California, Berkeley, and the M.S. degree in information and computer science from the University of California, San Diego, in 1966 and 1969, respectively. In 1970, he was awarded an Academic Fellowship from the Naval Undersea Center, San Diego, and received the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1976.

Dr. Roese is currently with the Naval Ocean Systems Center, San Diego. He has conducted research in the area of automatic sonar target classification, interframe coding of digital television images, stereoscopic display techniques, and image enhancement and feature detection as applied to sonar data displays.

★

William K. Pratt (S'57-M'61-SM'75) received the B.S. degree in electrical engineering from Bradley University, Peoria, IL, in 1959, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1961 and 1965, respectively.

He received Masters and Doctoral fellowships from Hughes Aircraft Company, and was employed there from 1959 to 1965. He became an Assistant Proessor of Electrical Engineering at the University of Southern California in 1965, an Associate Professor in 1969, and a Full Professor in 1975. In this capacity he is presently concerned with teaching and research in the areas of image processing and laser communications. Dr. Pratt is the former Director of the University of Southern California Image Processing Institute and of the Engineering Computer Laboratory. In 1976 he was awarded a Guggenheim fellowship for research in image analysis techniques.

Dr. Pratt is a member of Sigma Tau, Omicrom Delta Kappa, Sigma Xi, and the Optical Society of America.

★

Guner S. Robinson (M'70) was born in Nazilli, Turkey. She received the M.S. degree from Istanbul Technical Institute University, Istanbul, Turkey and the Ph.D. degree from the Polytechnic Institute of Brooklyn, NY, both in electrical engineering in 1961 and 1966, respectively. She has been the recipient of several fellowships including the Fulbright Exchange Student Fellowship for Ph.D. studies at the Polytechnic Institute of Brooklyn and a Graduate Assistantship at that same institution.

During 1966-1968 she was an Assistant Professor in the Department of Electrical Engineering of the Middle East Technical University, Ankara, Turkey. From 1968 to 1973, she was a Member of the Technical Staff of the COMSAT Laboratories, Gaithersburg, MD, where she conducted research and development in data compression of speech and television signals and also contributed to the analysis of satellite transponder nonlinearities. From 1973 to 1976, she was a Research Scientist at the Image Processing Institute of the University of Southern California, Los Angeles, California, where she conducted analysis and computer simulation of various image processing systems including intra- and interframe image coding, multi-spectral image processing and edge detection systems. Since May 1976 she has been a Member of the Research Technical Staff of the Northrop Research and Technology Center, where she is involved with real-time image processing applications with specific emphasis on real-time image coding and enhancement, image feature description, and scene analysis.

Interframe Coding of 525-Line, Monochrome Television at 1.5 Mbits/s

BARRY G. HASKELL, SENIOR MEMBER, IEEE, PAT L. GORDON, ROBERT L. SCHMIDT, AND JAMES V. SCATTAGLIA

***Abstract*—Interframe coder simulation results are described for standard-broadcast-rate monochrome television in applications where the television camera is largely stationary, such as TV-conferencing or video telephone. An algorithm for coding at 1.5 Mbits/s, whose parameters were optimized via trial and error, produces pictures with little resolution loss for low-to-moderate movement, but somewhat visible moving-area resolution loss for higher movements.**

THIS paper reports on recent research in inter-frame coding of 525-line monochrome television for possible application to video conferencing or video telephone. Basic to the assumptions underlying this work is the expectation that the television cameras will not be moved very often and that scenes will consist, for the most part, of stationary background area with moving people or objects in the foreground. Moreover, it is assumed, unlike broadcast television transmission where no visible impairment is allowed, that perceptible resolution loss in rapidly moving areas can be tolerated since it should not affect to a great degree the basic utility of the visual communication medium as long as low-to-moderate motion is portrayed with good resolution. The validity of these assumptions has not yet been tested in the marketplace; indeed, laboratory testing has only recently begun. Thus, the work still has to be classified as experimental. However, the potential savings of using interframe coding and digital transmission instead of straightforward analog transmission is sufficient to warrant careful consideration in any visual communication system design.

Manuscript received November 20, 1976; revised April 1, 1977. This paper was presented at the National Telecommunications Conference, New Orleans, LA, December 1975.

B. G. Haskell, R. L. Schmidt, and J. V. Scattaglia are with Bell Laboratories, Holmdel, NJ 07733.

P. L. Gordon was with Bell Laboratories, Holmdel, NJ 07733. He is now with Intermedics, Inc., Freeport, TX.

I. INTRODUCTION

Previous work on interframe coding has, for the most part, dealt with fairly low resolution TV signals, e.g., 267 line, 1 MHz PICTUREPHONE® [1]. However, in many applications, e.g., graphics scenes or more than one person in the picture, higher resolution is needed. This requirement has sparked interest in redundancy reduction (source coding) methods for standard broadcast rate signals, first because of the greater resolution which this picture format provides, and second because of the comparative ease of obtaining equipment operating at this rate. Interframe coding has the advantage over other methods of being able to display stationary graphics with full camera resolution while, at the same time, being able to portray movement (movement is displayed with varying degrees of resolution depending on the bit-rate provided for transmission). Its main disadvantage lies in the requirement for frame storage both in the coder and in the decoder and in the complexity of the signal processing. Thus, at the present time, interframe coder terminals are relative expensive devices. Even so, in most long distance applications the savings in transmission costs more than offset the terminal costs. This fact, coupled with the continuing trend of falling prices for digital logic and digital memory, indicates a promising potential for interframe coding.

Seven bit PCM coding of a 525-line, 4 MHz, monochrome signal sampled at the Nyquist rate requires 56 Mbits/s for transmission. Intraframe DPCM coding at 3 bits per pel (picture element) gives good results [3] and produces 24 Mbits/s, while intraframe transform coding at 2 bits/pel gives reasonable results [3] producing 16 Mbits/s. Interframe transform coding has been done [4] at 1 bit/pel or 8 Mbits/s, and interframe DPCM plus conditional replenishment coding at .75 bits/pel or 6 Mbits/s has yielded [5] video-conference quality color pictures.

The work to be described in this paper was aimed at 1.5 Mbits/s or .19 bits/pel and, generally speaking, built upon our previous work with lower resolution television, that is, followed the philosophy of DPCM and conditional replenishment. The bit-rate of 1.5 Mbits/s is very convenient as far as transmission is concerned. It is the rate of the ubiquitous and economical T1 transmission facility, as well as the Data-Under-Voice (DUV) system. Also, a variety of multiplexing and demultiplexing equipment exists which is readily available and field tested. If as little as 20 Mbits/s can be transmitted on one long-haul broadcast television facility, then over a dozen interframe coded signals could be sent via the same channel now being used to send one analog signal

II. BASIC METHODOLOGY

It is fairly common knowledge that conditional replenishment [6] is eminently well suited to operate under the assumptions outlined above, namely, that scenes consist of stationary background areas with moving objects in the foreground. With conditional replenishment, most of the transmitted information is devoted to moving objects in the picture, i.e., those areas not already available at the receiver. Moving areas are detected by a segmenter [7, 8] which examines significant frame-to-frame differences, eliminates as best as it can the ones due to noise, and blocks in small gaps between differences in order to ease the addressing requirement. Once defined, the moving-area pels are efficiently coded using higher-order frame-to-frame DPCM [9] and passed to an elastic store (buffer) to await transmission by the channel to the receiver buffer. At the decoder the received pels are inserted into the appropriate locations in a frame memory and the resulting picture is displayed.

During periods of rapid movement, too many data are generated to be sent right away by the channel, and the buffer would tend to fill up and overflow if steps were not taken to lower the data rate. Typically, this is done by reducing the resolution in the moving-areas of the coded picture. Spatial resolution is automatically reduced in proportion to the velocity of movement by the integrating effect of the television camera. This can be taken advantage of by adaptively varying the moving-area sampling rate (or subsampling) [10] in order to try to match the rate of data generation to the transmission channel rate. Subsampling (plus appropriate interpolation of untransmitted pels) can take place in either the horizontal or vertical direction.

Temporal resolution can be reduced by low-pass filtering in the temporal frequency domain [11] through the use of recursive filters. The effect of such filtering is to blur the moving-areas in proportion to their velocity in the same manner, but to a greater degree, as camera integration. Temporal filtering lowers the entropy of the DPCM differential signal and, thereby, causes a reduction in the overall data rate. Temporal resolution can also be reduced by subsampling along the time axis, i.e., lowering the transmitted frame rate. This usually causes jerkiness of moving objects, although the effect can be reduced somewhat by proper interpolation of the untransmitted frames.

Pel amplitude resolution can be reduced by varying the quantization of the DPCM signal. In general, the coarser the quantization, the lower the DPCM entropy and amplitude resolution. However, if the quantization is made too coarse the basic efficiency of the DPCM coding and of conditional replenishment itself can be destroyed.

The result of this general approach is a "multimode" interframe coder. During periods of low movement, the system operates in its lower order modes, and moving-areas are displayed with little loss of resolution. As movement increases, the coder progressively switches to its higher order modes, and more and more resolution loss is introduced into the moving-areas of the picture. The mode switching takes place under control of the buffer queue-length. As the queue-length increases to a point where buffer overflow threatens, the system switches to a higher order mode of operation. When the queue-length decreases to a safe level, a lower order mode is invoked.

® Registered service mark of American Telephone and Telegraph Company.

III. CODER SIMULATION SYSTEM

Attempting to optimize a multimode interframe coder necessarily involves the adjustment of a large number of parameters. Moreover, most, if not all, of the parameters are interdependent. To add to this difficulty, the quantity being optimized, namely picture quality, is multidimensional and highly subjective, and since, for these authors, the picture format was completely new, past experience could not be drawn upon to any great extent. Thus, in carrying out the work of this paper it was extremely important to have a versatile system whose parameters could be changed easily with the resulting effect on picture quality immediately available to an observer.

To accomplish these ends, an interframe coder simulation system was constructed for 525-line, 4 MHz monochrome television, which is capable of carrying out in real-time all of the picture processing functions mentioned above, but whose parameters are set through an attached minicomputer and software residing therein. The minicomputer is synchronized with the video system via interrupts and is capable of adaptively controlling the processing, interrogating the buffer queue-length at any time and statistically measuring the behavior of various components during the simulation. Almost as important, the minicomputer allows for rapid testing of the entire system for malfunctions, which in multi feedback-loop systems such as this can be obscure and extremely difficult to diagnose. As in previous work, all of the coder functions are implemented with the exception of the buffer and the channel coder. The buffer is simulated using up-down counters. A system such as this is capable of producing pictures which are identical to those which would appear at a receiver in the absence of digital transmission errors and, thus, is very useful in investigating picture quality versus bit-rate tradeoff when using various algorithms.

Figure 1 shows the configuration of the real-time signal processing portion of the system. The basic clock rate of all components in 8 MHz. Not shown are the PCM coder and decoder (both operating at 8 MHz), the minicomputer and the numerous registers necessary for delay equalization. Construction and design of hardware was simplified wherever possible by using computer loadable random-access-memories (RAM's) as look-up tables and by taking advantage of the computer's ability to perform logical operations and output appropriate control signals. What follows is a brief description of the components of the system and the signal processing functions which they individually perform. How they are made to work in conjunction with one another is described later.

Input Signal Processing–RAM's A and B

By means of a selector switch, the input signal to the system can be either 8-bit PCM video, video passed through RAM-*B* or the output of RAM-*A*. By appropriate loading of RAM-*B*, the PCM video signal can be altered in a number of ways. For example, by deleting some low significance bits, 7-bit or 6-bit PCM can be produced. By loading with various linear functions, the contrast and brightness can be changed.

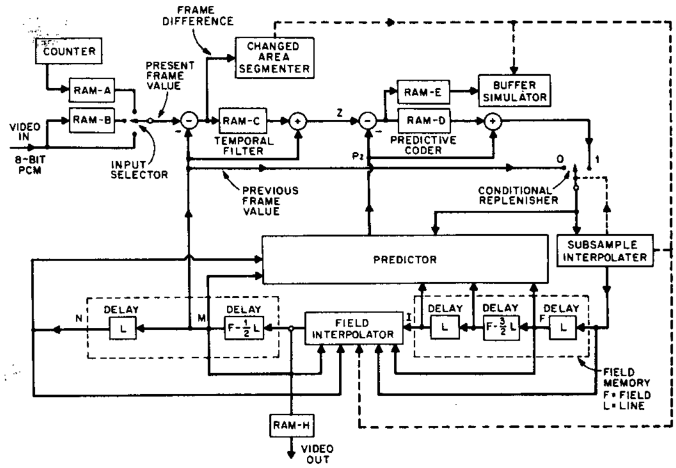


Figure 1. Configuration of the real-time signal processing portion of the coder simulation system. Basic clock rate is 8 MHz. Not shown are the PCM coder and decoder, minicomputer and the numerous registers necessary for delay equalization.

By loading with nonlinear functions, the gamma characteristic can be changed.

Connected to the address input of RAM-*A* is a counter operating at the pel sampling rate (8 MHz) and reset during the horizontal sync time. By this means, test signals can be fed into the system for debugging purposes. By rewriting RAM-*A* during the vertical blanking period, test signals with frame-to-frame variation can be generated. This is extremely useful in producing stand-alone test programs for the system.

Changed-Area Segmenter

Figure 2 shows in somewhat more detail the operation of the changed-area segmenter. The dotted lines represent control signals or data from the computer. The frame difference is first gated to eliminate pels which were or are to be deleted due to subsampling. Hereafter these will be referred to as "subsampled" pels. They will have artificially high frame differences due to interpolation error. The signal is then passed through an alterable digital low-pass filter* to remove, as much as possible, high frequency noise, and the unsigned magnitude is formed. Following this are two threshold detectors with alterable thresholds $T1$ and $T2$. Typically, $T2 > T2$, and differences $\geqslant T2$ are classified immediately as significant. However, differences $\geqslant T1$ must undergo isolated point rejection before being classified. A line delay and two tapped delay lines make available four $T1$ flags on either side of the pel of interest and three $T1$ flags from the previous line. Of these, an alterable switch selects eight flags to be used in the isolated point rejection which is implemented by appropriate loading of RAM-*F*. The output of RAM-*F* either rejects or passes $T1$ flags via an AND gate, after which they are OR'ed with the $T2$ flags. Next, in order to increase addressing efficiency [1], small gaps between significant changes are filled in or "bridged". As before, this is easily implemented using a RAM. Gaps of size 1-2, 1-4, 1-6, ···, 1-14 can be bridged by appro-

*Filter length = 5 pels. See Table III for impulse responses used.

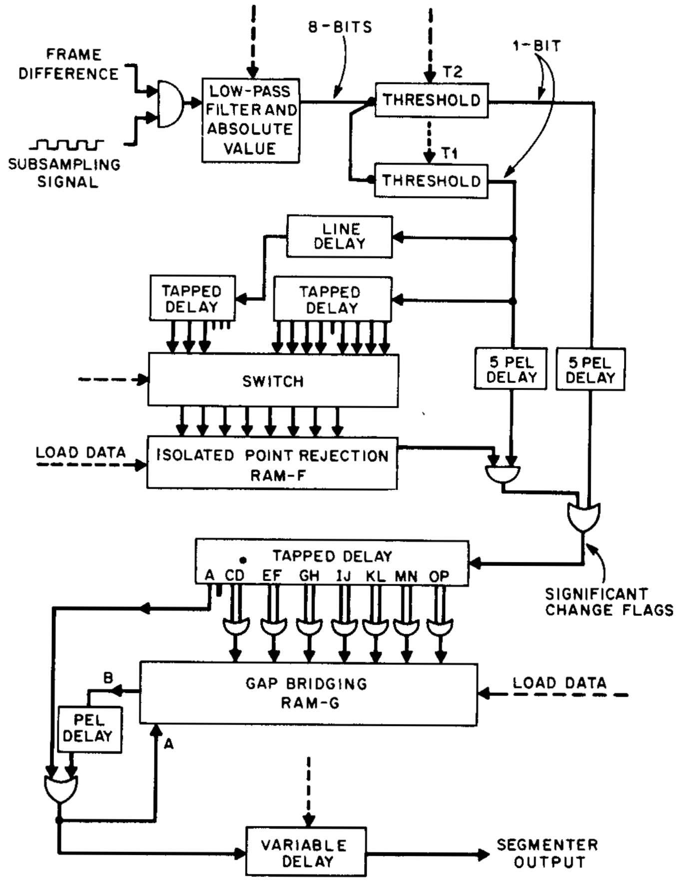


Figure 2. Changed-area-segmenter. Dotted lines represent control signals or data from the computer. This device performs frame difference filtering, thresholding, isolated point rejection and gap bridging.

PELS: A B C D E F G H I J K L M N O P

Figure 3. Configuration of pels input to the gap-bridging logic of Figure 2.

priate loading of RAM-G. If in Figure 3, pel A has changed, then RAM-G can tell if changes have also occured in pels C or D, E or F, $\cdots$ O or P. This is sufficient information to determine if pel B should be bridged or not. During the next clock period this decision is OR'ed with the actual significant change flag for the pel, after which the segmenter signal is delayed appropriately for use by the rest of the system.

Temporal Filter

A limited amount of temporal filtering can be accomplished [11] given the available memory and stability requirements. Here it is implemented by altering the magnitude of the frame difference signal. By appropriate loading of RAM-C, the frame differences can be multiplied by a constant $\leqslant 1$ or changed nonlinearly in such a way as to attenuate the small differences more than large ones. Nonlinear filtering might be done, for example, in an attempt to avoid blurring of moving edges in the scene.

Predictive (DPCM) Coder

This is a straightforward DPCM loop except that the quantizer is implemented using RAM-D. Variable word-length

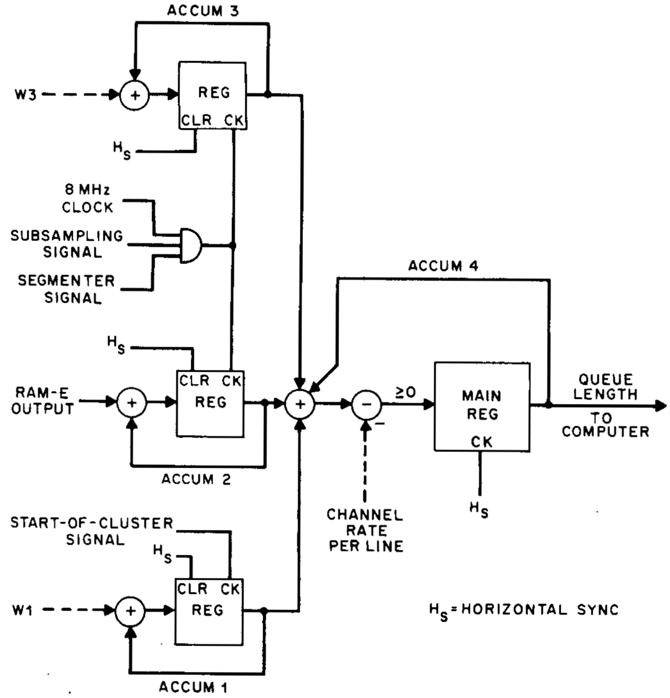


Figure 4. Buffer simulator. Accumulators 1-3 operate at 8 MHz and are reset every horizontal blanking period. Main accumulator 4 operates at the horizontal line rate.

(entropy) coding is simulated via RAM-E which, for each differential signal value, feeds a word length to the buffer simulator. By rewriting these RAM's during the vertical blanking period, the quantization can be changed to adapt to the amount of motion in the scene.

Buffer Simulator

Figure 4 shows in somewhat more detail the buffer simulator. Dotted lines indicate data from the computer. Accumulators 1-3 operate at 8 MHz and are reset every horizontal blanking period, while main accumulator 4 operates at the horizontal line rate. Accumulator 1 counts the bits required to address horizontal clusters of transmitted pels. Accumulator 2 counts, for moving-area, non-subsampled pels, the DPCM amplitude bits output by RAM-E. Accumulator 3 is similar, but counts a constant number of bits per transmitted pel. It is useful for approximating forced-updating with PCM. In the horizontal blanking period the channel-rate per line is subtracted and the result clipped to be non-negative. Then the new queue length is clocked into the main register and made available to the computer.

Conditional Replenisher

This is a simple switch which is placed in the 1 position to pass new values only for moving-area, non-subsampled pels. Otherwise it remains in the 0 position, thus repeating pels from the previous frame.

Subsample Interpolator

Here, moving-area, subsampled pels are replaced with interpolated values obtained from adjacent non-subsampled pels. Two in-field subsampling techniques were found to be useful.

With "horizontal" subsampling [10], only alternate pels in a horizontal, moving-area cluster are transmitted, the subsampled pels being linearly interpolated from their neighbors. It was found necessary to subsample one pel on either side of each cluster to avoid erroneous values and that staggering the subsampling pattern from line to line gave slightly better picture quality. With "vertical" subsampling, only alternate lines within the moving-area of a field are transmitted, the missing pels again being linearly interpolated. Since the segmenter signal is zero for the entire subsampled line, moving-area pels were identified by OR'ing the segmenter signals from the two adjacent non-subsampled lines.

Field Memories

Two field memories, each of approximately 1 Mbit capacity, were constructed using MOS shift registers. Several additional line memories were also built using the same technology. Input and output of all delay modules was 8-bit PCM at the system clock rate of 8 MHz. Configured as a tapped delay line, the memory provides signals for the temporal filter, conditional replenisher, predictor and field interpolater.

Field Interpolator

One of the most effective data-rate reduction techniques in the system is moving-area field interpolation [12]. With this method (see Figure 1) the conditional replenishment switch is held in the 0 position during alternate field periods, thus repeating the field and producing no data for transmission. One-sixtieth second after a field passes through the conditional replenisher, it arrives at the input to the field interpolator. At this time a moving-area indicator is constructed by OR'ing together two adjacent lines of segmenter signal, and moving-area pels in the repeated field are replaced by an average value obtained from adjacent fields. Background pels remain unchanged. Describing this process in terms of labeled pels (see Figure 5), a moving-area indicator for pel I is obtained by OR'ing together the segmenter signals corresponding to pels Z and F. If the indicator equals 'true', then pel I is replaced by an average of pels Z, F, M and N. Otherwise, it is left unchanged.

Predictor

The computer switchable predictor is capable of carrying out several types of frame-to-frame linear prediction based on previously transmitted pels. These are summarized in Table I for the pel configuration shown in Figure 5. In the predictions which use a previous element in the same line, pel A was used instead of the nearest pel B in order to allow for horizontal subsampling and to somewhat ease hardware implementation. Of those predictions implemented, the ones with the lowest entropy during moderate motion are the line-difference-of-frame-difference and the two-element-difference-of-frame-difference. [9] The two-line-difference-of-frame-difference predictor is useful during vertical subsampling when the previous line may have substantial interpolation error. Field-difference prediction is useful when coding a field which had previously been interpolated.

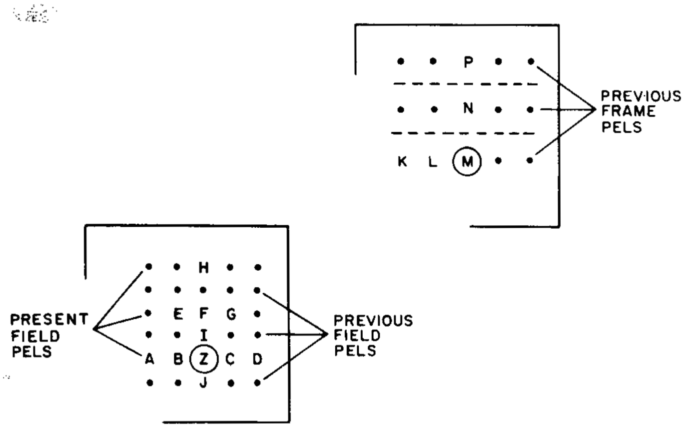


Figure 5. Configuration of pels input to the system shown in Figure 1.

TABLE I
LINEAR PREDICTORS IMPLEMENTED IN THE PREDICTOR

Differential Signal	P_Z = Prediction of Pel Z (See Figure 5)
Two Element Difference	A
Line Difference	F
Two Line Difference	H
Frame Difference	M
Field Difference	(I+J)/2
Line Difference of Frame Difference	M+F-N
Two Line Difference of Frame Difference	M+H-P
Two Element Difference of Frame Difference	M+A-K

RAM-H

This RAM is used in exactly the same way as RAM-*B* except that it modifies the output video signal. It allows, for example, compression of the video signal by RAM-*B* for coding purposes and expansion back to the original range by RAM-*H* for display.

IV. ADAPTIVE CODER STRATEGY FOR 1.5 Mbits/s

Up to the present time all efforts at coder optimization with this system have been aimed at a transmission rate of 1.5 Mbits/s. Algorithms developed for this rate will work at other rates. However, the picture quality will not be as good as might otherwise be obtained with a somewhat different algorithm optimized for the new rate. The general philosophy was, as outlined previously, to minimize moving-area resolution loss during periods of low movement by operating as much as possible in the lower order modes of the coder. For increased movement, the coder was to switch into its higher order modes, introducing loss of moving-area resolution, as needed, to avoid overflow of the buffer. The coder design was evolutionary starting with a very crude rank ordering of bit-rate reduction techniques according to their moving-area resolution loss. Using a camera of approximately 55 dB weighted

SNR**, parameters were then adjusted and refinements added to improve the overall picture quality as perceived by the authors and their colleagues. Although detailed discussion of the evolutionary design process would be very educational, space does not allow for it. Thus, what follows is a description of one of the latest algorithms developed for coding at 1.5 Mbits/s.

Twelve modes are used in this algorithm. In general, switching from mode i to the next higher mode $i + 1$ takes place as soon as the buffer queue length exceeds the Upper Buffer Threshold UBTi (see Table II for values). Switching to the next lower mode occurs after the buffer queue length falls below the Lower Buffer Threshold LBTi (and then only between fields). There are a few exceptions to this rule, and these are noted in the more detailed mode descriptions.

The two-element-difference-of-frame-difference predictor is used (except where noted) because of its ability to limit transmission errors, for the most part, to one line in the picture. Using this prediction and variable word-length coding, average bit-rates are in the range of two bits per transmitted pel.

Segmenter thresholds and frame-difference filtering are variable. Isolated point rejection uses three pels from the previous line and five pels from the present line (pels *A*-*G* and *Z* in Figure 5). Isolated double changes, i.e., *BZ* or *ZC*, are deleted as well as isolated single changes. Note from Figure 2 that isolated point rejection is non-recursive. Gaps $\leqslant 6$ pels are bridged.

In the lower modes linear temporal filtering is used, i.e., frame differences are multiplied by a constant $\leqslant 1$. Nonlinear temporal filtering is used in the higher modes to reduce entropies without excessive blurring of moving edges (see Figure 6).

Forced updating of one entire line per field (about 8 s per frame) is used in all modes except mode 0. This serves to clean up background areas and in a complete system would correct transmission errors. Approximately four bits/pel is simulated for this function by setting buffer weight $W3 = 3$ bits (see Figure 4) and segmenter threshold $T2 = 0$ (see Figure 2) for one line period. In the simulator, quantizer RAM's can only be changed in the vertical blanking period; thus, during forced updating, whatever quantizer and predictor are currently in use are employed for the actual pel updating. In a complete system, two-element-differential prediction would probably be used for forced updating.

For addressing purposes, 9 bits per horizontal sync, 9 bits per start-of-cluster address and 5 bits per end-of-cluster word are assumed. The sync word is simulated by appropriately reducing the channel rate, while cluster addressing is simulated by setting buffer weight $W2 = 14$ bits (see Figure 4). Since horizontal synchronization methods are known which only use approximately one bit per line, [11, 13] the excess sync bits could be used for error protection.

What follows is a somewhat more detailed description of the modes of the coding algorithm. Buffer queue-length thresholds, quantizers, variable word-length codes, segmenter parameters, temporal filtering and subsampling are summarized in Tables II-IV. In all modes, buffer emptying is of no concern, since in a complete system, channel stuffing bits can always be generated.

**Ratio of peak signal to standard CCIR weighted rms noise.

TABLE II
PARAMETERS OF CODING MODES. LOWER ORDER MODES ARE FOR LOW MOVEMENT; HIGHER ORDER MODES ARE FOR RAPID MOVEMENT

MODE (i)	BUFFER THRESHOLDS Lower (LBTi)	Upper (UBTi)	QUANTIZER (See Table IV)	SEGMENTER (SEE FIG. 2) Filter (See Table III)	(Scale of 0 to 255) T1	T2
0	0	6400	Fine	FA	1	6
1	1600	38400	"	FA*	1	6
2	8192	38400	"	FB	3	5
3	3200	38400	"	FA	1	3
4	3200	51200	"	"	2	3
5	3200	54400	"	FC	2	3
6	32000	70400	Coarse	FD	2	3
7	32000	83200	"	"	2	3
8	32000	96000	"	"	2	3
9	32000	100000	Fine	FA	1	3
10	64000	99200	Coarse	FD	2	3
11	64000	100000	"	"	2	3

*For noisy input signals segmenter filter *FE* is used in mode 1.

TABLE II (cont'd)
FOR NONLINEAR TEMPORAL FILTER; SEE FIGURE 6.

MODE (i)	TEMPORAL FILTER	SUBSAMPLING
0	.75	NONE
1	"	NONE
2	"	NONE
3	"	Field
4	.625	"
5	"	Field, H
6	"	"
7	Nonlinear	"
8	"	Field, H, V
9	.75	Field
10	Nonlinear	Field, H, V, Frame
11	11	"

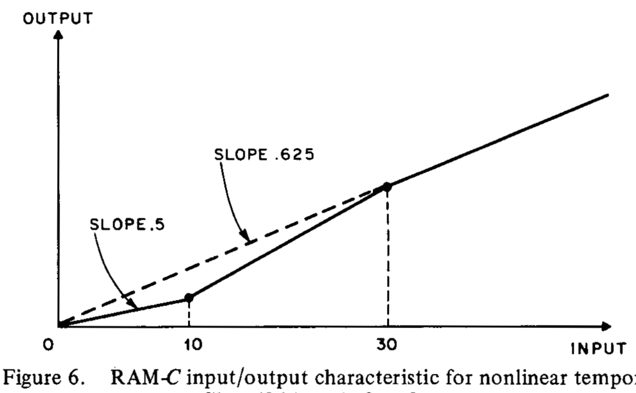


Figure 6. RAM-*C* input/output characteristic for nonlinear temporal filter (8-bit scale 0 to 255).

Mode 0 is the mode with lowest resolution loss. Within a field, normal data are transmitted until a certain line number (denoted here by $L0$) is reached. At that time, forced updating is invoked, as described previously, which continues until the buffer queue length exceeds UBT0. If this does not happen

TABLE III
SEGMENTER FRAME DIFFERENCE FILTER UNIT IMPULSE RESPONSES; FILTER LENGTH = 5 PELS

Frame Differences Filter	Impulse Response (successive pels)				
FA	.0625	.125	.25	.125	.0625
FB	.0625	.0625	.25	.0625	.0625
FC	.125	.125	.25	.125	.125
FD	.25	.125	.25	.125	.25
FE	.25	.25	.25	.25	.25

TABLE IV
DPCM QUANTIZER OUTPUT LEVELS (ON A 9-BIT SCALE -255 to 255) AND VARIABLE WORD LENGTHS OF CODE WORDS. THE 5-BIT CODE WORD 00000 IS RESERVED FOR INDICATING END OF A CLUSTER OF SIGNIFICANT CHANGES. THRESHOLD VALUES ARE HALF-WAY BETWEEN REPRESENTATIVE LEVELS.

Quantizer	Output Levels and Code Word Lengths						
Fine	Levels	0	±3	±8	±16	±27	±42
	Bits	1	2,3	4,7	7,8	8,8	8,8
Coarse	Levels		±3	±12	±24		
	Bits		1,2	3,4	6,6		

before vertical blanking, then $L0$ is reset to the beginning of the field. Otherwise, it is set to the line number in which mode 1 is invoked. $L0$ is only changed in alternate fields. The effect of mode 0 is to force update about four lines per field, i.e., one frame in about 2 s. Faster updating can obviously be obtained by either raising UBT0 or keeping track of several $L0$'s within the field. The effect on picture quality is minimal, however, except with transmission errors which, as stated previously, were not considered in this study.

Mode 1 is invoked after forced updating in mode 0 or with a slight amount of movement. It is the same as mode 0, except that as in the modes to be described below, only one line per field is forced updated. When the buffer queue-length exceeds UBT1, the coder defines the present field to be *even* and switches to *mode 3*. This is an exception to the rule of switching to the next higher state. Switching to mode 0 occurs in vertical blanking if at that time the buffer queue-length is below LBT1.

Mode 2 can only be entered from mode 3 in vertical blanking (after an odd field). It is used to coarsely code even fields which had previously been interpolated. The field-difference predictor is used for this purpose. This is the only mode which does not use the two-element-difference-of-frame-difference predictor. Switching to a higher or lower mode takes place in the normal way.

Mode 3 codes odd fields in the normal manner, but interpolates even fields as described previously. No data are transmitted for even fields during input since they are repeated at the conditional-replenishment switch. Switching to a higher mode occurs normally. Switching to a lower mode is allowed only in vertical blanking after an odd field if at that time the buffer queue-length is below LBT3.

Mode 4 comprises interpolation of even fields, more temporal filtering and higher segmenter thresholds. As stated previously, the temporal filter RAM-*C* can only be changed at the end of a field. Switching to a higher mode occurs in the normal manner. However, switching to a lower mode only takes place in vertical blanking after an odd field, and then only after ten successive frames at the end of which the buffer queue-length is below LBT4. This step is necessary to prevent switching too often between temporal filters, thus causing visible jerkiness in moving areas of the picture.

Mode 5 adds horizontal subsampling to the above. Some more segmenter frame difference filtering is also introduced, but note that with horizontal subsampling, alternate frame difference entering this filter is zero. This is equivalent to raising the segmenter thresholds somewhat. Again, switching to another mode only occurs within or at the end of an odd field.

Mode 6 introduces a coarser quantization into the DPCM loop. However, since the quantizer RAM's *D&E* can only be written during vertical blanking, nothing happens upon entering this mode until the end of the field. Coarser quantization causes an increased noise level in moving-areas; thus, more segmenter frame difference filtering is required. Mode switching occurs only within or at the end of odd fields.

Mode 7 increases the amount of temporal filtering, but, as in mode 6, this can only occur in vertical blanking. Also, odd field mode switching continues to be enforced.

Mode 8 adds vertical subsampling to the above coding methods. This occurs immediately upon entry to this mode. Odd field mode switching continues, except that in this mode when the buffer queue length exceeds UBT8 the coder switches to mode 10 (not mode 9).

Mode 9 can only be entered from mode 10. It is included in the algorithm on the assumption that a scene change or other violent movement must have occurred in mode 10 and that this activity has largely ceased. This mode constructs a full resolution odd field (even fields are still interpolated) starting from the top of the picture. Ten to 15 frame periods are required. During this process, when the buffer queue-length exceeds UBT9 the coder immediately switches to mode 10 (and then to frame repeating) as usual. However, if at the end of an odd field the buffer queue-length is below LBT9 indicating that a full resolution field has been obtained, the coder switches all the way down to *mode 3* so as not to corrupt the picture with subsampling.

Mode 10 employes 2:1 frame repeating. It is not usually entered unless the camera is being moved or a complete scene change occurs. Upon entry the present field is defined to be field 3, and thereafter fields are number modulo 4. Fields 0 and 2 are interpolated, field 1 is repeated (by setting the segmenter signal = 'false') and field 3 is coded as in mode 8. This mode causes visible jerkiness of moving objects in the scene. Switching between modes is restricted to occur within or at the end of field 3.

Mode 11 is the highest mode in this algorithm. Also, for lower bit-rates or larger buffer sizes it can be used to implement, 4:1, 6:1, $\cdots$ frame repeating. Functionally, it is exactly the same as mode 10, except that if the buffer queue length exceeds UBT11 replenishment is immediately stopped (by setting the segmenter signal = 'false') and not resumed until the coder switches to mode 10.

(a)

(b)

(c)

(d)

(e)

(f)

Figure 7. (a) and (b) uncoded frames containing little or no motion. (c) (uncoded) and (d) (coded, 1.5 Mbits/s–frames containing motion sufficiently rapid to cause operation in state 5), (e) (uncoded) and (f) (coded, 1.5 Mbits/s–frames containing violent motion sufficient to cause operation in state 7).

This coding algorithm has not been completely optimized. Parameters were obtained almost entirely via trial and error by using objects moving at various speeds and in each case trying to minimize the moving-area resolution loss. Somewhat better coder performance should result, for example, by using statistical and subjective measurements to improve the quantizers and variable word-length codes. Also, with a somewhat more sophisticated algorithm, reduction in buffer size should be obtainable.

If the input video signal has an SNR less than the 55dB used to design this algorithm, changes in the segmenter parameters are necessary. For example, using a camera with 50dB weighted SNR it was found necessary to change the mode 1 frame difference filter to *FE* in Table III and set the corresponding $T1 = 3$. AGC effects in cameras can also be bothersome if they work on the principle of constant average signal value. For then, movement of an object into and out of a scene can cause change in virtually the entire signal with resulting resolution loss in the coded picture. Undoubtedly other problems will also arise as these coding techniques move from the research stage toward practical application.

Picture Quality at 1, 1.5 and 3 Mbits/s

Beyond what was stated in the introduction to this paper, not a lot can be said regarding picture quality until tests involving actual users are conducted. There is not much point in testing for "detectable degradations" since with enough movement, this can always be obtained, and the amount of degradation users will tolerate before complaining probably depends on the cost to them of the service and on the particular value they place on good rendition of rapid movement.

Examination of single frames is also not very helpful. For example, Figures 7a and b show uncoded frames containing little or no motion. Figures 7c and d show uncoded and coded frames, respectively, containing motion sufficiently rapid to cause the coder to operate in state 5. For an unpracticed eye, degradations here are not easily seen, whereas in a live video picture they are much more discernable. Figures 7e and f show uncoded and coded frames, respectively, containing violent motion sufficient to cause the coder to operate in state 7.

Nevertheless, with these shortcomings in mind, rudimentary subjective tests were conducted using the five point scale shown in Table V, the scenes described in Table VI and three different bit rates. Results are shown in Figure. 8.

In the tests the coded pictures were recorded in 10 s segments and randomly interpersed on videotape. Each scene was used twice at each bit-rate for a total of 54 segments. Ten subjects with varying amounts of picture viewing experience participated in the tests. Results were anchored in the sense that a PCM coded picture† was shown for 60 s, and subjects were told to regard it as having imperceptible degradation. In this way tape noise and dropouts were removed from consideration. If for the same picture a subject gave scores which differed by two or more on the scale of Table V, both scores were deleted. Twenty-six of the 540 responses were dropped due to this consistancy criterion. Scores were averaged and plotted in Figure 8. At 1.5 Mbits/s, only very rapid movement gave average scores below 3.5.

†The total PCM signal chain consisting of SONY AVC-3200 camera, 4 MHz filter, ADC, DAC, 4 MHz filter and SONY AV-3650 videotape recorder produced a picture with about 300 lines resulution and weighted SNR = 48 dB.

TABLE V
FIVE POINT RATING SCALE USED IN SUBJECTIVE TESTS

	Degradation Scale
5	Imperceptible
4	Perceptible But Not Annoying
3	Slightly Annoying
2	Annoying
1	Very Annoying

TABLE VI
SCENES USED IN THE SUBJECTIVE TESTS AND IN FIGURE 7; EACH SEGMENT WAS 10 s IN DURATION

1. One person talking. (Figure 7a).
2. Two persons, one talking. (Figure 7b).
3. Graphics, low motion with pointer.
4. Two persons, one with hand, head and shoulder movement.
5. One person with hand, head and shoulder movement.
6. Graphics, with a change from one picture to another.
7. Graphics, with a switch from one camera to another.
8. Two persons, one standing up and sitting down.
9. One person standing up and sitting down.

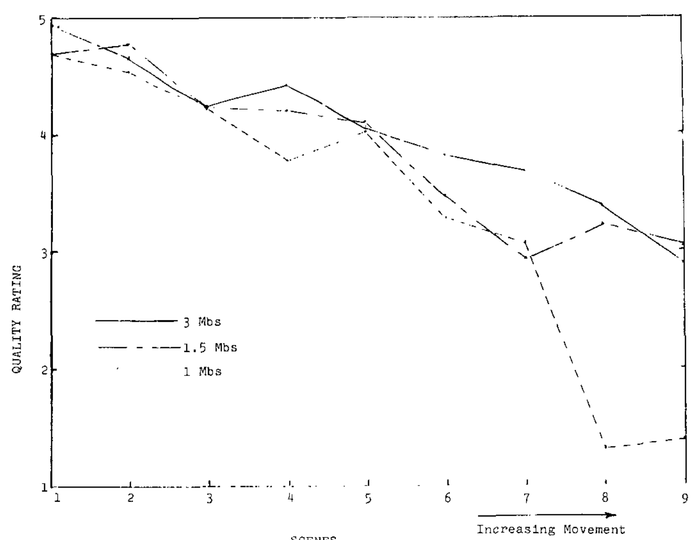


Figure 8. Results of subjective tests using the scenes of Table VI coded at three different bit rates. At 1.5 Mbits/s, only very rapid movement gave average scores below 3.5.

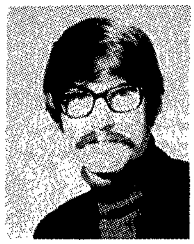
V. CONCLUSION

For certain applications which do not involve movement of the TV cameras, interframe coding can yield very significant savings in transmission. If, in addition, users can tolerate resolution loss during periods of rapid movement then the potential savings are even greater. This paper describes experimental results of interframe coder simulations in the 1-3 Mbits/s range using as an input a standard broadcast-rate monochrome signal. An algorithm was obtained for 1.5 Mbits/s coding using trial-and-error optimization of parameters which produced a coded picture with the expected characteristics, namely, good resolution during low-to-moderate movement, less resolution during rapid movement, and scene-change capability of 1 to 3 s.

REFERENCES

1. Haskell, B. G., F. W. Mounts, and Candy, J. C., "Interframe Coding of Videotelephone Pictures," *Proc. IEEE,* 60, No. 7, (July, 1972), pp. 792-800.
2. Connor, D. J., Brainard, R. C., and Limb, J. O., "Intraframe Coding for Picture Transmission," *Proc. IEEE,* 60, No. 7, (July, 1972), pp. 780-791.
3. Wintz, P. A., "Transform Picture Coding," *Proc. IEEE,* 60, No. 7 (July, 1972), pp. 809-820.
4. Knauer, S. C., "Real-Time Video Compression Algorithm for Hadamard Transform Processing," *IEEE Transactions on Electromagnetic Compatibility,* EMC-18, No. 1 (February, 1976), pp. 28-36.
5. Iinuma, K., et al., "Interframe Coding for 4-MHz Color Television Signals," *IEEE Transactions on Communications,* COM-23, No. 12 (December, 1975), pp. 1461-1466.
6. Mounts, F. W., "A Video Encoding System Using Conditional Picture-Element Replenishment," *B.S.T.J.,* 48, No. 7, (September, 1969), pp. 2545-2554.
7. Connor, D. J. Haskell, B. G., and Mounts, F. W., "A Frame-to-Frame Picturephone Coder for Signals Containing Differential Quantizing Noise," *B.S.T.J.,* 52, No. 1 (January, 1973), pp. 35-51.
8. Limb, J. O., Pease, R. F. W., and Walsh, K. A., "Combining Intraframe and Frame-to-Frame Coding for Television," *B.S.T.J.,* No. 6, (August, 1974), pp. 1137-1173.
9. Haskell, B. G., "Entropy Measurements for Nonadaptive and Adaptive, Frame-to-Frame, Linear Predictive Coding of Videotelephone Signals," *B.S.T.J.,* 54, No. 6, (August, 1975), pp. 1155-1174.
10. Pease, R. F. W., and Limb, J. O., "Exchange of Spatial and Temporal Resolution in Television Coding," *B.S.T.J.,* 50. No. 1, (January, 1971), pp. 191-200.
11. Haskell, B. G., "A Low Bit-Rate Interframe Coder for Videotelephone," *B.S.T.J.,* 54, No. 8, (October, 1975), pp. 1475-1495.
12. Limb, J. O., and Pease, R. F. W., "A Simple Interframe Coder for Video Telephony," *B.S.T.J.,* 50, No. 6, (August, 1971), pp. 1877-1888.
13. Haskell, B. G., "Conditional Replenishment Video Encoder with Sampling Grouping and More Efficient Line Synchronization," U.S. Patent No. 3, 962, 535, June, 1976.

★



Barry G. Haskell (S'65–M'68–SM'76) was born in Lewiston, Maine on September 1, 1941. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of California, Berkeley in 1964, 1965, and 1968, respectively.

From 1965 to 1968 he has a Research Assistant in the University of California Electronics Research Laboratory. Since 1968 he has been at Bell Laboratories, Holmdel, New Jersey and is presently Head of the Radio Communications Research Department. In 1977 he was a part-time Faculty Member of the Department of Electrical Engineering at Rutgers University. His research interests include television picture coding and transmission of digital and analog information via microwave radio.

Dr. Haskell is a member of Phi Beta Kappa and Sigma Xi.

★

Pat L. Gordon was born in Sapulpa, Oklahoma on October 29, 1943. He received the B.S., M.S., and Ph.D. degrees in Electrical Engineering from the University of California, Berkeley, in 1965, 1967, and 1972, respectively. He received the 1965 University of California Medal for outstanding undergraduate academic performance. He was a University Science Fellow from 1965 to 1966 and a NSF Graduate Fellow from 1966 to 1971. His graduate work was in quantum electronics.

From 1965 to 1971 he was a part-time member of technical staff at the Watkins-Johnson Company, Palo Alto, California, involved in the design of digitally organized microwave receivers. From 1972 to 1977, he was a member of the technical staff at Bell Telephone Laboratories, Holmdel, N.J., where he conducted research in digital processing of T.V. signals for efficient transmission. Since 1977, he has been Vice President of Technical Operations at Intermedics, Inc., Freeport, Texas, where he is engaged in the design and manufacture of implantable cardiac pacemakers.

Dr. Gordon is a member of the Tau Beta Pi, Phi Beta Kappa, and Sigma Xi.

★

Robert L. Schmidt received an A.S. degree in engineering from Brookdale Community College in 1976, and is now working on a B.S.E.E. at Monmouth College.

He has been employed at Bell Laboratories since 1972 and is currently a Senior Technical Associate in the Radio Communications Research Department at the Crawford Hill Laboratory in Holmdel, New Jersey.

★

James V. Scattaglia is a graduate of the Advanced Technology course at RCA Institutes. He has been employed at Bell Laboratories since 1952 and is currently a Senior Technical Associate in the Visual Communications Research department at Holmdel, New Jersey.

He has been involved with digital video processing experiments for several years. Previous fields of involvement include PICTUREPHONE® experiments and research in PCM systems.