "near merging" phenomenon, similar to that in convolutional codes, also happens to the real-valued ARMA encoder if certain conditions are satisfied.

Although a real-valued ARMA filter has an infinite number of possible states, the outputs are bounded for a bounded-input bounded-output (BIBO) stable filter. An ARMA filter is thus a bounded operator which maps a bounded set U (i.e., the range of inputs) to another bounded set S (i.e., the range of outputs). Since the sets U and S are finite dimensional and all valid norms in finite-dimensional spaces are equivalent [14], the boundedness in the above statement can thereby be defined using any specific norm. If all possible output sequences of an ARMA filter are distinct in any finite time interval, then the number of branches (outputs) at a tree level increases exponentially in time. As a consequence, the outputs (i.e., the tree branches) become dense in a subset of S, so that the distance (norm) between two nearest branches becomes increasingly small as the tree grows.

The Shannon source coding theory asserts that the minimum possible distortion of a well-defined source at a fixed coding rate R has to be greater than the distortion rate function $D(R)$ [9]. If the distortion difference between two outputs (coded values) is ϵ, which is much less than $D(R)$, there is no need to distinguish between these two branches since they give essentially the same distortion. Hence we say these two branches are *ϵ-indistinguishable*.

The above definition not only depends on the values of two branches in the code tree, but also involves the source sequence. If a single-letter distortion is of concern, and the distortion measure satisfies the norm criteria [14], then the above definition can be replaced by the following definition.

Definition: If the norm (induced from the distortion measure d) of the difference between two branches is less than ϵ, they are called *relative ϵ-indistinguishable*.

It is easy to see that relative ϵ-indistinguishability is stronger than ϵ-indistinguishability. Indeed, consider two branches $s^1 = (s_1^1, s_2^1, \cdots, s_r^1)$ and $s^2 = (s_1^2, s_2^2, \cdots, s_r^2)$ and a source sequence $u = (u_1, u_2, \cdots, u_r)$); the distortions associated with branches s^1, s^2 are $\Sigma|s_i^1 - u_i|$ and $\Sigma|s_i^2 - u_i|$. The difference of distortion between s^1 and s^2 is

$$\sum_{i=1}^{r} |s_i^1 - u_i| - \sum_{i=1}^{r} |s_i^2 - u_i| = \sum_{i=1}^{r} \left\{ |s_i^1 - u_i| - |s_i^2 - u_i| \right\} \leq \sum_{i=1}^{r} \left\{ |s_i^1 - s_i^2| \right\}.$$

The right-hand side of the above expression is the distortion of the difference of s^1 and s^2. Therefore relative ϵ-indistinguishability implies ϵ-indistinguishability.

If two branches are relative ϵ-indistinguishable, they are "almost the same" and hence we only have to evaluate one of them in the search process. This can be interpreted geometrically. The space S is partitioned into ϵ-neighborhoods and the branches within one neighborhood can be regarded as the same, therefore the distortion associated with that neighborhood can be computed by simply using one representative branch in it. In a long tree, many branches may cluster in one neighborhood and, thus, we may reduce the computations greatly by considering the representative branches only.

Nevertheless, two tree paths may be ϵ-indistinguishable at a certain tree level and have very different succeeding branches in the future. The coding performance lies in the global behavior of the path, thus neither one of the above branches can necessarily be neglected. To drop the ϵ-indistinguishable branches, some further restrictions have to be introduced. Two *trees* are said to be *relative ϵ-indistinguishable* if for any path in one tree, we can find a path in the second tree such that the branches on these two selected paths are relative ϵ-indistinguishable. Two *paths* in a tree code are called *relative ϵ-merged* at tree level J, if the trees stemming from these two paths are relative ϵ-indistinguishable. The path generator has the relative ϵ-merging property, if when the two paths generated by this path generator are relative ϵ-indistinguishable at level J, then they are also (relative) ϵ-merged at level J.

Again, relative ϵ-merging implies ϵ-merging if the distortion measure is a valid norm. In light of the above definitions, two paths are effectively "the same" after level J if they are ϵ-merged at level J, and the average distortion difference between these two paths is less than ϵ/r per symbol after level J. Once we can show that the path generator has the relative ϵ-merging property, we can drop the worse of two branches whenever they are relative ϵ-indistinguishable, and the final result is essentially unaffected.

To illustrate the use of the ϵ-merging property, consider an ϵ-merging path generator together with the (M, L) search algorithm. Since the M paths retained by the (M, L) algorithm are all good ones, it is likely that some of them are ϵ-merged at a certain level. If the merged paths are kept in the tree, they will result in similar distortions eventually. However, the (M, L) algorithm's ability to explore a path of initially high distortion but lower accumulated distortion at a later part of the tree, is destroyed by this merging phenomenon. In order to remedy this, a certain path separation criterion should be applied to the path generating procedure.

The requirement of a relative ϵ-merging path generator is very strong. However, some practical image tree coding schemes do possess the ϵ-merging property as shown in Section IV. This property can be extended to a larger class of path generators by relaxing the definition of ϵ-merging. For example, the zero input response of an asymptotically stable ARMA filter approaches zero with arbitrary finite initial conditions. Hence the tree codes generated by this path generator still suffer from the near merging effect in the long run.

IV. Some Experiments in Image Coding

The tree search schemes used for one-dimensional signals can be extended directly to images. The tree search order is simply defined to be the row-by-row scanning order and, thus, a one-dimensional-like tree search can be formulated as illustrated in Fig. 1. The mean value of the image is shifted to zero before processing and the boundaries used by the path generator are assumed to be fixed at zero. Following the ARMA filter assumption, the path generator of an image tree encoder can be described by

$$s(m,n) = \sum_{\mathscr{R}^1_{\oplus+}} c_{ij} s(m-i, n-j) + \sum_{\mathscr{R}^2_{\oplus+}} d_{ij} u(m-i, n-j) \tag{2}$$

where

- $u(m,n)$ assumes a finite number of discrete values;
- $s(m,n)$ is the reproduced image intensity;
- $\mathscr{R}^1_{\oplus+}$ is a subset of the nonsymmetric half-plane which excludes the origin; and
- $\mathscr{R}^2_{\oplus+}$ is another subset of the nonsymmetric half-plane which includes the origin.

We can either use a regenerative scheme which regrows the entire tree after releasing each data point, or a nonregenerative scheme which attaches new branches to the old tree. The nonregenerative scheme has much higher computational efficiency, by a factor of the delay length L on the average. Consider using the nonregenerative (M, L) search for image source coding. A popular image model which has a compact filter support (Fig. 2) is

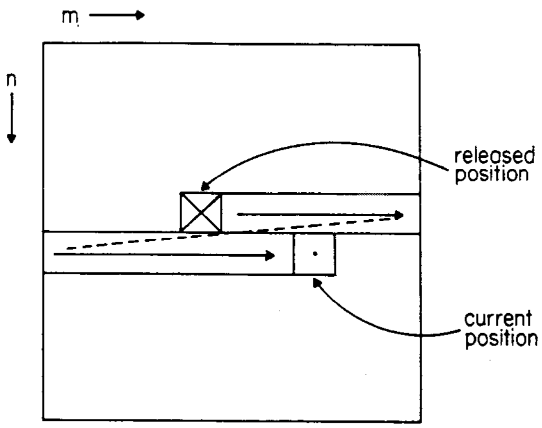

Fig. 1. Geometry of one-dimensional-like tree search.

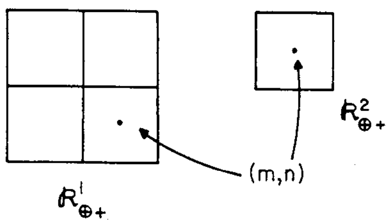

Fig. 2. Compact filter support for simulation.

used [5]. The difference $e(m,n)$ between any two branches, $s^1(m,n)$ and $s^2(m,n)$, in the code tree is

$$\begin{aligned} e(m,n) &= s^1(m,n) - s^2(m,n) \\ &= c_{10}\left[s^1(m-1,n) - s^2(m-1,n)\right] \\ &\quad + c_{01}\left[s^1(m,n-1) - s^2(m,n-1)\right] \\ &\quad + c_{11}\left[s^1(m-1,n-1) - s^2(m-1,n-1)\right] \\ &\quad + \left[u^1(m,n) - u^2(m,n)\right]. \end{aligned} \tag{3}$$

If the search length is less than one line, then the points $(m, n-1)$ and $(m-1, n-1)$ in (3) are fixed previously released data. Hence the error equation can be reduced to

$$\begin{aligned} e(m,n) &= c_{10}\left[s^1(m-1,n) - s^2(m-1,n)\right] \\ &\quad + u^1(m,n) - u^2(m,n) \\ &= c_{10}e(m-1,n) + \left[u^1(m,n) - u^2(m,n)\right]. \end{aligned} \tag{4}$$

This is clearly a relative ϵ-merging path generator for $|c_{10}| < 1$, since u^1 and u^2 would be selected as the same for purposes of merging.

In case the search length is longer than one line, then the global state, i.e., a whole line in this example, rather than just the local state has to be considered. The error function $e(m,n)$ in (3) is no longer monotone decreasing as in (4). However, if the encoding filter is two-dimensional asymptotically stable [15], we could estimate the peak of transient response due to initial conditions, and then select ϵ' such that the error in (3) is bounded by ϵ for the global initial conditions less than ϵ'. Hence the path generator still possesses the near merging property. Since one row of an ordinary image is fairly long (more than 250 pixels per line), it is observed that the paths merge before reaching the second search line for a reasonably large M in the (M, L) algorithm. The above analysis can also be extended to an encoding filter which has larger support than Fig. 2, and by considering the global state similar conclusions are obtained.

TABLE I
ESTIMATED VERTICAL AND HORIZONTAL CORRELATION COEFFICIENTS FOR TEST IMAGE

Vertical Correlation	Horizontal Correlation
0.973	0.977

TABLE II
OPTIMIZED PERFORMANCE* (dB) OF NONREGENERATIVE SEARCH CODING AT DIFFERENT PARAMETERS

M	Search Length L	No Path Separation	$THF = 0.05$	$THF = 0.1$	$THF = 0.2$
1	1,DPCM	15.7	—	—	—
2	5	17.6	17.9	18.0	18.0
	10	16.8	17.9	18.0	17.9
	40	16.4	17.8	17.9	17.9
	258	16.1	—	17.9	17.9
	515	15.8	—	17.9	17.9
4	5	18.4	—	18.8	18.8
	40	16.8	—	18.9	19.1
	258	16.3	—	18.7	19.1
	515	16.3	—	18.7	19.1
8	5	18.9	—	19.1	19.1
	40	17.6	—	19.1	19.2

*The optimum SNR is subject to 0.1 or 0.2 dB deviation.
—Not applicable or not simulated.

Examples

The simulations used a 256×256 pixel image having 8-bit gray scale. The image model was borrowed from [5], a 1×1 order quarter plane, separable model. The model coefficients were obtained by least square fit to the vertical and the horizontal correlation coefficients [5], which are listed in Table I. Two-level or 1-bit quantized inputs and square error distortion measure are used throughout this experiment. It can be expected that as M and L vary, the quantizer step size (output level spacing) should be adjusted accordingly. The step size in this experiment was selected to be optimal for the test image. Results are summarized in Table II. The SNR is the ratio of the coded image variance to the distortion. Conventional DPCM is equivalent to tree coding with a delay $L = 1$ and $M = 1$.

Fig. 3 is a plot of signal-to-noise ratio (SNR) versus the quantization step size for the case $M = 2$. The horizontal coordinate is normalized by the standard deviation of the image model input which was obtained from the model identification process. It can be inferred from these curves that the optimum quantization step size cannot be evaluated precisely. The reasons may be partly due to the nonlinearity of the encoder at low rates and partly to the inhomogeneity of the image. Consequently, a deviation of 0.1 or 0.2 dB on SNR is not significant in this kind of experiment.

If the coding algorithm is conducted without deletion of near-merged paths, the performance goes down as the delay length L increases. This degradation is clearly seen from the first column of Table II, and can also be observed in Fig. 3. It may be improved a little for small L by increasing M, but the performance decreases rapidly for large L. This comes from the fact that the retained M paths are near merged after long delays, and thus these M paths are effectively one path.

A simple and effective path separation criterion can be used to eliminate near merging for this example. We force the branches retained in the search to be $THF \cdot Q$ apart, where THF is the separation threshold factor and Q is the quantization step size.

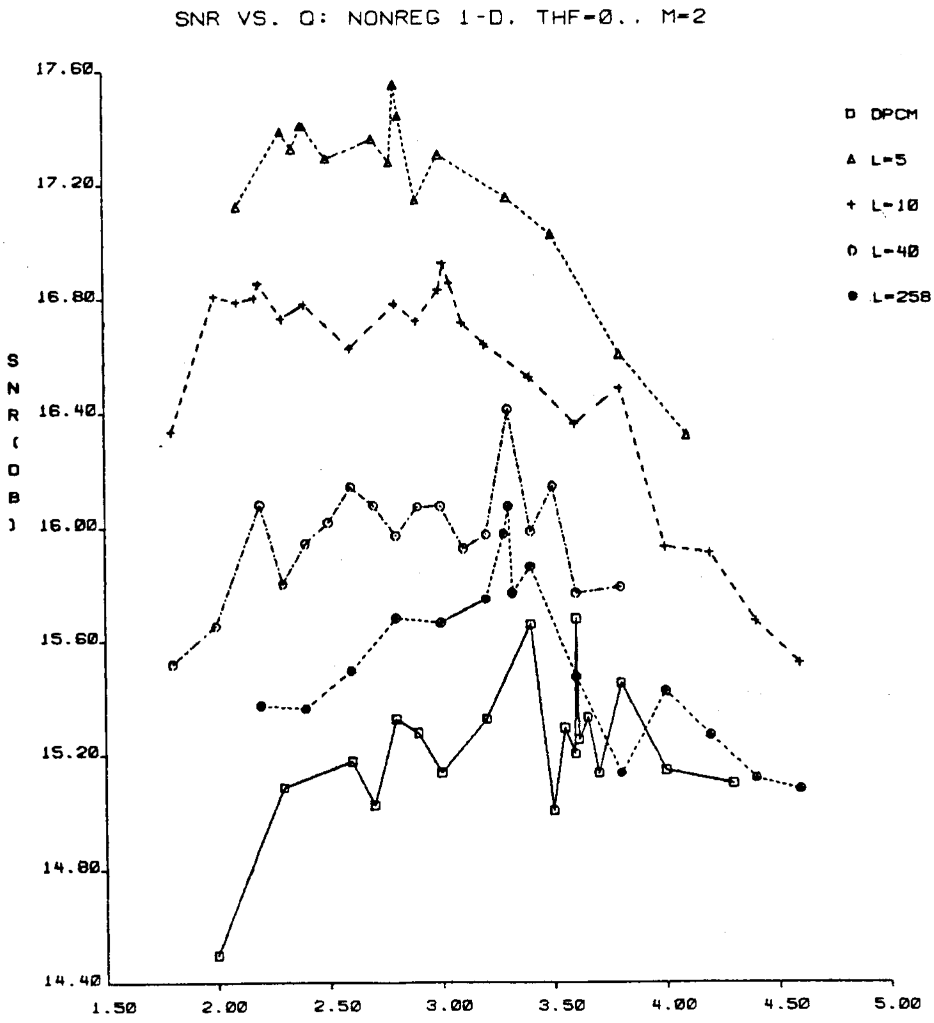

Fig. 3. SNR versus quantization step size.

Because the error (4) depends only on the most recent branch in the tree, two current nearby branches will lead to two near-merged paths in the future without referring to the past history. Therefore this criterion applies only on the current tree branches rather than the whole paths. If it is impossible to get M branches with distance $THF \cdot Q$ between them, this algorithm will select as much as it can and then pick the branches with the least accumulated distortions among the rest. Significant SNR improvements are obtained by using this separation technique as shown in Table II. It is interesting that the coding performance is quite insensitive to the THF value.

Little SNR improvement is obtained by increasing M from 4 to 8. As a consequence, we may suspect that the performance saturates around $M = 8$, $L = 40$, and $THF = 0.2$. The original test image and some reconstructed images are shown in Fig. 4. For easy comparison, parts of Fig. 4 are enlarged and displayed in Fig. 5. It can be observed from these pictures that tree coding can offer visible improvements over DPCM.

To compare with the nonregenerative (M, L) scheme, several simulations using the regenerative (M, L) search were tried for $M = 2$, which are listed in Table III. Increasing the delay L in the regenerative search does not change the results much, as we expect, since the near-merged paths give us essentially the same distortion. The path separation technique can still provide some SNR improvement.

Table IV shows the number of branches calculated for some selected simulations. These numbers represent the computational complexity of various schemes and parameters. The regenerative algorithm requires much more computation than the nonregenerative ones. The average and the variance of the search length for both regenerative and nonregenerative methods are shown in Table V. They effectively measure the required storage size. These values were calculated from the minimum length of the tree at which the released branch can be decided without ambiguity. It may be concluded from this table that the path separation reduces the maximum search length and hence allows smaller storage. This effect can be explained by the following. If the path separation is not included, the merged paths stemming from distinct branches at the released point would continue growing without end because the distortions of merged paths are comparable. Therefore, the decision on the released branch cannot be made until the search length L is reached. On the other hand, when the path separation criterion is used, some of merged paths are dropped which can result in an unambiguous decision at an earlier point in the search process.

V. Discussion

The near merging of paths in tree searching and one feasible solution to it are brought out in this correspondence. The effect of near merging on image coding and the improvement using path separation are illustrated by experiments. If an asymptotically stable ARMA path generator is used, the near merging phenomenon occurs in the (M, L) tree search for small M and

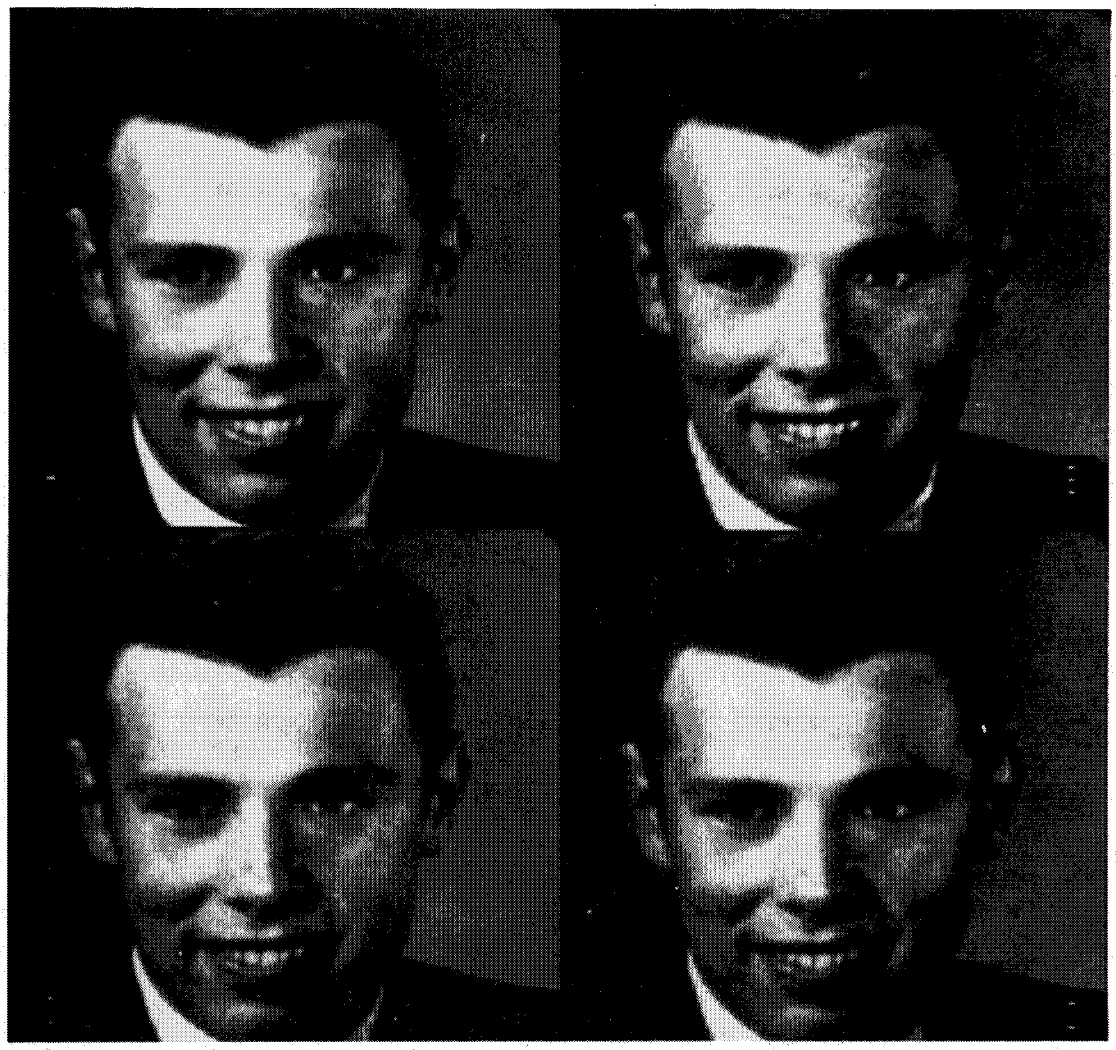

Fig. 4. Test images. (a) Original test image. (b) DPCM: SNR = 15.7 dB. (c) Tree coding: $M = 2$, $L = 10$, $THF = 0.1$, SNR = 18.0 dB. (d) Tree coding: $M = 8$, $L = 40$, $THF = 0.2$, SNR = 19.2 dB.

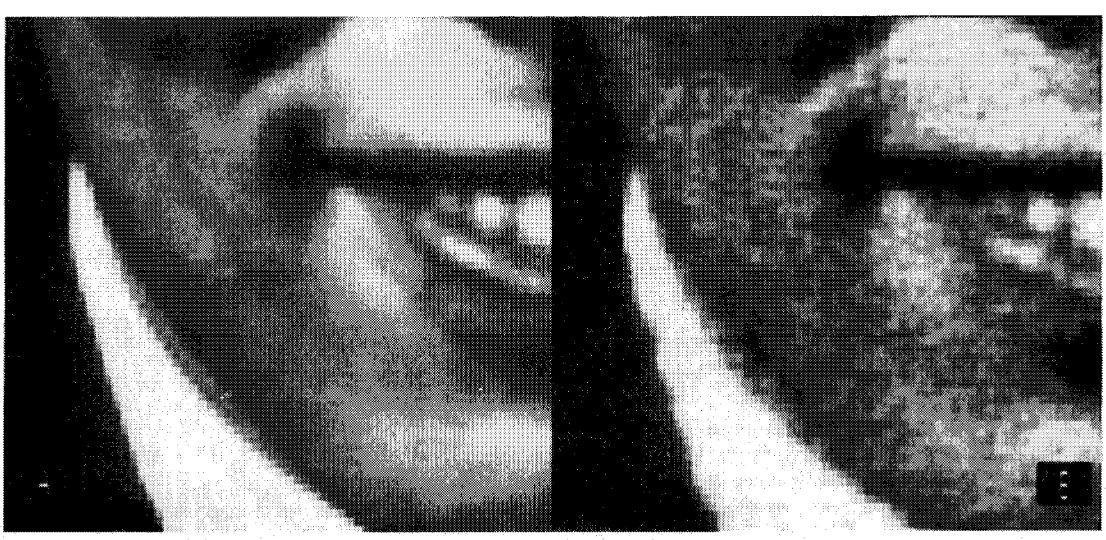

Fig. 5. Enlarged segments of images of Fig. 4.

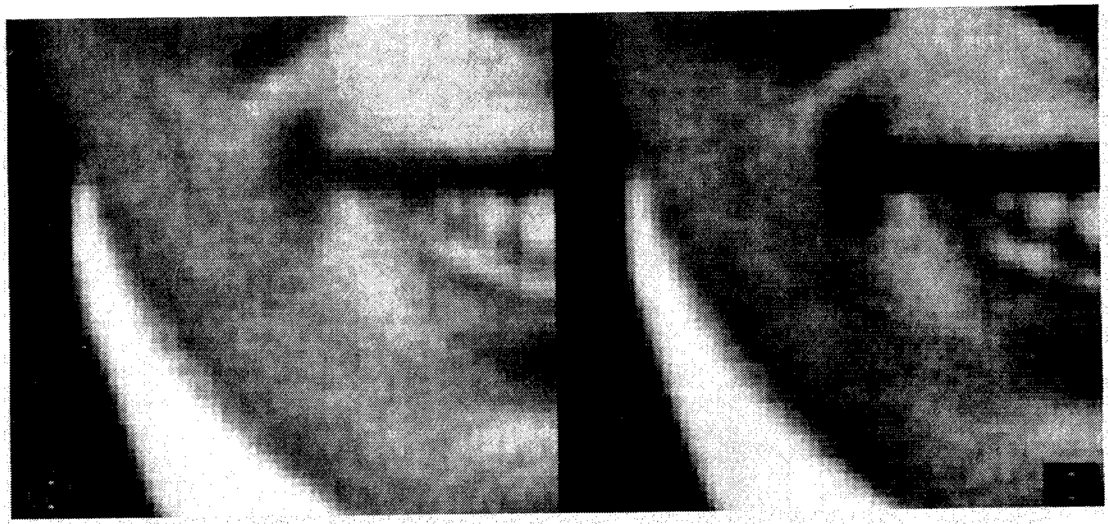

Fig. 5. (*Continued*)

TABLE III
OPTIMIZED PERFORMANCE* (dB) OF REGENERATIVE SEARCH CODING AT DIFFERENT PARAMETERS

M	Search Length L	No Path Separation	$THF = 0.1$
2	5	17.9	18.0
	10	17.8	17.9
	40	17.8	18.0

*The optimum SNR is subject to 0.1 or 0.2 dB deviation.

TABLE IV
NUMBER OF BRANCH COMPUTATIONS

Scheme	M	Search Length L	No Path Separation ($\times 10^5$)	$THF = 0.1$ ($\times 10^5$)
Nonregenerative	1	1,DPCM	1.31	—
	2	5	2.48	2.61
		40	2.61	2.62
		515	2.62	2.62
	4	5	4.60	5.14
		40	5.19	5.24
		515	5.24	5.24
	8	5	8.15	8.51
		40	10.31	10.43
Regenerative	2	5	6.18	4.57
		40	10.66	4.59

TABLE V
MEAN AND VARIANCE OF SEARCH LENGTH

Scheme	M	Search Length L	No Path Separation	$THF = 0.1$
Nonregenerative	1	1,DPCM	m: 1.0 v: 0.0	—
	2	5	m: 2.8 v: 1.5	m: 2.3 v: 0.3
		40	m: 14.1 v: 116.1	m: 2.3 v: 0.6
		515	m: 52.3 v: 5585.0	m: 2.3 v: 0.6
	4	5	m: 4.0 v: 0.9	m: 4.2 v: 0.3
		40	m: 23.0 v: 103.7	m: 7.1 v: 44.6
		515	m: 88.8 v: 7584.0	m: 7.1 v: 44.6
	8	5	m: 4.7 v: 0.3	m: 4.9 v: 0.1
		40	m: 28.2 v: 75.7	m: 20.2 v: 89.8
Regenerative	2	5	m: 2.9 v: 1.4	m: 2.2 v: 0.3
		40	m: 4.6 v: 43.2	m: 2.3 v: 0.3

—Not applicable.

large L. In the previous research on tree coding the value of M is comparable with L, therefore, the near merging effect did not appear. In the image coding case, one may want to use very long delay to encompass nearby image pixels in succeeding lines, and the merged paths due to this long search length can distort the results. Other suboptimal search schemes should have similar characteristics.

The above experiment was performed using a simple AR encoding filter and a suboptimal search. It does not seem helpful to use a search length greater than several pixels. In comparison to Modestino *et al.* [5], we did not use a smoothing filter nor an optimal rate-distortion derived filter. Although the analysis of the near merging of paths would be more difficult in these cases, we believe that our results would remain qualitatively the same.

REFERENCES

[1] F. Jelinek and J. B. Anderson, "Instrumentable tree encoding of information sources," *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 118–119, Jan. 1971.

[2] J. B. Anderson, "A stack algorithm for source coding with a fidelity criterion," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 211–226, Mar. 1974.

[3] J. B. Anderson and J. B. Bodie, "Tree encoding of speech," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 379–387, July 1975.

[4] J. B. Anderson and C. W. Law, "Real-number convolutional codes for speech-like quasi-stationary source," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 778–782, Nov. 1977.

[5] J. W. Modestino, V. Bhaskaran, and J. B. Anderson, "Tree encoding of images in the presence of channel errors," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 677–697, Nov. 1981.

[6] J. W. Modestino and V. Bhaskaran, "Robust two-dimensional tree encoding of images," *IEEE Trans. Commun.*, vol. COM-29, pp. 1786–1798, Dec. 1981.

[7] V. Bhaskaran, private communication, Nov. 1982.
[8] F. Jelinek, "Tree encoding of memoryless time-discrete sources with a fidelity criterion," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 584–590, Sep. 1969.
[9] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.
[10] J. B. Anderson and S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Trans. Commun.*, vol. COM-32, pp. 169–176, Feb. 1984.
[11] J. B. Anderson and F. Jelinek, "A 2-cycle algorithm for source coding with a fidelity criterion," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 77–92, Jan. 1973.
[12] A. J. Viterbi, "Convolutional codes and their performance on communication systems," *IEEE Trans. Commun. Tech.*, vol. COM-19, pp. 751–772, 1971.
[13] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.
[14] G. Bachman and L. Narici, *Functional Analysis*. New York: Academic, 1966.
[15] E. W. Kamen, "Asymptotic stability of linear shift-invariant two-dimensional digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-27, no. 12, pp. 1234–1240, Dec. 1980.

Comments on "On a Direct Method of Analysis and Synthesis of the SPRT"

GIOVANNI CORSINI, ENZO DALLE MESE, MEMBER, IEEE, GIOVANNI MARCHETTI, AND LUCIO VERRAZZANI

***Abstract*—In this brief communication, some comments on the above paper are made.[1] The efficiency of the method of analysis proposed by Vrana and other methods available in literature are compared.**

I. Introduction

We have read with great interest the above paper[1]; the argument treated by Vrana has a great practical relevance. Applications of the sequential probability ratio test (SPRT) are found for example, in digital communications and radar target detection. An exact analysis of the performance of the SPRT is very desirable. The method proposed by Vrana is based on the recursive calculation of the risk function of a sequential *a posteriori* probability test (SAPT), which is equivalent to the SPRT to be analyzed. The performance measures considered in Vrana's paper[1] are the probabilities of both types of errors, α_0, α_1, and the average sample numbers (ASN), E_0 and E_1, conditioned respectively on the two alternative hypotheses, H_0 and H_1. Some general remarks are

a) Other parameters, in addition to those calculated by Vrana, are important in order to fully specify the test characteristics. They are the operating characteristic function (OCF) and the moments of the random variable "length of the test" N. Moreover, the knowledge of the probability density function (pdf) of the test statistic Z_k at stage k (usually equal to the logarithm of the likelihood ratio) is required for specific application.

b) Vrana's ingenious method, described in his paper,[1] yields a considerable simplification of calculation only if his eqs (A13) and (A14) hold. Unfortunately, these conditions are not fulfilled, in general, as occurs, for example, in incoherent radar target detection.

c) Two more methods for the exact calculation of the SPRT performance are available, we recall them briefly for comparison purposes.

II. Computation of SPRT Performance

Let Z_k denote the logarithm of the likelihood ratio (test statistic) and $z_k = Z_k - Z_{k-1}$, the statistic increment at stage k. In the case of an independent identically distributed (i.i.d.) observations $(x_1, x_2, \cdots, x_n, \cdots)$, we have $z_k = \log[f(x_k|H_1)/f(x_k|H_0)]$, where $f(x_k|H_i)$, $i = 0,1$, is the pdf of the kth observation, and $p_z(\cdot|H_i)$, $i = 1,0$, (the pdf of z_k) does not depend on k.

The first method for the exact calculation of the SPRT performance is due to Albert [1], who considered a SPRT more general than Wald's test. The relevant equations are obtained through the following simple observation: a SPRT that starts at point w and ends at stage n is equivalent to a SPRT that starts at point v, where $(v - w)$ is the increment of the test statistic at stage 1, and ends at stage $(n-1)$. So, if the starting point of the SPRT is w, we have[2]

$$\alpha_0(w) = \int_A^{\infty} p_z(v - w|H_0)\,dv + \int_B^A \alpha_0(v)\,p_z(v - w|H_0)\,dv, \tag{1}$$

$$\alpha_1(w) = \int_{-\infty}^{B} p_z(v - w|H_1)\,dv + \int_B^A \alpha_1(v)\,p_z(v - w|H_1)\,dv, \tag{2}$$

$$M_1(w|H_i) = 1 + \int_B^A M_1(v|H_i)\,p_z(v - w|H_i)\,dv, \tag{3}$$

and

$$M_k(w|H_i) = 1 + \sum_{s=1}^{k-1}\binom{k}{s}\int_B^A M_s(v|H_i)\,p_z(v - w|H_i)\,dv + \int_B^A M_k(v|H_i)\,p_z(v - w|H_i)\,dv, \tag{4}$$

for $i = 1,0$, $k = 2,3,\cdots$; $B < w < A$, and where $M_j(w|H_i)$ is the conditioned jth moment of the random variable N. The parameters of interest are obtained from (1), (4) by setting $w = 0$. Eqs. (1), (4) are Fredholm integral equations of the second kind and their numerical solution does not present any difficulty. Furthermore, they are quite general, and do not depend on any specified model of $p_z(\cdot|H_i)$. This method has been applied successfully to evaluate the performance measures of the SPRT used in radar target detection [2], [6] and in communication systems [5].

The second method was suggested by Rozanov [4], who developed a recursive method to calculate the pdf of Z_k at each stage k. It is obvious that this knowledge enables us to compute all the parameters of interest of the SPRT. By observing that $Z_k = Z_{k-1} + z_k$, the pdf of Z_k is obtained through the convolution integral of the pdf's of Z_{k-1}, and z_k. The main problem arising in the application of Rozanov's method is that the pdf of Z_k for all the k is required in order to obtain the performance measures of the SPRT; however, this problem can be overcome by properly choosing the numerical integration procedure and the truncation step. Also, this method is quite general.

Manuscript received March 28, 1983; revised September 16, 1983. This work was supported in part by the Italian National Research Council.

G. Corsini and E. Dalle Mese are with the Institute of Electronics and Telecommunications, University of Pisa, 56100 Pisa, Italy.
G. Marchetti and L. Verrazzani are with the Institute of Electronics and Telecommunications and with the CSMDR of the Italian National Research Council, University of Pisa, 56100 Pisa, Italy.

[1] I. Vrana, *IEEE Trans. on Informat. Theory*, vol. IT-28, no. 6, pp. 905–911, 1982.

[2] The thresholds A and B are the logarithm of the thresholds considered by Vrana. In addition, we put $A > B$ according to the most common notation.

0018-9448/84/0500-0573$01.00