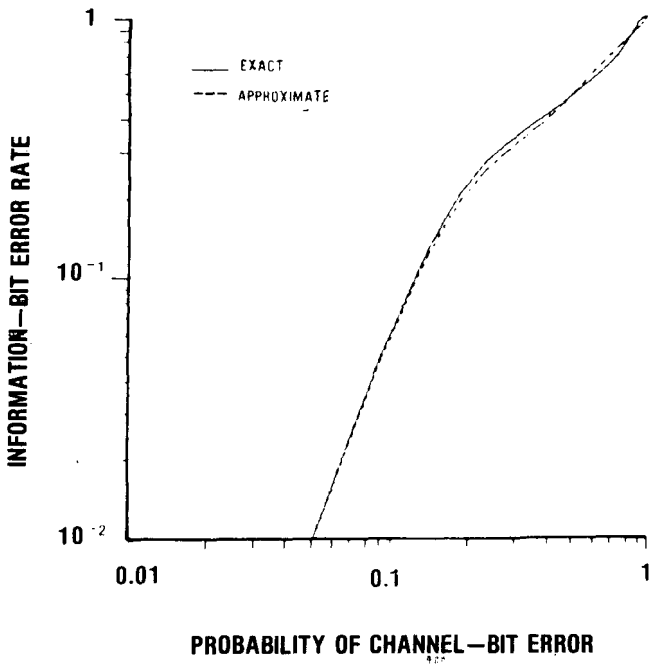

Fig. 1. Information-bit error rate as a function of channel-bit error probability for Golay (23, 12) code.

For repetition codes and hard decisions, $k = 1$, $d = n$, and n is odd, so (19) reduces to

$$P_{ib} = \sum_{i=(n+1)/2}^{n} \binom{n}{i} P_b{}^i (1 - P_b)^{n-i}, \tag{20}$$

which is the correct, exact formula in this case. Another demonstration of the accuracy of (19) is provided by examination of the Golay (23, 12) code, for which $d = 7$ and $t = 3$. The information-bit error rate can be precisely calculated from the data provided by Patterson [5]. The exact error rate always exceeds the lower bound of inequality (12). A numerical comparison of the exact error rate and the approximation of (19) indicates that the approximation produces an error of approximately 7 percent or less. As P_b decreases below 0.25, the exact error rate monotonically converges toward the approximation. Fig. 1 illustrates the comparison for $P_{ib} \geq 10^{-2}$.

In nonbinary communications, an information symbol represents m information bits. It is assumed that an incorrectly decoded information symbol is equally likely to be any of the remaining $2^m - 1$ symbols in the alphabet. Among those symbols, a given bit is incorrect in 2^{m-1} cases. Thus,

$$P_{ib} = \frac{2^{m-1}}{2^m - 1} P_{is}. \tag{21}$$

For Reed–Solomon codes, $n = 2^m - 1$. Equations (18) and (21) imply that

$$P_{ib} \approx \frac{n+1}{2n^2} \left[d \sum_{i=t+1}^{d} \binom{n}{i} P_s{}^i (1 - P_s)^{n-i} + \sum_{i=d+1}^{n} i \binom{n}{i} P_s{}^i (1 - P_s)^{n-i} \right] \tag{22}$$

for hard-decision decoding.

Approximations other than (18), (19), or (22) have often been used in the communications literature. A common approximation is

$$P_{is} \approx \frac{1}{n} \sum_{i=t+1}^{n} i \binom{n}{i} P_s{}^i (1 - P_s)^{n-i} \tag{23}$$

(for example, see [6]). This approximation satisfies neither inequality (12) nor (20) in the corresponding cases. For the Golay (23, 12) code, it produces an estimate of $P_{ib} = P_{is}$ that is less than six-tenths of the correct value for small values of $P_b = P_s$.

References

[1] G. C. Clark and J. B. Cain, *Error-Correction Coding for Digital Communications*. New York: Plenum, 1981.
[2] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.
[3] R. J. McEliece, *The Theory of Information and Coding*. Reading, MA: Addison-Wesley, 1977.
[4] W. W. Peterson and E. J. Weldon, *Error-Correcting Codes*, 2nd ed. Cambridge, MA: M.I.T. Press, 1972.
[5] P. L. Patterson, "New performance analyses for the Golay and shortened Golay codes," presented at GLOBECOM'82, paper E7.5.
[6] E. A. Geraniotis, "Coding for BPSK SFH/SSMA communications over fading channels," presented at IEEE MILCOM'82, paper 35.1.

A Low Complexity Stack Decoder for a Class of Binary Rate $(n-1)/n$ Convolutional Codes

A. J. VINCK

Abstract—**In this paper we discuss the implementation of a modified stack decoder for a class of binary rate $R = (n-1)/n$ convolutional codes used on a binary symmetric channel (BSC). For large values of n, the classical implementation of the stack decoder quickly becomes impractical, as each extension of an information sequence estimate gives rise to $2^{(n-1)}$ successor estimates. A Fano type of sequential decoder is then preferable. However, by using the structure of a class of systematic rate $(n-1)/n$ codes, with optimum distance profile (ODP), we are able to modify the classical stack decoder such that it is of comparable complexity. The average number of stack reorganizations, as well as the average number of successors per extension, can be reduced considerably, without increase of decoding error probability.**

Introduction

The stack decoding algorithm is a sequential decoding algorithm for convolutional codes. It searches through the nodes of the code tree in an efficient way. Each node examined represents an information sequence estimate through part of

Paper approved by the Editor for Data Communication Systems of the IEEE Communications Society for publication after presentation at the International Symposium on Information Theory, St. Jovite, P.Q., Canada, September 1983. Manuscript received October 4, 1982; revised July 15, 1983.

The author is with the Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands.

0090-6778/84/0400-0476$01.00 © 1984 IEEE

the tree. In the stack decoder, these information sequence estimates together with their respective Fano metrics are stored in a so-called ordered stack [1], [2]. The position in the ordering depends on the Fano metric.

Whatever the organization structure or size of the stack is, decoding complexity is determined by two basic steps.

1) The decoder has to find the best estimate from the stack.

2) The best estimate is removed from the stack and, for binary rate $R = k/n$ codes, extended to 2^k successor estimates to be stored in the stack.

Of course, organization and size do influence the complexity of both steps. A stack reorganization is defined to be the event where one or more successors are to be stored in the stack.

In the next section, we first describe a modification of the second basic step for a class of binary rate $R = (n-1)/n$ codes. If received frames of L n-tuples are to be decoded, then the classical stack decoder stores roughly $L\ 2^{(n-1)}$ successor estimates. Instead of storing $2^{(n-1)}$ successors per extended estimate, we store only a very small fraction of this number. Then we give a second modification such that the average number of stack reorganizations per information $(n-1)$-tuple depends on the BSC error probability p.

We conclude with a comparison between the complexity of the modified stack decoder and the Fano decoding algorithm [3].

Modifications

We first describe the modifications of an extension for rate $R = (n-1)/n$ codes. We assume that a rate $R = (n-1)/n$ encoder G has zero delay, and a matrix of lowest order coefficients of rank $(n-1)$. As an example one can take the class of systematic codes.

Per definition, the stack decoder always extends the best estimate, which is said to be the top node of the stack. The $2^{(n-1)}$ possible successor estimates that follow from this extension will be called "siblings," as they have the same father node. Let $\boldsymbol{I} * i$ be one of the successors to the information sequence estimate $\boldsymbol{I}$. The Fano metric $L_f(\boldsymbol{I} * i)$ depends on the probability $p(\hat{\boldsymbol{n}})$ of the error n-tuple estimate $\hat{\boldsymbol{n}}$ that corresponds to the extension from $\boldsymbol{I}$ to $\boldsymbol{I} * i$. For $R = (n-1)/n$ codes, this metric is

$$L_f(\boldsymbol{I} * i) \triangleq L_f(\boldsymbol{I}) + \log_2 p(\hat{\boldsymbol{n}}) + 1.$$

For a BSC, the probability $p(\hat{\boldsymbol{n}})$ equals $(1-p)^{n-W_H}p^{W_H}$, where W_H is the Hamming weight of the error n-tuple estimate.

Note that a sibling of $\boldsymbol{I} * i$ with smaller metric can appear at the top of the stack, only if $\boldsymbol{I} * i$ has already been there before. This observation justifies the following modification of the extension of the top node.

Step 1: Remove the top node from the stack.

Step 2: Generate the complete set of siblings of this prior top node having the same metric as this prior top node and place the best "son" of each in the stack as well as the best son of the prior top node.

Step 3: Generate (if it exists) a "next best" sibling (i.e., a sibling whose metric is maximum among those siblings with smaller metrics) of the prior top node and place this sibling in the stack.

The nodes in the stack can now be seen as representatives for the siblings with the same or smaller metric. In the classical stack decoder the top node is removed from the stack and all its sons are placed in the stack.

The above modification is applicable to all rate $R = k/n$ codes. The complexity of an extension then depends on how difficult it is to find the siblings with the same Fano metric, and a best one with smaller metric. We therefore illustrate the above modification for the class of systematic $(n-1)/n$ optimum distance profile (ODP) codes as given by Hagenauer [4]. For these codes, the entries in the nth column of the zero-order generator matrix all have a nonzero constant term. As observed by Hagenauer, there are only two different sets of weights associated with the $2^{(n-1)}$ successors of an estimate, depending on the parity bit value of the all zero extension. Hence, if we extend an information sequence estimate to $2^{(n-1)}$ successors, then the corresponding error n-tuple estimates are either all of even or all of odd Hamming weight. As there are exactly $2^{(n-1)}$ different n-tuples with the same type of weight, we also know exactly how many error estimates there are of a certain Hamming weight. Let $W(t)$ denote the Hamming weight of the error n-tuple estimate at time interval t. Then, the application of the above modification for the decoder located at time interval t, $t \geqslant 1$, can be described as follows.

Step 1: Remove the top node from the stack.

Step 2: Determine the Hamming weight $W(t-1)$ of the error n-tuple corresponding to the last branch to this node and generate all $\binom{n}{W(t-1)}$ siblings that have the same metric [same $W(t-1)$]. Place the best son of each in the stack as well as the best son of the prior top node.

Step 3: Generate (if it exists) a next bext sibling of the prior top node with an error n-tuple estimate of Hamming weight $W(t-1)+2$ and place this sibling in the stack.

Note that for systematic ODP codes, the sons in step 2 all have an associated error n-tuple estimate of Hamming weight $W(t) = 0$ or 1.

For systematic ODP codes [4], or any code with an equivalent zero-order matrix, we can further simplify the algorithm as follows. Let the top node represent an information sequence of length t, $t > 1$.

Step 1: Take the top node from stack.

Step 2: Determine the Hamming weight $W(t-1)$, and generate the $\binom{n}{W(t-1)}$ siblings with the same metric. Place the best son of each in the stack as well as the best son of the prior top node.

Step 3: If $W(t-1) \neq 0$, generate (if it exists) the next best sibling of the prior top node with an error n-tuple estimate of Hamming weight $W(t-1)+2$ and place this sibling in the stack.

Step 4: If $W(t-1) = 1$ or 2, and $W(t-2) = 0$, generate the father node of the prior top node. Generate a sibling of this father node with $W(t-2) = 2$, and place this sibling in the stack.

The correctness of this modification is proven as follows. Let $\boldsymbol{I} * i$ and $\boldsymbol{I} * i'$ be two siblings with last branch error n-tuple estimates with weight $W(t-2) = 0$ and $W(t-2) = 2$, respectively. Then node $\boldsymbol{I} * i'$ can only reach the top of the stack after the son of $\boldsymbol{I} * i$ with $W(t-1) = 1$ or 2 has been there, as both latter nodes have a higher Fano metric. But, if one of these two sons is at the top of the stack, then $\boldsymbol{I} * i'$ is stored at step 4 in the algorithm.

In Fig. 1, we illustrate in part the modifications for a rate $R = 2/3$ code taken from [4]. We assume that an isolated error occurs at time interval $(t-1)$. We have shown the Fano metric of the nodes at time interval $(t-2)$, $(t-1)$, and t. Along the branches are the noise estimates that correspond with the information sequence estimates.

If, for our modified decoder, steps 1–4 give rise to only one successor with weight $W(t) = 0$, then no estimate need be stored in the stack, and hence, no stack reorganization takes place, as this successor is known to have a higher Fano metric than all estimates present in the stack.

From Fig. 1, one can see that, in the isolated error case, four additional stack entries are needed in two stack reorganizations. In general, if only one single error occurs, we store $(n+1)$ estimates in two stack reorganizations. Hence, the average number of stack entries and the average number of

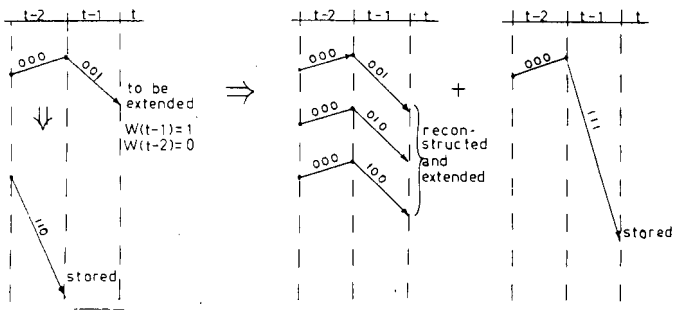

Fig. 1. Illustration of the modification for a specific single error at time interval $t-1$, and $R = 2/3$. The case where $W(t-1) = 1$ and $W(t-2) = 0$.

stack reorganizations per information $(n-1)$-tuple are proportional to $(n+1) \times n \times p$ and $2 \times n \times p$, respectively, for sufficiently small p. For instance, for a rate $R = 2/3$ code and $p = 0.0172$ (R equals R_{comp}), $(n+1) \times n \times p = 0.2$. The classical stack decoder needs $(2^{(n-1)} - 1)$ additional stack entries per extended estimate, even in the error free case. For $n = 8$ and $p = 0.002$ (R equals R_{comp}), $(n+1) \times n \times p = 0.1$, whereas $(2^{(n-1)} - 1) = 127$.

If a node is extended with an information $(n-1)$-tuple equal to the first $(n-1)$ digits of a received n-tuple, then the nth code digit determines whether the extension has weight $W(t) = 0$ or 1.

From the above, it follows that the complexity of an extension is very low if we use the systematic ODP codes. In the next section we discuss simulation results for the modified, the classical stack decoder, and the Fano decoder.

Simulations

We have simulated the modified as well as the classical stack decoder for a constraint length 15 rate $R = 2/3$ ODP code from [4] for several channel error probabilities. The most important one is $p = 0.0172$, when R equals R_{comp}. Before encoding, the two information sequences are divided into frames of 241 digits followed by 15 all-zero digits. Each run consists of 10 000 frames. We measured the distribution of the number of stack reorganization, C_r, per information pair; see Fig. 2. For $p = 0.0172$, 0.0086, and 0.0043, the classical algorithm gives an average $\bar{C}_r = 1.5$, 1.15, and 1.1, respectively, whereas the modified algorithm gives $\bar{C}_r = 0.26$, 0.08, and 0.03, respectively. Hence, for the modified decoder, $\bar{C}_r$ decreases almost linearly with p as can be expected for low channel error probabilities.

The average number of stack items per information pair $\bar{C}_s$, as well as its distribution (see Fig. 3), are important parameters for determining the probability of stack overflow. For $p = 0.0172$, 0.0086, and 0.0043 the classical decoder gives $\bar{C}_s = 3.5$, 3.1, and 3.0, respectively, whereas the modified decoder gives $\bar{C}_s = 0.63$, 0.20, and 0.09, respectively. For the modified decoder, $\bar{C}_s$ decreases again linearly with p.

From the above, and Figs. 2 and 3, it follows that the average complexity of the modified decoder decreases linearly with p. The described modifications give rise to a low complexity stack decoder for high-rate systematic ODP codes. In the next paragraph, we compare our modified decoder to a Fano decoder.

The Fano decoder is an alternative for decoding convolutional codes sequentially. It uses the Fano metric of only one estimate and a threshold. The Fano decoder moves forward or backward from one node to an adjacent node. The decoder is allowed to move only if the metric of the new node is above

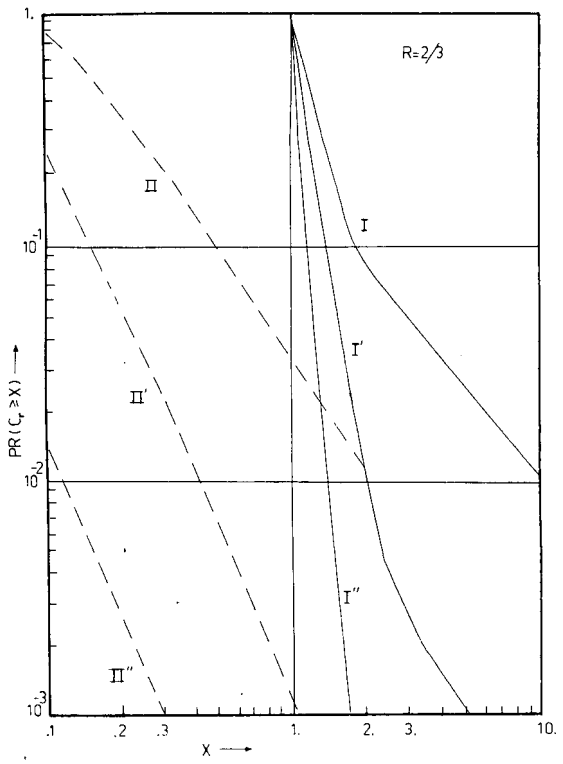

Fig. 2. Distribution of the number of stack reorganizations C_r per information pair for the classical (I, I', I'') and the modified decoder (II, II', II'') for $p = 0.0172$, 0.0086, and 0.0043, respectively.

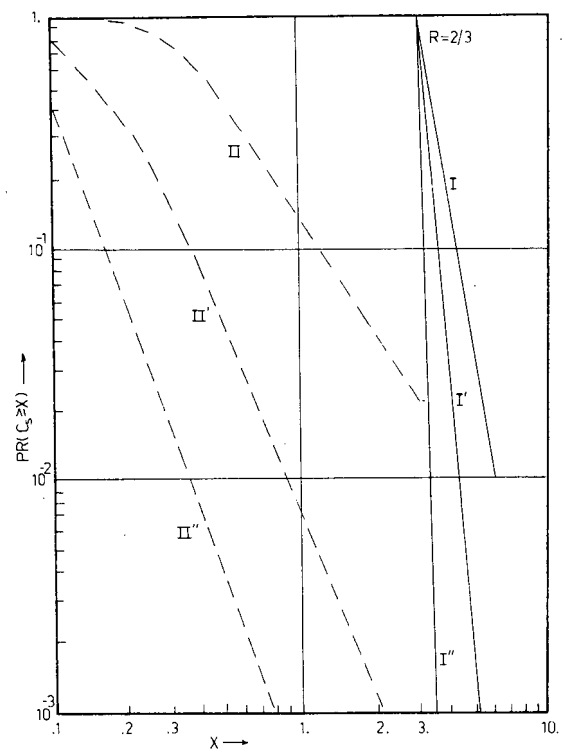

Fig. 3. Distribution of the number of stack items C_s per information pair for the classical (I, I', I'') and the modified decoder (II, II', II'') for $p = 0.0172$, 0.0086, and 0.0043, respectively.

a threshold. If this is not possible, then the threshold is lowered. Backward moves are considered to be of low complexity. After a backward move, the decoder tries to move forward to the next best adjacent node. After a forward move, or a threshold lowering, the decoder looks forward to the best adjacent node. This forward looking is usually taken as a unit of computation. The Fano algorithm is organized such that it never moves to the same node with the same threshold more than once. This means that the decoder may visit the same node repeatedly with different threshold values. In the stack decoder the same extension is never done more than once.

In the noiseless case, the Fano decoder always moves forward with $W(t) = 0$, and tries to tighten the threshold. In this case, the decoding complexity is the same as for the modified stack decoder. If only one single isolated error occurs, then the Fano decoder lowers its threshold such that a search among all $\binom{n}{1}$ possible forward moves with $W(t) = 1$ is permitted, in order to find the correct one. In this case the modified stack decoder basically also only investigates the same number of estimates.

In the simulations for the Fano decoder, the threshold step size Δ was chosen as the negative metric contribution for one single error per n-tuple. For this value of Δ, the decoder has to do trace-back work in order to lower the threshold such that the single error nodes can be explored. The amount of trace-back work increases if Δ increases, and hence, p decreases.

In the simulations we took the constraint length 15 $R = 2/3$ ODP code from [4]. We compared the number of forward looks for the Fano decoder with the number of successors for our modified decoder per information pair. The average of both numbers is denoted as $\bar{C}_f$. Thus, for $p = 0.0$, the decoders are of the same complexity. For $p = 0.0172$, 0.0086, and 0.0043, the Fano decoder gives $\bar{C}_f \doteq 5.5$, 1.8, and 1.3, respectively. The modified decoder gives $\bar{C}_f = 1.6$, 1.2, and 1.1, respectively. Hence, for $p = 0.0172$ our modified decoder clearly outperforms the Fano decoder. A similar conclusion in favor of the stack decoder when $R \approx R_{\text{comp}}$ can be found in Geist [5]. The distributions that correspond with $\bar{C}_f$ are given in Fig. 4.

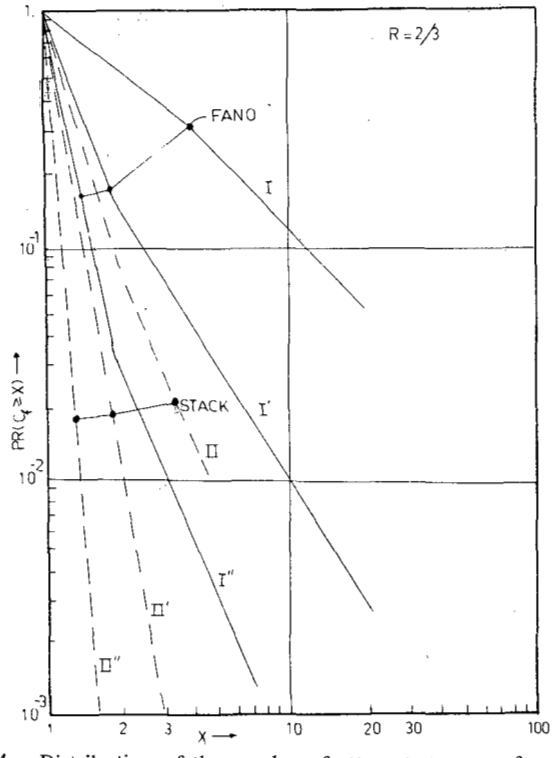

Fig. 4. Distribution of the number of attempts to move forward for the Fano decoder (I, I', I") and the number of successor estimates for the modified stack decoder (II, II', II") for $p = 0.0172$, 0.0086, and 0.0043, respectively.

Conclusions

We developed a modified stack decoder for a class of systematic $R = (n-1)/n$ ODP codes. If the number of successors in the modified stack decoder is compared to the number of attempts to move forward in the Fano decoder, then the complexity of the modified decoder is comparable to the complexity of the Fano decoder for low channel error probabilities. For $R = 2/3$ and $p = 0.0172$ (R equals R_{comp}), our modified decoder is superior to the Fano decoder. Simulation results indicate that for high rate codes our modifications enable a low complexity stack decoder.

Acknowledgment

The author would like to express his thanks to one of the referees for many helpful comments and suggestions.

References

[1] K. Zigangirov, "Some sequential decoding procedures," *Probl. Peredach. Inform.*, vol. 2, no. 4, pp. 13–15, 1966.

[2] F. Jelinek, "A fast sequential decoding algorithm using a stack," *IBM J. Res. Develop.*, vol. 13, pp. 675–685, Nov. 1969.

[3] R. M. Fano, "A heuristic discussion of probabilistic decoding," *IEEE Trans. Inform. Theory*, vol. IT-9, pp. 64–74, Apr. 1963.

[4] J. Hagenauer, "High rate convolutional codes with good distance profiles," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 615–618, Sept. 1977.

[5] J. M. Geist, "An empirical comparison of two sequential decoding algorithms," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 415–419, Aug. 1971.

Image Decimation and Interpolation Techniques Based on Frequency Domain Analysis

T. C. CHEN AND RUI J. P. DE FIGUEIREDO

***Abstract*—A scheme for a spatial domain image data preprocessing decimation and postprocessing interpolation is presented. The scheme is implemented by appropriate FIR digital filters. Frequency and spatial domain specifications are discussed in the design of the corresponding digital filters. Fast approximation techniques in the spatial domain are presented.**

Paper approved by the Editor for Signal Processing and Communication Electronics of the IEEE Communications Society for publication without oral presentation. Manuscript received July 6, 1982; revised January 18, 1983.

T. C. Chen was with the Department of Electrical Engineering, Rice University, Houston, TX 77251. He is now with Bell Communications Research, Holmdel, NJ 07733.

R. J. P. de Figueiredo is with the Departments of Electrical Engineering and Mathematical Sciences, Rice University, Houston, TX 77251.

0090-6778/84/0400-0479$01.00