THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 61, No. 5, May–June 1982
Printed in U.S.A.

Adaptive Intra-Interframe DPCM Coder

By P. PIRSCH

(Manuscript received August 4, 1981)

Adaptive prediction schemes provide lower transmission rates than those obtained by simple previous frame prediction. In this paper, we measure the entropy of prediction errors for two types of adaptive intra-interframe prediction algorithms. In the first case, that predictor which results in the least prediction error for previously transmitted neighboring pels is selected from a set of predictor functions. In the second case, prediction is a weighted sum of previous frame and intraframe predictions, where the weights are changed from pel to pel by gradient techniques. We also investigate various modifications of the basic methods. Further, a new type of variable length encoding in which the locations of the nonzero prediction errors are coded by horizontal run lengths is discussed. Compared with the pel entropy of previous frame prediction, the run length coding gives a gain of 2 to 16 percent, depending on the scene. Compared to simple previous frame prediction the first type of adaptive scheme in combination with horizontal run length coding provides a gain in entropy of 18 to 29 percent, whereas the second type of adaptive scheme provides a gain of 20 to 32 percent.

I. INTRODUCTION

The bit rate required for digital transmission of television pictures can be significantly reduced by interframe DPCM encoding. The coding method which has been widely proposed for video-telephone and video-conference application is conditional replenishment.[1,2] In conditional replenishment, each frame of a television sequence is segmented into changed and unchanged areas. Various methods can be used for encoding the changed parts of a frame. Intraframe predictive coding is very efficient for these parts.[3,4] In conditional replenishment, no information about the unchanged areas is transmitted. At the receiver, the unchanged areas are reconstructed by repeating from the previous frame. However, it is necessary to transmit address information that

indicates the location of the changed areas. Several modifications and improvements of the basic method of conditional replenishment have been made. Most by them are described in a survey by Haskell.[5]

This paper describes adaptive intra-interframe prediction. It is obvious that stationary background of a frame is best predicted from a pel in the previous frame which has the same position as the pel to be predicted, whereas parts of a frame with moving objects are better predicted by an intraframe predictor. Therefore, a prediction scheme which provides automatic switching between the two types of predictors, depending upon the part of the picture, will result in lower bit rates. To avoid the transmission of additional predictor control information, the adaptive prediction schemes described here are based on previously transmitted reconstructed pels. Further, no forward segmenter like that of conditional replenishment is used. Therefore, only the quantized prediction error has to be coded and transmitted.

A block diagram of such a DPCM encoder with adaptive prediction is shown in Fig. 1. The investigations in this paper concern a comparison of the performance of two types of adaptive predictors. The first one, denoted by predictor selection, is a scheme where one predictor is selected from a set of predictors. In the second scheme, the predictor is a weighted sum of predictors and the prediction coefficients are changed continuously by a gradient algorithm. As a measure of predictor performance, the entropy of the quantized prediction error is used. For three different television scenes an estimate of the entropy is obtained from DPCM simulations. The necessary measures against buffer overflow and underflow, in case of variable length encoding, have not been considered here.

This paper is organized as follows. Section II gives a detailed description of the two basic algorithms and their modifications. Section III describes a variable length encoding scheme which is especially suited for DPCM coders that have improved prediction. Results of simulations on real scenes are given in Section IV.

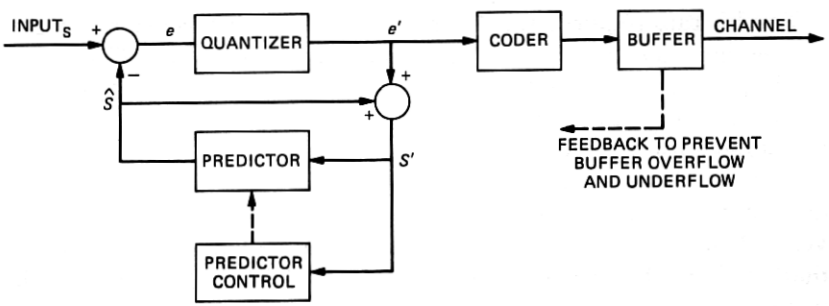

Fig. 1—Block diagram of a DPCM coder with adaptive predictor.

II. DESCRIPTION OF THE PREDICTION ALGORITHMS

Let f_i be one of M predictor functions. If each f_i is a linear predictor function, then

$$f_i = \sum_{j=1}^{N} a_{ij} s'_j, \tag{1}$$

where a_{ij} are the weighting coefficients and s'_j are previously transmitted pels. The prime in s'_j indicates that these are reconstructed pels which are known at the receiver. The subscripting for pels neighboring the present pel s_0 is shown in Fig. 2. The predictor functions f_i, $i = 1, 2, \cdots M$, are linear combinations of N pels, s'_j, $j = 1, 2, \cdots N$, which form a vector

$$\mathbf{s}' = \begin{vmatrix} s'_1 \\ s'_2 \\ \cdot \\ \cdot \\ s'_N \end{vmatrix}. \tag{2}$$

In vector notation, equation (1) can be written as

$$f_i = \mathbf{a}_i^T \mathbf{s}'. \tag{3}$$

Here the superscript T denotes the transpose of a vector or matrix, and $\mathbf{a}_i$ is the vector formed by the coefficients a_{ij}, $j = 1, 2, \cdots N$. The prediction value $\hat{s}_0$ is a weighted sum of all predictor functions,

$$\hat{s}_0 = \sum_{i=1}^{M} b_i f_i. \tag{4}$$

If $\mathbf{f}$ denotes the vector of elements f_i, $i = 1, 2, \cdots M$, and $\mathbf{b}$ denotes the vector of elements b_i, $i = 1, 2, \cdots M$, then

$$\hat{s}_0 = \mathbf{b}^T \mathbf{f}. \tag{5}$$

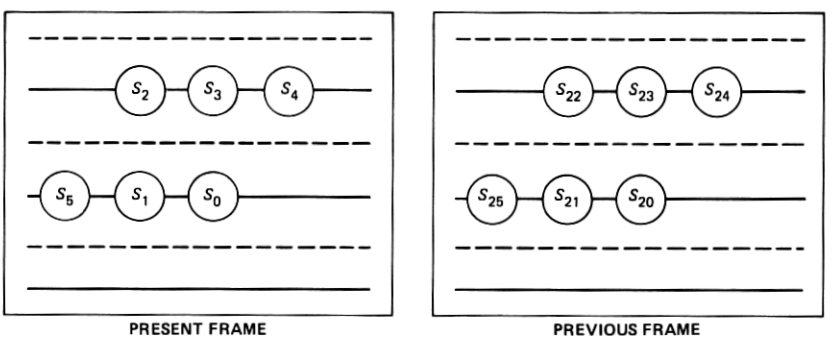

Fig. 2—Configuration and subscripting of picture elements. Pel s_0 is the present pel to be predicted. Dotted lines denote scan lines from previous fields.

This description is general and includes switched prediction by allowing special values of $\mathbf{b}$, such as $b_k = 1$ and $b_i = 0$ for all $i \neq k$. Combining (3) and (5) it follows that

$$\mathbf{f} = \mathbf{A}\mathbf{s}' \tag{6}$$

$$\hat{s}_0 = \mathbf{b}^T\mathbf{A}\mathbf{s}', \tag{7}$$

where

$$\mathbf{A} = \begin{vmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_M^T \end{vmatrix} \tag{8}$$

is an $M \times N$ matrix. The set of predictor functions is described by the matrix $\mathbf{A}$, with the coefficient vectors $\mathbf{a}_i$ chosen such that a particular predictor function provides a good prediction for a specific area of a television scene, like stationary background, moving objects, etc. The algorithm then seeks to automatically adapt the vector $\mathbf{b}$ to various areas of a scene so as to minimize the prediction error.

In this investigation, the set of predictor functions is restricted to a previous frame predictor

$$f_1 = s'_{20} \tag{9}$$

and an intraframe predictor

$$f_2 = a_{21}s'_1 + a_{22}s'_2 + a_{23}s'_3. \tag{10}$$

The following prediction algorithms are described for two predictor functions, but they can easily extend to more than two.

2.1 Predictor selection schemes

From a set of predictor functions, the predictor which results in the least prediction error for previously transmitted neighboring pels is selected as the predictor for the present pel. For each predictor function, a decision function u_i is defined, which is the sum of the amount of the prediction errors for each pel in a small window of neighboring pels. The predictor which has the smallest value for the decision function is chosen as predictor. This criterion was also used by Stuller et al.[6] for gain and displacement compensation. The basic selection rule for two predictor functions is as follows:

$$\hat{s} = \begin{cases} f_1 & \text{if} \quad u_1 \leq u_2 \\ f_2 & \text{if} \quad u_1 > u_2, \end{cases} \tag{11}$$

where

$$u_i = \sum_{k \in W} |s'_k - f_i(s'_k)|. \tag{12}$$

The subscript k denotes a pel in a neighborhood W. The window W is chosen such that s'_k is known at the receiver. The decision function u_i can be evaluated at the receiver without transmission of additional information about predictor selection. For previously transmitted nearest neighbors, the index set W is

$$W_\alpha = \{1, 2, 3, 4\}. \tag{13}$$

For real-time implementation the choice of W_α creates problems, because of the use of the pel s'_1. The time constraint for calculation of u_i can be reduced by using the index set

$$W_\beta = \{2, 3, 4, 5\} \tag{14}$$

or

$$W_\gamma = \{2, 3, 4\} \tag{15}$$

instead of W_α. The window W_γ is also used by Stuller et al.[6]

A further simplification for hardware implementation can be obtained by introducing a quantizer function $Q_s[\cdot]$ in (12). Then the decision functions u_i are given by

$$u_i = \sum_{k \in W} Q_s \quad [|s'_k - f_i(s'_k)|]. \tag{16}$$

A modification which leads to a simpler implementation than the basic selection rule (11) can be described as follows. Choose the predictor function f_i which has within a window W most frequent minimum magnitude of the difference

$$d_{ik} = s'_k - f_i(s'_k). \tag{17}$$

In the case of two predictor functions at each position k, a binary variable v_k which describes which predictor function is better, is defined as follows,

$$v_k = \begin{cases} 1 & |d_{1k}| \leq |d_{2k}| \\ 0 & |d_{1k}| > |d_{2k}|. \end{cases} \tag{18}$$

The decision functions u_i are now given by

$$u_1 = \sum_{k \in W} v_k$$

$$u_2 = \sum_{k \in W} \bar{v}_k, \tag{19}$$

where $\bar{v}_k$ is the complement of v_k. The predictor with smallest value u_i is chosen. The selection rules as discussed above require that one predictor function be chosen even if both decision functions u_i are identical. An improvement can be obtained by using a "soft-predictor

switch," i.e., the prediction value is a weighted sum of predictor functions, as given by (4), with the weights b_i being proportional to the frequency of preference of the predictor function f_i. Hence, for two predictor functions,

$$b_1 = \frac{1}{n} \sum_{k \in W} v_k$$

$$b_2 = \frac{1}{n} \sum_{k \in W} \bar{v}_k, \tag{20}$$

where n is the number of pels in the window W. To avoid the division by 3, for $W = W_\alpha$ the contribution of the pel at position 3 to (20) is doubled, and n is chosen to be 4 for this special case.

2.2 Adaptive prediction based on a steepest descent method

The steepest descent[7] is a mathematical method which has been often used for optimization. One advantage of this method is its simplicity. This method has been used frequently for adaptive systems. It is also proposed by Netravali and Robbins[8] and Stuller et al.[6] for motion-compensated prediction. Here it will be applied to adaptive intra-interframe prediction.

Let us assume that the prediction value is a weighted sum of predictor functions as given by (5). Then the prediction error is given by

$$e = s - \mathbf{b}^T\mathbf{f}. \tag{21}$$

In the following, the present pel is denoted by s, rather than s_0. The variance of the prediction error e is a quadratic function in $\mathbf{b}$.

$$F(\mathbf{b}) = E[(s - \mathbf{b}^T\mathbf{f})^2], \tag{22}$$

where $E[\cdot]$ is the expected value. The gradient with respect to $\mathbf{b}$ is given by

$$\begin{aligned} \mathbf{g} = \nabla_b F(\mathbf{b}) &= -2E[(s - \mathbf{b}^T\mathbf{f})\mathbf{f}] \\ &= -2E[e\mathbf{f}]. \end{aligned} \tag{23}$$

The steepest descent is an iterative method, where starting from an initial guess the vector $\mathbf{b}$ is modified recursively according to,

$$\mathbf{b}^{(m+1)} = \mathbf{b}^{(m)} - \gamma^{(m)}\mathbf{g}^{(m)}. \tag{24}$$

The adjustment of the vector $\mathbf{b}^{(m)}$ is made in the direction of the negative gradient. The scalar $\gamma^{(m)}$ has to be optimized by a one-dimensional search scheme at each step m. However, real-time appli-

cations are performed with a constant value of γ. The best value of γ depends on the type of data. In addition, the value of γ influences the stability and the speed of convergence of $\mathbf{b}$.

From eqs. (23) and (24), it follows that an adaptive prediction scheme which, based on a gradient method, is given by

$$\mathbf{b}^{(m+1)} = \mathbf{b}^{(m)} + 2\gamma E_W[e\mathbf{f}]^{(m)}, \tag{25}$$

where $E_w[\cdot]$ is the expected value within a small window of neighboring pels as given by (13), (14), or (15). The coefficient vector $\mathbf{b}$ is updated on a pel-by-pel basis along the scanning direction, i.e., if $\mathbf{b}^{(m+1)}$ is the coefficient vector at the present pel, $\mathbf{b}^{(m)}$ is the coefficient vector at the previous pel. At the beginning of each line, an initial estimate of $\mathbf{b}$ is used, e.g., the mean of $\mathbf{b}$ at the previous line. Simulations indicate that because of a fast adjustment an initial vector $\mathbf{b}$ with elements $b_i = 1/M$, $i = 1, 2, \cdots M$ is appropriate.

In this study, several modifications of the recursion given by (25) have been investigated. The various algorithms will be compared with respect to prediction gain and cost of implementation. A high prediction gain requires an appropriate value of γ in (25). Simulations with several values of γ indicate that for video signals with normalized range [0, 1] the optimum value of γ is about one. In such a case the adjustment from pel to pel is relatively small, and the transition from one predictor function to another takes several pels. By introducing an additional constraint

$$\sum_{j=1}^{M} b_j = 1, \tag{26}$$

the value of optimum γ is increased to about 64. The increased value of γ provides a shorter transition from one predictor function to another and the constraint (26) improves the stability of the algorithm.

With the constraint of (26), the steepest descent method has to be modified to minimize the augmented function of (22)

$$\Phi(\mathbf{b}, \lambda) = E[(s - \mathbf{b}^T\mathbf{f})^2] + \lambda(\mathbf{b}^T\mathbf{o} - 1), \tag{27}$$

where $\mathbf{o}$ is a vector with all elements equal to 1. The coefficient vector $\mathbf{b}$ is updated recursively by

$$\mathbf{b}^{(m+1)} = \mathbf{b}^{(m)} - \gamma(-2E[e\mathbf{f}]^{(m)} + \lambda^{(m)}\mathbf{o}). \tag{28}$$

Using (26) to eliminate $\lambda^{(m)}$ from (28), and replacing $E[\cdot]$ by $E_W[\cdot]$, then

$$\mathbf{b}^{(m+1)} = \mathbf{b}^{(m)} + 2\gamma\, \mathbf{C}\, E_W[e\mathbf{f}], \tag{29}$$

where $\mathbf{C}$ is an $M \times M$ matrix given by

$$\mathbf{C} = \mathbf{U} - \frac{1}{M}\mathbf{o}\mathbf{o}^T = \begin{vmatrix} 1-\frac{1}{M} & -\frac{1}{M} & \cdots & -\frac{1}{M} \\ -\frac{1}{M} & 1-\frac{1}{M} & \cdots & -\frac{1}{M} \\ \vdots & \vdots & & \\ -\frac{1}{M} & -\frac{1}{M} & \cdots 1 & -\frac{1}{M} \end{vmatrix} \tag{30}$$

and $\mathbf{U}$ is the unit matrix. Because of (26), it follows that

$$s - \mathbf{b}^T\mathbf{f} = \mathbf{b}^T(s\mathbf{o} - \mathbf{f}) = \mathbf{b}^T\mathbf{d}, \tag{31}$$

where $\mathbf{d}$ is a vector of differences similar to (17). This leads to an equivalent recursion of (29), given by

$$\mathbf{b}^{(m+1)} = \mathbf{b}^{(m)} - 2\gamma\mathbf{C}E_W[e\mathbf{d}]. \tag{32}$$

In the recursions given above, the coefficient vector $\mathbf{b}$ at the previous pel is updated by an adjustment to obtain the coefficient vector at the present pel. However, a picture is two-dimensional in nature, the values of $\mathbf{b}$ for pels from the previous line in the immediate neighborhood of the present pel are quite close to that of the present pel. This idea results in a modification of (25) which is given below.

$$\mathbf{b}^{(m+1)} = E_W[\mathbf{b}]^{(m)} + 2\gamma\, E_W[e\mathbf{f}]^{(m)}. \tag{33}$$

Let us assume that the samples s and the predictor functions f_i are represented by 8 bits. In such cases, in the recursions given above at each position within the window, a product of two 8-bit numbers has to be calculated. A reduction in the cost of implementation can be achieved by using the three-level quantizer, shown in Fig. 3, for the prediction error e and the differences d. These investigations show that a three-level quantizer with a dead zone is more advantageous than the signum function used by Netravali and Robbins.[8]

The algorithm (29) and (33) for the case of two predictor functions, in combination with a three-level quantizer Q_D, results in the following recursive scheme,

$$\begin{aligned} b_1^{(m+1)} &= E_W[b_1]^{(m)} + \gamma E_W[Q_D(e)Q_D(f_1 - f_2)]^{(m)} \\ b_2^{(m+1)} &= E_W[b_2]^{(m)} - \gamma E_W[Q_D(e)Q_D(f_1 - f_2)]^{(m)}, \end{aligned} \tag{34}$$

with the constraints

$$\begin{aligned} b_1 + b_2 &= 1 \\ 0 &\le b_1 \\ 0 &\le b_2. \end{aligned} \tag{35}$$

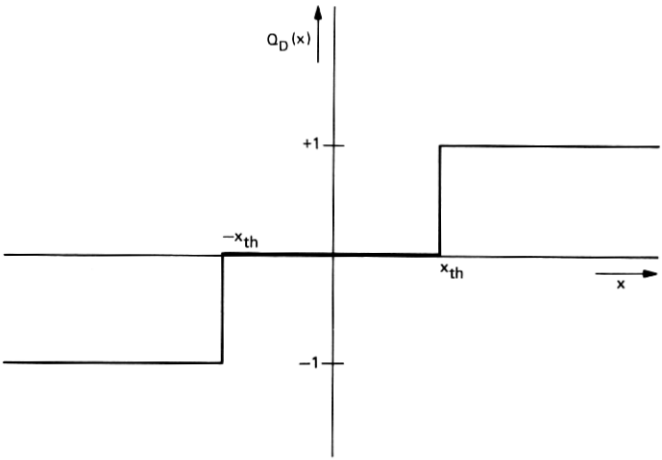

Fig. 3—Three-level quantizer for gradient quantization.

The latter two constraints of (35) were introduced to avoid negative weighting coefficients.

III. VARIABLE LENGTH ENCODING BY HORIZONTAL RUN LENGTH

An adaptive prediction scheme leads to many predictable pels. A pel is described as predictable if its quantized prediction error is represented by the level zero. To obtain a low transmission rate, the quantized prediction error is coded by a variable length code. There is always a loss in mean transmission rate compared to the entropy if not all of the negative logarithm of the probability of the prediction error representative levels are integer. This loss is especially high if one level has a probability much larger than 0.5. For adaptive prediction schemes, this is true for the quantizer level zero. To overcome this problem, block coding is frequently used. For the application described, a special coding scheme is proposed.

From each frame, a two-level picture is generated which indicates where the pels with zero code words (ZCW) and where the pels with nonzero code words (NZCW) are located. This new picture can be coded by known one-dimensional and two-dimensional coding techniques for two-level pictures. The NZCWs are coded in parallel by a variable-length code like a Huffman code and multiplexed with the code words of the two-level picture such that the receiver can decide between the two types of data. A block diagram of such a coder is shown in Fig. 4.

For a horizontal run length code, the set of symbols to be coded is listed in Fig. 5. For each of the sets, i.e., zero runs (ZR), nonzero runs (NZR) and nonzero code words (NZCW), a variable length code can be determined independently and matched to the probability of the

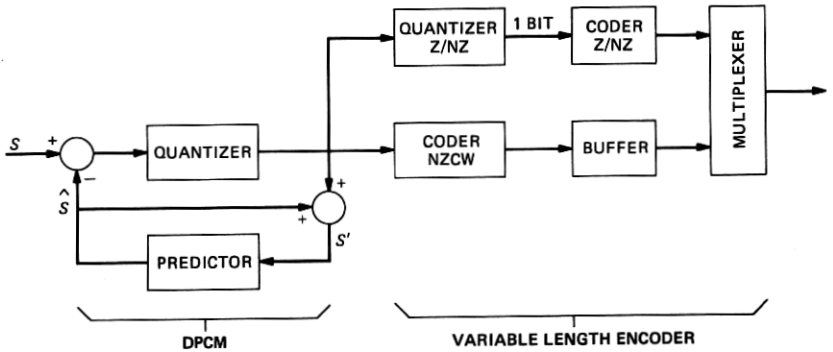

Fig. 4—Block diagram of a new type of variable length encoding.

REGULAR SET OF CODE WORDS

$\{0, 1, 2, \ldots, k\}$

NEW SETS FOR CODING

(i) Set of nonzero code words (NZCW)

$\{1, 2, \ldots, k\}$

(ii) Set of zero runs (ZR)

i	ZR
0	1
1	01
2	001
3	0001
⋮	⋮ ⋮ ⋱
n	0000 . . . 01
$n+1$	0000 . . . 00

ii) Set of nonzero runs (NZR)

i	NZR
0	0
1	10
2	110
3	1110
⋮	⋮ ⋮ ⋱
m	1111 . . . 10
$m+1$	1111 . . . 11

Fig. 5—Set of symbols for horizontal run length coding.

symbols of that particular set (e.g., Huffman code). The type of runs are chosen so as to allow a wrap-around coding from line to line. Wrap-around coding means that a run is not terminated at the end of a line but continued in the next line. Furthermore, the longest run to be

coded could be shorter than one line. The code words must be transmitted in a sequence so that the receiver always knows which code table must be used for decoding. Fig. 6 gives an example in which a ZR is transmitted at the beginning of a line. In this example, it is also assumed that the NZCWs are transmitted just after the corresponding run.

The entropy

$$H = -\sum_i p_i \log p_i \tag{36}$$

is used as an estimate for the mean code word length, with p_i being the relative frequency of the ith code word derived from the DPCM simulation of a TV sequence. The variable length code described above consists of three independent codes. Hence, the entropy H_{RUN} in bits per sample is given by

$$H_{\mathrm{RUN}} = \frac{n_{\mathrm{NZCW}}}{n_{\mathrm{PEL}}} H_{\mathrm{NZCW}} + \frac{n_{\mathrm{ZR}}}{n_{\mathrm{PEL}}} H_{\mathrm{ZR}} + \frac{n_{\mathrm{NZR}}}{n_{\mathrm{PEL}}} H_{\mathrm{NZR}}, \tag{37}$$

where n is the number of events specified by the subscript.

An advantage of the type of run length coding presented here is that in the case of statistically independent symbols, the overall entropy is not changed ($H_{\mathrm{PEL}} = H_{\mathrm{RUN}}$). In the case of interframe coding, the zeros and nonzeros are grouped together because they are related to the picture content. In this case, a decrease in entropy is achieved by the horizontal run length coding.

IV. SIMULATION RESULTS

Computer simulations were performed for the prediction algorithms given above using three different television sequences. These sequences are the same as those used in Refs. 6 and 8. Each sequence

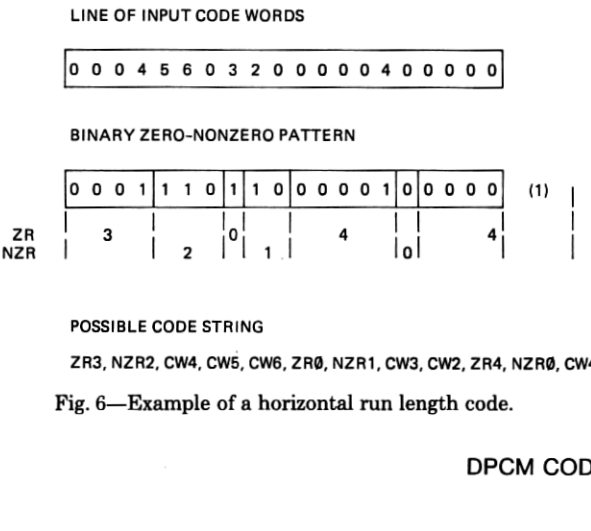

Fig. 6—Example of a horizontal run length code.

consists of 60 frames obtained by sampling a video signal of 1-MHz bandwidth, at the Nyquist rate. Each sample was quantized to 8 bits. One frame of each sequence is shown in Fig. 7.

One scene, called Judy, is a head-and-shoulders view of a person engaged in active conversation. The second scene, John and Mike, shows two people entering the camera field of view and walking briskly around each other. The third sequence, Mike and Nadine, is a panned view of two people always in view of the camera.

Even though the quantizer characteristic of a DPCM coder should be designed according to the prediction scheme, for simplification in these investigations, the same 35-level quantizer shown below was used for all simulations. The quantizer has the following positive representative levels: 0, 5, 12, 19, 28, 37, 46, 57, 68, 79, 90, 103, 116, 129, 142, 155, 168, 181. This quantizer was chosen since it gave good picture quality, although the quantization error was visible in specific picture areas under short viewing distance. The decision levels are always in the middle between two succeeding levels. The performance of the predic-

Fig. 7a One frame out of each sequence—Scene Judy.

Fig. 7b One frame out of each sequence—Scene John and Mike.

Fig. 7c One frame out of each sequence—Scene Mike and Nadine.

tion schemes was evaluated by computing the pel entropy, the entropy of a horizontal run length code, and the variance of the quantized prediction error.

For comparison of adaptive and nonadaptive schemes, results for four nonadaptive predictors were obtained. The nonadaptive prediction schemes which were used are given below.

$$\hat{s} = s'_{20} \tag{38}$$

$$\hat{s} = s'_1 - s'_{21} + s'_{20} \tag{39}$$

$$\hat{s} = \frac{3}{4}s'_1 - \frac{2}{4}s'_2 + \frac{3}{4}s'_3 + \frac{3}{4}s'_{20} - \frac{2}{4}s'_{21} + \frac{1}{4}s'_{22} - \frac{2}{4}s'_{23} \tag{40}$$

$$\hat{s} = \frac{7}{8}s'_1 - \frac{5}{8}s'_2 + \frac{6}{8}s'_3. \tag{41}$$

The first predictor (38) is simple previous frame prediction. The prediction scheme given by (39) is frequently proposed for interframe coding.[5,9] The predictor (40) is a three-dimensional predictor proposed by Klie[10] for moving areas of a picture. Equation (41) describes an intraframe predictor which minimizes the variance of the prediction error.[11]

The results of the nonadaptive predictors are shown in the upper part of Tables Ia, b, and c. These investigations show that previous frame prediction (38) is advantageous for sequences with not much motion (Judy), while the intraframe predictor (41) and the predictor (40) are better for sequences with rapidly moving objects (Mike and Nadine). An additional decrease in entropy can be obtained by using the horizontal run length coding scheme. This gain is especially high (16 percent) for the sequence Judy where ZR and NZR are better grouped.

Table Ia—Entropy per pel and variance of the prediction error for nonadaptive and adaptive predictors—Scene Judy.

Entropy in Bit Per Pel		Variance	
H_{PEL}	H_{RUN}	$E[e^2]$	Prediction Scheme
1.035	0.875	16.6	Previous frame, eq. (38)
1.120	0.953	8.5	2-D Interframe, eq. (39)
1.349	1.297	9.1	3-D Interframe, eq. (40)
1.840	1.760	31.7	2-D Intraframe, eq. (41)
0.838	0.765	5.3	Predictor selection, eq. (11), (12), $W\alpha$
0.781	0.718	4.8	Predictor selection with soft switch eq. (18), (20), $W\alpha$
0.783	0.730	4.9	Gradient algorithm, eq. (34), $W\alpha$

Table Ib—Entropy per pel and variance of the prediction error for nonadaptive and adaptive prediction—Scene John and Mike.

Entropy in Bit Per Pel		Variance	
H_{PEL}	H_{RUN}	$E[e^2]$	Prediction Scheme
2.393	2.190	142.1	Previous frame, eq. (38)
2.400	2.286	114.1	2-D Interframe, eq. (39)
2.154	2.094	61.5	3-D Interframe, eq. (40)
2.397	2.323	88.9	2-D Intraframe, eq. (41)
1.795	1.711	39.6	Predictor selection, eq. (11), (12), $W\alpha$
1.774	1.687	36.7	Predictor selection with soft switch eq. (18), (20), $W\alpha$
1.724	1.629	34.2	Gradient algorithm, eq. (34), $W\alpha$

Table Ic—Entropy per pel and variance of the prediction error for nonadaptive and adaptive predictors—Scene Mike and Nadine.

Entropy in Bit Per Pel		Variance	
H_{PEL}	H_{RUN}	$E[e^2]$	Prediction Scheme
2.859	2.809	194.9	Previous frame, eq. (38)
3.008	2.982	250.0	2-D Interframe, eq. (39)
2.537	2.504	108.0	3-D Interframe, eq. (40)
2.546	2.506	117.1	2-D Intraframe, eq. (41)
2.385	2.353	87.4	Predictor selection, eq. (11), (12), $W\alpha$
2.370	2.336	80.8	Predictor selection with soft switch eq. (18), (20), $W\alpha$
2.325	2.284	77.2	Gradient algorithm, eq. (34), $W\alpha$

Adaptive prediction schemes as given in Section II were simulated with (38) and (41) as predictor functions. The average bit rate per pel for three schemes are shown in the lower part of Tables Ia, b, and c. The adaptive schemes give an additional decrease in entropy if the horizontal run length coding technique is used; this improvement depends upon the type of picture.

Compared to the case of simple previous frame prediction, the predictor selection in combination with horizontal run length coding results in reductions of 18 to 29 percent. The corresponding reductions for the more sophisticated gradient method are 20 to 32 percent. The minimum and maximum entropy of a single frame within a sequence are reduced by about the same amount as the average entropy of the sequence. This can be recognized for the gradient method in Fig. 8, which shows the entropy per pel of each frame versus frame number.

In Section II, several modifications of the basic methods, to obtain a simpler hardware implementation, were presented. Most of these modifications have only a small influence on the entropy. The basic predictor selection scheme requires the summations of 8-bit numbers for determination of the decision functions (12). A coarse four-level quantizer

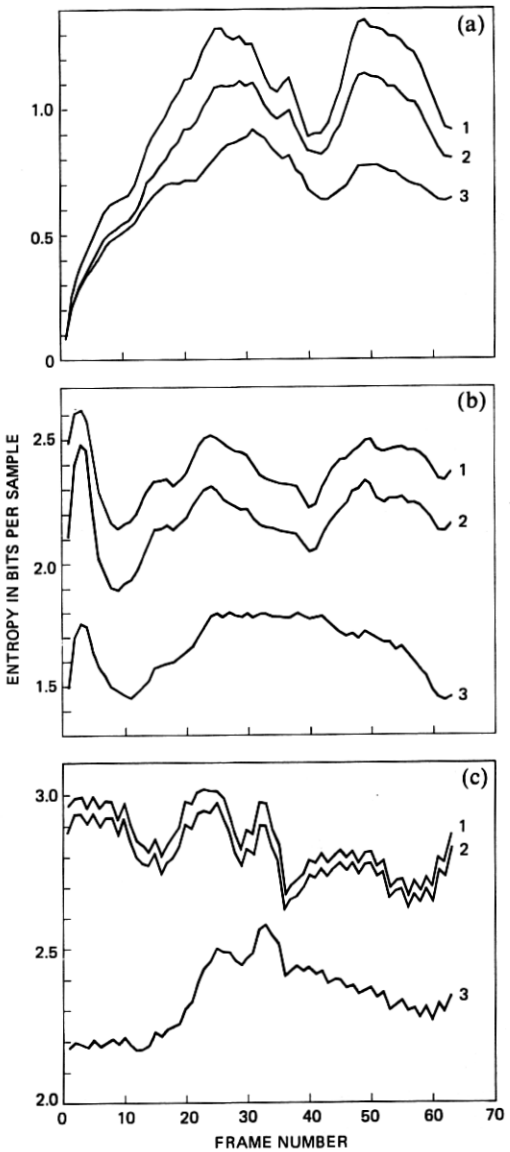

Fig. 8—Plots of entropy per pel versus frame number for each sequence. Configuration one shows the pel entropy H_{pel} of previous frame prediction; two shows the horizontal run length entropy H_{RUN} of previous frame prediction; and three shows the horizontal run length entropy H_{RUN} of the gradient algorithm (33) with the constraint (26). (a) Scene Judy. (b) Scene John and Mike. (c) Scene Mike and Nadine.

$$Q_s(x) = \begin{cases} 0 & 0 \leq |x| < 6 \\ 1 & 6 \leq |x| < 18 \\ 2 & 18 \leq |x| < 36 \\ 4 & 36 \leq |x| \end{cases} \tag{42}$$

for determination of the decision function (16) increases the entropy by about 1 percent.

The use of a binary variable v_k, equation (18), which indicates which predictor function is advantageous at the position k, in combination with the soft-switch algorithm of equation (20) is to be preferred. Compared to the predictor selection scheme (11), (12), this algorithm provides a reduction of up to 7 percent in entropy. In addition, it is easier to implement.

For the gradient method the algorithm (34) which incorporates several modifications of the original method is useful concerning cost of implementation and the reduction in entropy. The constraint (26) is especially advantageous. For the algorithm (34), a three-level quantizer with thresholds at ± 4 was used. The optimum value of γ was found to be 1/4. Each line started with initial values $b_1 = 1/2$ and $b_2 = 1/2$ for b. As long as the weighting coefficients b_j are represented by more than 4 bits, the gradient method provides a small gain in entropy compared to the predictor selection schemes.

In these investigations, three windows W_α, W_β, and W_γ were used. The window W_β provides results very close to that of W_α, whereas W_γ provides an increase of about 2 percent in entropy.

Further, it was found that using three predictor functions (the intraframe predictor is now split into two functions, one for horizontal prediction and one for vertical prediction) is not better. Besides the intraframe predictor function (39), the predictor function

$$f_2 = \frac{1}{2} s'_1 + \frac{1}{2} s'_3 \tag{43}$$

was also used. This resulted in an increase of 4 to 5 percent in the entropy.

It is of interest to know how these adaptive schemes perform in comparison with conditional replenishment and displacement compensation schemes. The results published in Ref. 6 (Table I, page 1235) based on the same source data are of some interest in this context. Hence, a comparison is possible, but it should be noted that the 35-level quantizer used in this investigation is a modification of the one used in Refs. 6 and 8. Further in this investigation, an additional thresholding of prediction error is not performed.

Compared to conditional replenishment the adaptive schemes provide a reduction in entropy of 19 to 38 percent, depending upon the

scene. For active scenes like John and Mike and Mike and Nadine, the adaptive schemes provide a data rate close to that of displacement compensation (within ±5 percent range). The run length coding scheme provides an additional reduction in entropy for sequences with low activity. For the sequence Judy, this reduction is 26 percent compared to conditional replenishment in case of previous frame prediction in combination with run length coding.

V. CONCLUSION

The performance of two types of adaptive intra-interframe predictors in combination with horizontal run length coding was studied. The gain in entropy of the predictor selection scheme is nearly as high as that of an adaptive scheme which is based on a gradient technique. Various modifications of the two basic methods which were investigated provided only small changes in entropy. Therefore, the adaptive algorithm which has the lowest cost of implementation should be chosen.

Further investigations are necessary for the quantizer design and the buffer control in a fixed rate system. A combination of the described adaptive intra-interframe algorithms with motion compensation will result in a more sophisticated system which provides further entropy reduction.

REFERENCES

1. F. W. Mounts, "A Video Encoding System Using Conditional Picture-Element Replenishment," B.S.T.J., *48,* No. 7 (September 1969), pp. 2545–54.
2. B. G. Haskell, F. W. Mounts, and J. C. Candy, "Interframe Coding of Videotelephone Pictures," Proc. IEEE, *60,* No. 7 (July 1972), pp. 792–800.
3. J. O. Limb, R. F. W. Pease, and K. A. Walsh, "Combining Intraframe and Frame-to-Frame Coding for Television," B.S.T.J., *53,* No. 6 (August 1974), pp. 1137–73.
4. H. Wendt, "Interframe-Codierung für Videosignale," Internationale Elektr. Rundschau (1973), pp. 2–7.
5. B. G. Haskell, "Frame Replenishment Coding of Television," in *Image Transmission Techniques,* W. K. Pratt, ed., New York: Academic Press, 1979.
6. J. A. Stuller, A. N. Netravali, and J. D. Robbins, "Interframe Television Coding Using Gain and Displacement Compensation," B.S.T.J., *59,* No. 7 (September 1980), pp. 1227–40.
7. P. R. Adby and M. A. H. Dempster, *Introduction to Optimization Methods,* London: Chapman and Hall, 1974.
8. A. N. Netravali and J. D. Robbins, "Motion-Compensated Television Coding: Part I," B.S.T.J., *58,* No. 3 (March 1979), pp. 631–70.
9. H. Yasuda et al., "Transmitting 4-MHz TV Signals by Combinational Difference Coding," IEEE Trans. Commun., *COM-25,* No. 5 (May 1977), pp. 508–16.
10. J. Klie, "Codierung von Fernsehsignalen für niedrige Übertragungsbitraten," Ph.D. Dissertation, University of Hannover, West Germany, 1978.
11. P. Pirsch and L. Stenger, "Statistical Analysis and Coding of Color Video Signals," Acta Electronica, *19,* No. 4 (1976), pp. 277–87.