THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 54, No. 8, October 1975
Printed in U.S.A.

A Low-Bit-Rate Interframe Coder for Videotelephone

By B. G. HASKELL and R. L. SCHMIDT

(Manuscript received January 7, 1975)

It has been suggested that customers for videotelephone service may be more interested in graphical information and in views of stationary objects than in head-and-shoulder views of people engaged in conversation. For this reason, an interframe coder simulation was constructed of a system that transmits graphics with full seven-bit PCM *resolution, but displays scenes containing much movement with visible smearing in the moving areas.*

With the coder operating at 200 kb/s (0.1 bit per pel for a 1-MHz signal), a very usable (somewhat reduced-resolution) graphics picture can be transmitted in about one-half second, which is about as fast as the human eye can assimilate the information. A full-resolution picture is built up after 3 to 5 seconds but, except for high-detail scenes, it is very difficult to tell the difference between the half-second picture and the 5-second picture.

Head-and-shoulder views of people engaged in low-key conversations are transmitted with quite adequate picture quality. Moving lips appear somewhat smeared, but it may not be enough to be objectionable if the audio is suitably delayed. However, large area movement is very visibly smeared—even to the point of being unrecognizable at moderate speeds. Whether or not this feature makes the coder unusable depends upon the value the user places on high-quality animated face-to-face conversation.

Briefly, the coder works as follows: First, the signal is temporally prefiltered. Then moving-area pels are sent as line-to-line differences of frame-to-frame differences. As the buffer fills, field-to-field, pel-to-pel, and frame-to-frame subsampling as well as adaptive quantization are brought in as needed to reduce the data rate.

I. INTRODUCTION AND SUMMARY

The use of videotelephone for graphical information and for views of stationary objects has profound implications in long-distance transmission of video signals via frame-to-frame coding, where the required data rate is directly dependent on the amount of movement to be accommodated in the scene.

If it can be shown that visible degradation of moving areas in a television picture is not detrimental to the effectiveness of visual communication, then a significant saving in transmission costs is possible with frame-to-frame coding.[1] With these techniques, stationary areas of pictures would be transmitted with full resolution, while moving areas would be sent with visibly reduced resolution.

Transmission of graphics or still pictures can be accomplished in a particularly pleasing way, subjectively speaking. A reduced-resolution, but quite recognizable, picture appears very quickly at the receiver. Full resolution is then built up over a period of time that depends on the transmission channel data rate. However, for the majority of pictures, it is difficult for an observer to tell the difference between the full resolution picture and the earlier-appearing reduced-resolution picture. In this regard, such a system would be much more usable for interactive visual communication than would a facsimile or slow-scan system operating at the same data rate where a complete picture would not be visible for a relatively long time. Also, interframe coding can handle small amounts of movement such as adding a few lines to a sketch or using a pointer with stationary graphics, whereas a slow-scan system would be very unsatisfactory.

Scenes of people engaged in conversation do not fare as well as scenes in which there is little or no movement. Moving areas such as a person's lips and eyes are visibly smeared and, depending on the data rate, large-area movement is jerky because of coder overload. Even so, a well-behaved subject can present a very decent picture to the receiver if he or she is aware of the limitations of the medium. However, it is this aspect of low-bit-rate interframe coding that raises questions in most people's minds. Whether or not this feature makes such techniques unusable depends upon the value the user places on high-fidelity, animated, face-to-face conversation.

To move closer to the answers to some of these questions, an interframe coder simulation was constructed for 1-MHz videotelephone signals that was designed to operate in the hundreds of kilobits per second range (below $\frac{1}{4}$ bit per picture element). Many techniques are used to adaptively reduce the moving-area resolution (both spatial and temporal) in proportion to the amount of motion, and to restore full resolution to the display as quickly as possible after motion ceases.

With such a system operating at 200 kb/s (0.1 bits/pel), a recognizable, somewhat reduced-resolution graphics picture is displayed at the receiver in about one-half second. Full resolution requires 3 to 5 seconds. With head-and-shoulder views of people engaged in low-key conversation, moving lips appear somewhat smeared, but this may not be enough to be objectionable if the audio is suitably delayed. However, large-area movement is very visibly smeared and jerky, even to the point of being unrecognizable at moderate speeds.

With the system operating at 50 kb/s, a reduced-resolution graphics picture requires about 2 seconds for transmission, while full resolution takes 10 to 15 seconds. At 50 kb/s, face-to-face conversation loses much of its naturalness. Lip motion can be followed only if the subject remains otherwise absolutely still, and large area motion is portrayed as a series of snapshots occurring at a rate of about 1 per second. It is interesting to note, however, that, even at 50 kb/s, useful interactive visual communication is still possible using interframe coding whereas, with slow-scan operating at the same data rate and requiring about 10 seconds per frame for transmission, interactive communication is severely hampered.

In the following sections, the technical aspects of the coder and the simulation are discussed.

II. MULTIMODE CONDITIONAL REPLENISHMENT

It is well known that, in a television signal, successive frames are very much alike. The frame-to-frame differences are negligibly small except in areas of the picture that contain moving objects. Thus, if frame memories are provided at the transmitter and receiver of a video communication system, it is necessary only to transmit those areas of each frame where the frame differences are significant. The remaining picture elements (pels) can be repeated from the previous frame. This technique is called conditional replenishment.[2] Conditional replenishment requires addressing the pels which are transmitted (the changed pels or "moving-area" pels) and buffers at the transmitter and receiver.

For example, in Ref. 3 a conditional replenishment coder for eight-bit PCM videotelephone signals[*] is described which operates at 2 Mb/s (one bit per pel on the average) and uses a number of techniques to reduce the bit rate required for transmission. The pels to be transmitted are addressed along the line in clusters, and their amplitudes are sent as frame-to-frame differences. When the transmitter buffer starts to fill, indicating active motion, only every other changed pel is transmitted,[3,4] with the unsampled pels being replaced by the average of their neighbors. When the buffer fills completely, replenishment is stopped for one frame period, allowing the buffer to empty before resuming transmission.

[*] 30-Hz frame rate, 271 lines, 2:1 interlace, 3 dB down at 1 MHz, 2-MHz sampling rate, 8-bits/sample, 210 visible samples/line.

Other multimode conditional replenishment coders are described in Refs. 5, 6, and 7. A variety of techniques control the rate of data generation to prevent buffer overflow.

Other functions of conditional replenishment coders, such as the sending of synchronizing information and the accommodation of transmission errors, are also discussed in Refs. 1 to 7.

III. LINEAR PREDICTIVE CODING

A linear predictive coder forms a prediction of each pel to be sent by computing a linear combination of previously transmitted pels. The difference between the actual value and the prediction is then quantized, coded, and transmitted. The inverse process takes place at the receiver. The better the prediction, the smaller the entropy of the differential signal and the bit rate required for transmission. Figure 1 shows two successive frames with interlacing assumed (two interlaced fields per frame). Suppose Z is a moving-area pel we wish to transmit. Pels A, B, C, G, and H are in the field presently being scanned; pels D, E, F, R, S, and T are in the previous field; and the remaining pels are one frame period back from the present field. Pel M is the previous frame value of Z, and if it is used as a prediction of Z, then Z − M, the differential signal which is transmitted, is the frame difference as discussed above.

In Refs. 8 and 9 it was found that using M + (B − J) as a prediction of Z resulted in a relatively-low-entropy, differential signal compared with other nonadaptive predictive coders. In this case, the transmitted differential signal is the line-to-line difference of the frame-difference signal (Z − M) − (B − J).

Transmitting line differences of frame differences has several other advantages as well. Since it does not use pels along the present line

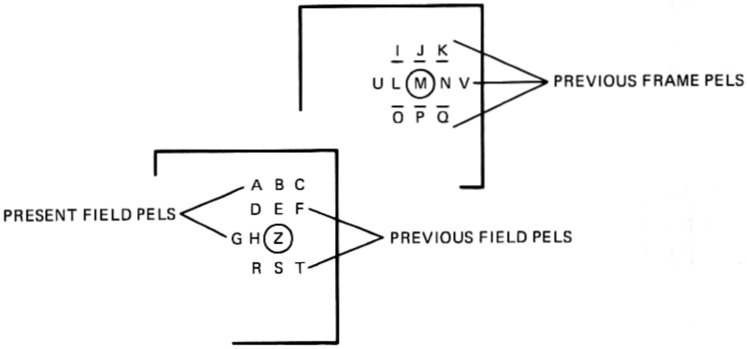


Fig. 1—Two successive television frames, interlacing assumed (two interlaced fields per frame). Pels Z and M are exactly one frame apart.

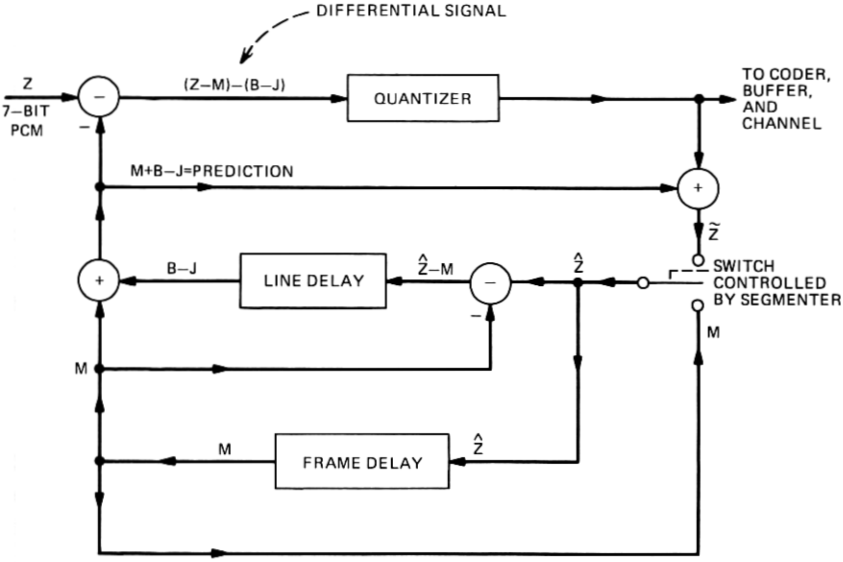


Fig. 2—Predictive coder which transmits only the moving-area pels. The differential signal (Z-M)—(B-J) is the line-to-line difference of the frame-to-frame difference. The segmenter (not shown) determines whether or not Z is a moving-area pel. If it is, the switch is put in the up position and $\tilde{Z}$, a new quantized value, enters the frame memory. Otherwise, the switch is put in the down position and the previous frame value M is recirculated. In any event, $\hat{Z}$ is the value displayed at the receiver in the absence of transmission errors.

or pels in the previous field for its prediction, pel subsampling and field subsampling can be employed without affecting the performance of the predictor. Also, it has been found that relatively few quantization levels are required to produce a good quality picture. Starting with seven-bit PCM, 11-level quantization* of the line difference of frame difference is sufficient for most pictures and most speeds of movement, whereas 30- to 40-level quantization is required for the frame-difference signal.

Figure 2 shows a single-mode conditional replenishment coder which transmits quantized line differences of frame differences in the "moving area." This is the predictive technique used in the coder described in this paper. As with all conditional replenishment coders, a "segmenter" is required to divide the picture into moving parts and stationary parts,[6,7] logic must be provided for sending addressing and synchronizing information, and a buffer is needed to smooth the data rate prior to transmission. If Z is a moving-area pel, the switch is in the up position to allow the quantized representation $\tilde{Z}$ to pass through

* On a scale of 0···127, the quantization levels are 0, ±1, ±3, ±10, ±23, ±48.

to the frame memory. If Z is a stationary-area pel, the switch is in the down position, and the previous frame pel M is repeated. At the receiver, the inverse process takes place, and the value $\hat{Z}$ is displayed.

To take advantage of the low entropy of the line difference of frame difference, a variable word-length coder should be used to code the quantized moving-area differential signal. A suitable code for 11-level quantization is given in Table I. The four-bit code word 0000 is reserved for signaling the end of a cluster of significant changes.[3] In a later section of this paper, nine-level quantization is discussed. The first nine code words of Table I are suitable for nine-level quantization.

IV. TEMPORAL FILTERING

A simple method of reducing the data rate in an interframe coder for television pictures is to subsample in the temporal direction and transmit only every other frame (odd field followed by even field) which enters the coder, i.e., send frames at a rate of only 15 Hz. At the receiver in place of each missing frame, one would display either the previous frame or an interpolation of the previous frame and the upcoming frame. However, when using this technique jerkiness is visible in the displayed picture for all except the very slowest movement.

The jerkiness is due to aliasing in the temporal-axis frequency domain, i.e., the input signal has significant power above the half-sampling frequency (here, 7.5 Hz). Aliasing can be reduced by filtering the input signal to reduce as much as possible the power above 7.5 Hz in the temporal frequency domain. Instead of jerkiness, the displayed signal then exhibits blurring in the moving area in proportion to the speed of movement. Many viewers find this type of distortion prefer-

Table I — Variable word-length code suitable for 11-level quantization with code word 0000 reserved for indicating the end of a cluster of significant changes

L_0	1
$+L_1$	01
$-L_1$	001
$+L_2$	0001000
$-L_2$	0001001
$+L_3$	0001010
$-L_3$	0001011
$+L_4$	0001100
$-L_4$	0001101
$+L_5$	0001110
$-L_5$	0001111

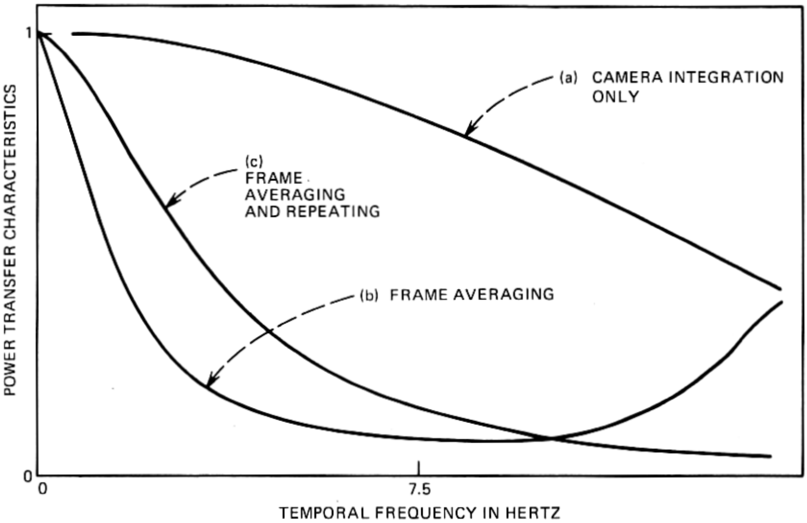


Fig. 3—Power transfer characteristics versus temporal frequency. (a) Light integration by the camera alone. (b) Simple frame averaging. (c) Averaging plus frame repeating as shown in Fig. 4.

able to jerkiness, since it is already present to some degree in all television pictures.

Ideal low-pass filtering using a $(\sin x)/x$ impulse response filter would require several frame memories. In this paper, we use a method of temporal filtering employing only one frame memory, namely, the one normally present in the interframe coder.

Some temporal filtering already takes place in a normal television camera because of its integrating action. Figure 3a shows the power transfer characteristic (derived in the appendix) owing to integration of the light falling on the camera target.

Additional temporal filtering using a frame memory can be carried out by a simple averaging of the incoming frame and the previous frame. The power transfer characteristic of this type of filtering (derived in the appendix) is shown in Fig. 3b. It is down by about 8 dB at 7.5 Hz.

Figure 4 shows the implementation of frame repeating plus temporal averaging. The switch is held in the down position during alternate input frames. Otherwise, it performs conditional replenishment under control of the segmenter as in Fig. 2. In this case, the "previous frame" coming out of the frame memory during conditional replenishment is not the previous frame at all, but, as a result of the frame repeating, it is actually the frame that was coded two frames ago. Because of this fact, increased temporal filtering occurs. Figure 3c shows the

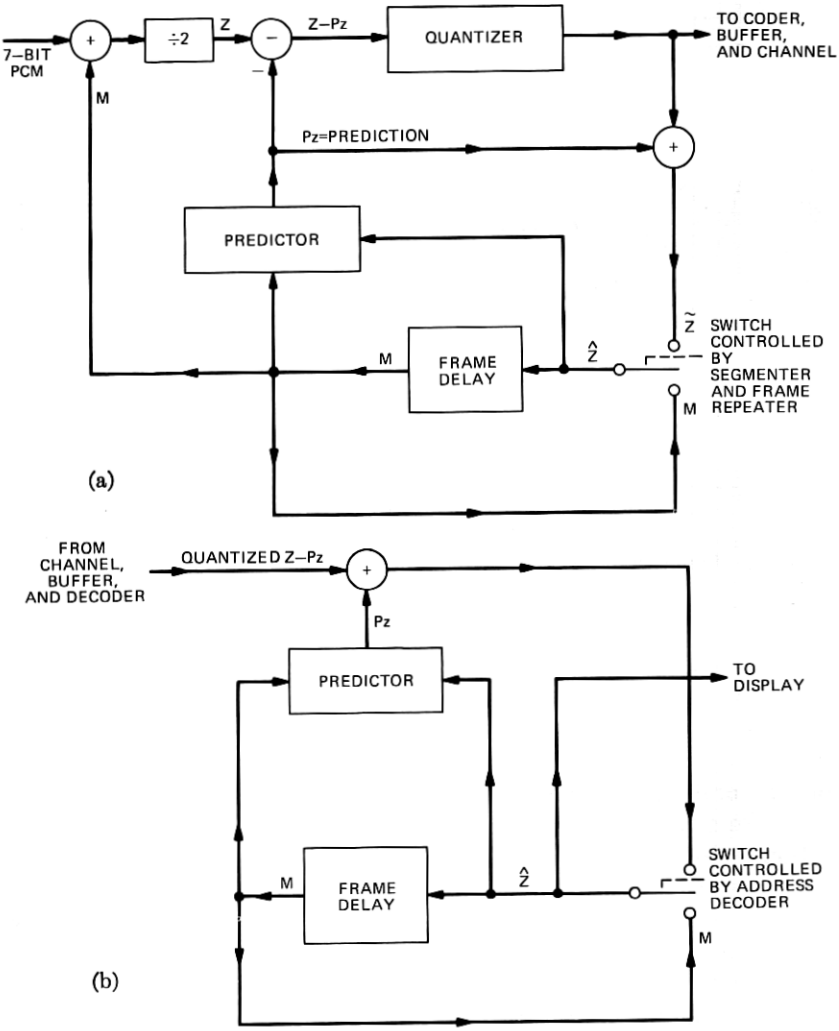


Fig. 4—Implementation of predictive coding with temporal filtering and frame repeating. (a) Transmitter. (b) Receiver. During alternate incoming frames, the switches are held in the down position, thus recirculating the contents of the frame memory, and no data are fed to the transmitter buffer.

power transfer characteristic of frame repeating plus temporal averaging (also derived in the appendix). It falls off much faster than curve b, and is down by about 10 dB at 7.5 Hz. However, unlike curve b it rises again at higher frequencies.

Temporal averaging and frame repeating as shown in Fig. 4 has been implemented, and jerkiness is difficult to detect. However, blurring is quite visible when the subject moves. Lip motion is also blurred somewhat.

Temporal filtering helps to reduce the data rate in two ways. First, as already mentioned, only every other television frame need be transmitted. Simple frame repeating at the receiver is sufficient to display a picture in which jerkiness is difficult to detect. Second, the blurring of the moving area makes the signal more amenable to predictive coding. With the blurred picture, the differential signal is smaller on the average, thus reducing its entropy and the bit rate required for transmission.

V. SEGMENTING, ADDRESSING, AND SYNCHRONIZING

The coder uses simple, well-known techniques for segmenting the picture into moving and stationary areas.[3] Ordinary seven-bit PCM requires updating of pels which have changed by 2 or more on a scale of $0 \cdots 127$ to present good picture quality in slowly moving areas. However, temporal filtering as shown in Fig. 4 amounts to halving all the frame differences. Thus, in the coder described here, frame differences larger in magnitude than 1 on a scale of $0 \cdots 127$ are detected and labeled as significant changes. As is described later, the frame-difference threshold is raised to 2 to reduce the data rate when buffer overflow threatens.

Significant changes because of camera noise are dealt with as in Ref. 3. That is, a change is ignored if the two pels on the left and the two pels on the right have not changed significantly.

Positioning information for the transmitted pels is also sent as in Ref. 3. The start of a cluster of significant changes is signaled by an eight-bit address indicating its position. The end of a cluster is specified by sending a four-bit code word which is distinguishable from the quantizer output code words (see Table I).

Small gaps between clusters are more efficiently handled by transmitting the pels therein than by ending one cluster and starting a new one.[3] This technique is called gap-bridging.

In Ref. 8, it was found that the entropy of the quantized line difference of frame-difference signal was somewhat above two bits per moving-area pel. Since each new cluster requires twelve bits for addressing, the coder bridges gaps of six pels or less prior to conditional replenishment.

Synchronizing is handled as follows. Since there are less than 256 visible pels along a line, frame sync, field sync, error detection words, and other events which occur relatively rarely can be signaled conveniently using eight-bit code words that are distinguishable from the eight-bit cluster addresses. However, line-to-line sync is not handled as easily.

If line sync were signaled with an eight-bit word, then, with the ≈8-kHz line rate used here, 64 kb/s would be devoted to line sync. For a coder operating at a few hundred kb/s, this is much too high a proportion of the total bit rate.

The method of line sync proposed for the coder requires slightly more than one bit per line. With frame repeating, this amounts to about 4 kb/s being used for line sync. The method relies on the fact that the first pel in the first cluster of a line is usually located to the left of the last pel of the last cluster of the previous line. In this case, no additional information need be transmitted to tell the receiver that a new line has begun. However, the receiver must be told if the above situation does not apply, and it must also be told which lines in the picture contain no clusters.

```
ooooooooooooooooooooooo
ABCDooooooooooooEFGH
ooooooooooooooooooooooo
ooooooooooooooooooooooo
ooIJKLooooMNPoooooooo
ooooooooooooooooooooooo
ooooooooooooooooooooQRS
```

Consider the field of pels shown above. Pels labeled A, B, C, · · ·, R, S have changed significantly and must be transmitted along with their cluster addresses. Pels labeled o will not be transmitted. Since the cluster ABCD is the first one in the field, the receiver need not be told that a new line is starting. It only needs to be told the number of lines at the beginning of the field that contains no clusters. A string of zeros equal in number to this amount followed by a one suffices to convey this information to the receiver. For implementation reasons which become apparent later, this string of bits is transmitted after the address word of cluster ABCD and before the pels A, B, C, D and the end-cluster message are sent. Cluster EFGH is sent in the normal manner, i.e., address, pels, and end-cluster.

Since pel I is to the left of pel H, the receiver can tell from the address of cluster IJKL that a new line has begun. Following the address word of cluster IJKL, the bits 001 are transmitted, indicating that two intervening lines contained no clusters. Cluster MNP is sent in the normal manner.

Since pel Q occurs to the right of pel P, the receiver cannot tell from the address of cluster QRS that a new line has begun. A special reserved address word must be transmitted to indicate a new line. Following this, the address of cluster QRS and the bits 01 are transmitted as usual. If small gaps between clusters are bridged, then the above procedure should be modified somewhat. In this case, the special reserved address word need be transmitted only if

$$\text{Q-address} > \text{P-address} + \text{minimum gap size.}$$

A system using these ideas would operate sequentially as follows:

(*i*) At the start of each field
  (*a*) A field sync word is transmitted.
  (*b*) An address register is set to maximum value.
  (*c*) A counter is reset to zero.

(*ii*) The counter is incremented by 1 at the end of each line which contains no clusters to be transmitted.

(*iii*) When the first cluster of a line is encountered
  (*a*) A check is made to see if the address of the first pel exceeds that in the address register. If it does, a special reserved eight-bit word is transmitted which is distinguishable from all the normal cluster address words. This should not occur very often when movement is significant.
  (*b*) The address of the cluster is transmitted.
  (*c*) A string of zeros is transmitted equal in number to the value stored in the counter. None are sent if the counter equals zero.
  (*d*) A one is transmitted, and the counter is reset to zero.

(*iv*) Normal conditional replenishment then resumes and continues until the end of the line.

(*v*) The address of the last pel of the last cluster of the line is added to the minimum gap size, and the result is stored in the address register.

(*vi*) Operation continues with Step (*ii*).

This technique was tested, and with scenes containing slow, moderate, or rapid movement the number of special words that had to be transmitted rarely exceeded two per field ($\approx$ 0.5 kb/s when frame repeating is employed). With no movement, the clusters of significant changes resulting from noise occurred randomly, and the number of special words was higher. But in this case the overall data rate is very small, and thus the special words do not overload the coder.

VI. MODE CONTROL

For a given transmission bit rate, a higher overall picture quality can be obtained if the coding is adapted to the amount of movement in the scene. For an interframe coder, the fullness of the transmitter buffer is the simplest and most useful measure of the amount of movement.[1–3] Imminent buffer overflow is a direct indication that the data rate being generated is too high and that the displayed moving area resolution should be reduced.

The basic operating mode of the low bit-rate coder is shown in Fig. 4, i.e., temporal filtering, frame repeating, and transmission of line differences of frame differences in the moving area. As with previous coders operating at higher bit rates,[2,3,5–7] the moving-area resolution is reduced by switching to a lower resolution mode if the buffer queue length exceeds some fixed threshold. Thus, as shown in Fig. 5, if the buffer queue length exceeds T_1, then coding mode 1 is invoked; if it exceeds T_2, then coding mode 2 is invoked; etc. Mode 4 is frame repeating, i.e., the switch in Fig. 4 is held in the down position, and no data are generated except synchronizing information. In this way, buffer overflow is prevented.

When motion in the scene ceases and the size of the moving area decreases, the buffer begins to empty, and a higher-resolution coding mode should be used. To prevent oscillations between coding modes, a higher-resolution mode is not invoked until the end of a field, and then only if the buffer queue length is below T_1 for modes 1 and 2 and T_2 for modes 3 and 4. Thus, for example, a change from mode 1 to mode 2 is possible any time the buffer queue length exceeds T_2, but a change from mode 2 to mode 1 can occur only at the end of a field in which the buffer queue length falls below T_1.

VII. MODES USED IN THE CODER

Mode 0 is the previously mentioned basic operating mode shown in Fig. 4. An odd field and an even field are coded as shown in Fig. 6a. Then the next two fields are skipped; at the receiver, the frame is repeated by displaying the stored signal. Mode 0 is the highest resolution mode of the coder.

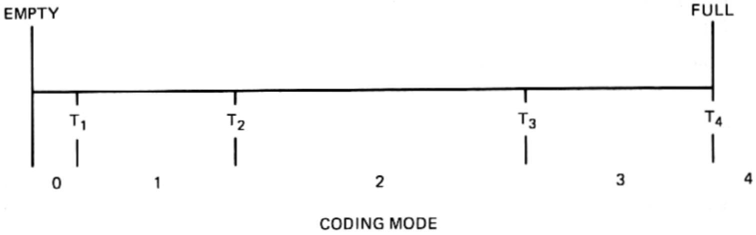
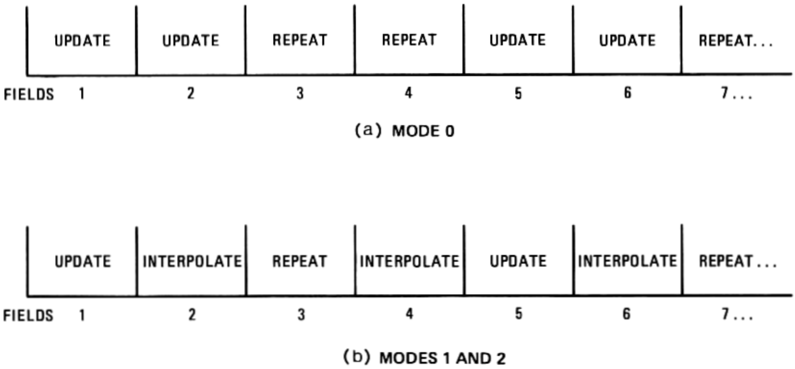


Fig. 5—Switching between coding modes under control of the transmitter buffer causes the moving area resolution to be reduced as the amount of motion in the scene increases. Mode 0 codes with the highest resolution, mode 4 with the lowest. When the buffer queue length exceeds T_i, mode i is invoked (except for mode 3 which is invoked at the end of the field). At the end of the field in which the buffer queue length falls below T_1 for modes 1 and 2 or T_2 for modes 3 and 4, mode i is revoked and mode $i-1$ is invoked. With this strategy, oscillations between coding modes are prevented.



(a) MODE 0

(b) MODES 1 AND 2

Fig. 6—(a) Simple frame repeating is used in mode 0. Two fields are updated, then two fields are repeated. (b) Frame repeating and field interpolation are used in modes 1 and 2. Only one out of four fields is updated. No data are generated for the remaining three fields.

Mode 1 is interpolation of even fields. In this mode, the data rate is halved by not transmitting even-numbered fields as shown in Fig. 6b. Instead, an interpolation between the previous odd field and the upcoming odd field[10] is displayed, thus reducing the vertical resolution in the picture by a factor of two.

Field interpolation is implemented as shown in Fig. 7. If, during input of an even field, mode 1 is invoked, then the conditional re-

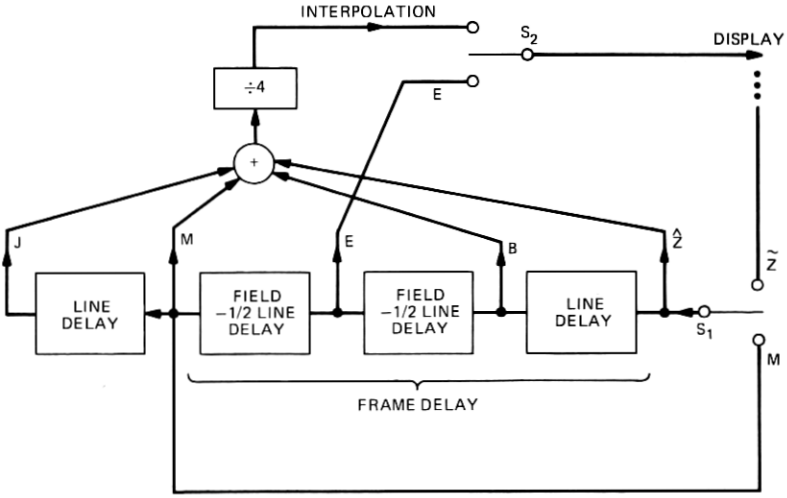


Fig. 7—Implementation of field interpolation. S_1 is held in the down position during input of repeated fields and interpolated fields. No data are generated for them. One field period later, S_2 is put in the up position to display interpolated fields and in the down position to display updated and repeated fields (see Fig. 1).

plenishment switch S_1 is held in the down position for the remainder of the field, and no updating occurs. During input of the next two successive odd fields, switch 2 is held in the up position to display interpolated values for the even fields. Otherwise, it is in the down position. Display of two interpolated fields is necessary because of the aforementioned frame repeating which would otherwise display the invalid contents of the frame memory.

Mode 2 consists of the field interpolation of mode 1 plus 2:1 horizontal subsampling,[3,4] i.e., only every other moving-area pel along a line is transmitted. The untransmitted pels are obtained from their neighbors by interpolation. Subsampling reduces the data rate by a factor of almost 2 over mode 1.

Mode 2 also employs coarser quantization of the line difference of frame-difference signal and an increase of the frame-difference threshold used by the segmenter. When mode 2 is invoked, the frame-difference threshold is raised from 1 to 2 on a scale of $0 \cdots 127$, and the two smallest nonzero levels of the quantizer are switched out of operation, reducing the number of levels to nine. The outputs of the nine-level quantizer are coded using the first nine code words of Table I. Coarser quantization reduces the entropy of the differential signal, and raising the frame-difference threshold reduces the number of pels that must be transmitted. Together they reduce the data rate by a factor of about 1.5, but this figure depends very much on the picture material and on the amount of movement in the scene.

Mode 3 is frame repeating at the end of a field. When mode 3 is invoked, all conditional replenishment is halted. The contents of the frame memory are displayed for odd-numbered fields, and interpolated values are displayed for even-numbered fields. But unlike the other modes, it is invoked only at the end of a field. The purpose of this is to avoid the picture breakup associated with the stopping of conditional replenishment in the middle of a field. As the amount of motion in the scene increases, mode 3 causes the coder to progressively operate in 4:1 frame repeating, 6:1 frame repeating, or as much as is necessary

Table II — Modes of the coder with mode 0 the highest resolution mode

Mode	
0	Temporal filtering and frame repeating.
1	Mode 0 plus interpolation of even fields.
2	Mode 1 plus 2:1 horizontal subsampling, increased frame-difference threshold, and coarser quantization.
3	Frame repeat at end of field.
4	Instantaneous frame repeat.

to accommodate the rate of data generation. Mode 3 is revoked at the end of the field during which the buffer queue length falls below T_2 (not T_1, as with modes 1 and 2).

Mode 4 is instantaneous frame repeating. It is rarely used and is invoked only to prevent data from being lost in an uncontrolled manner because of buffer overflow. It is revoked at the end of the field, and normal frame repeating under mode 3 then resumes. The modes of the coder are summarized in Table II.

VIII. CHOICE OF BUFFER QUEUE LENGTH THRESHOLDS

The objective of the coder is to operate in the mode that best matches the data generation rate with the channel transmission rate. Also, oscillation between modes must be avoided since it adversely affects picture quality in some cases. Correct choice of the buffer queue length thresholds is very important in accomplishing these objectives. As an example, the following illustrates how the thresholds might be chosen for a 200-kb/s channel rate.

Mode 0 is used only when there is little or no motion in the scene. Its most important function occurs just after motion in the scene has ceased and mode 1 (interpolation of even fields) has been revoked. The objective is to update the even field and restore full vertical resolution as quickly as possible. Shortly after even field update has begun, the buffer queue length will exceed T_1 and updating will cease. Little or no data will be produced for the remainder of the field and for the next three field periods. If during this time the buffer empties, then transmission time will have been wasted. Thus, T_1 should be chosen large enough so that the buffer cannot empty in four field periods (1/15 second). For 200 kb/s, $T_1 > 13333$ bits.*

During mode 1, data are produced in only one field out of four (see Fig. 6b). If the overall data generation rate happens to equal the channel transmission rate, then the coder should not produce any data if it should switch to mode 0, and it should not switch to mode 2. In Fig. 6 at the end of field 1 coded in mode 1, the buffer queue length will exceed T_1, and thus field 2 will be interpolated, field 3 will be repeated, and field 4 will be interpolated even if the coder drops into mode 0. To prevent mode 2 from being switched in during a mode 1 odd-field update, T_2 must be large enough to accommodate the accumulated difference between the data generation rate and the channel transmission rate. Somewhat more than three field periods of channel data may have to be buffered. Thus, for 200 kb/s, $T_2 > T_1 + 10000$ bits.

* This figure can be halved if, in mode 0, fields 4, 8, $\cdots$ (see Fig. 6) are updated instead of repeated.

With mode 2, the same sort of argument applies. Switching to adjacent modes when the data generation rate and the channel transmission are well matched can be prevented by separating the thresholds by more than three field periods of channel data. Thus, T_2 and T_3 should be more than 10000 bits larger than the next lower threshold.

Mode 3 (4:1, 6:1, ···frame repeating) is invoked at the *end* of a field in which the buffer queue length exceeds T_3. It is revoked at the end of a field in which the buffer queue length falls below T_2. If the buffer queue length exceeds T_3 at the end of field 1 in Fig. 6b, then 4:1 frame repeating will occur if the buffer queue length is still above T_2 at the end of field 4. This can be guaranteed by choosing $T_3 - T_2$ larger than three field periods of channel data. Thus, for 200 kb/s, $T_3 > T_2 + 10000$ bits. If $T_4 - T_3$ exceeds four field periods of channel data, then normal 6:1 frame repeating with no picture breakup can occur; if it exceeds eight field periods of channel data, then normal 8:1 frame repeating can occur; etc.

T_4 also determines the transmission delay owing to buffering. If 300 ms is the maximum tolerable one-way delay,[1] then T_4 must not exceed 0.3 × channel rate. For 200 kb/s, $T_4 < 60000$ bits.

Table III gives suitable buffer queue length threshold values for 200-kb/s and 50-kb/s operation. The distances $(T_2 - T_1)$ and $(T_4 - T_3)$ can be reduced somewhat without seriously affecting coder operation, and other more complex mode control strategies can probably be devised that do not require as much buffering. But for purposes of assessing the possible trade-offs between picture quality and channel bit rate, these settings are a valid compromise.

IX. DIGITAL TRANSMISSION ERRORS

It seems to be a general rule that the more the redundancy in a stream of information is reduced, the more vital the remaining information becomes. This is especially true for a low bit-rate interframe coder for television signals. Errors in the data which arrive at the receiver will usually cause discrepancies in its frame memory of which the transmitter is unaware. Thus, if no means are provided to ac-

Table III — Buffer queue length thresholds for 200 and 50 kb/s

Threshold	200-kb/s Operation	50-kb/s Operation
T_1	13400	3350
T_2	24900	6230
T_3	45000	11250
T_4	60000	15000

commodate for digital transmission errors, they will cause visible picture degradations that will last forever.

Many schemes for handling errors have been suggested.[1,3] A simple method, called forced updating, is to transmit a portion of the data as PCM. After a period of time, data in the receiver frame memory will be corrected and visible errors will disappear. However, using channel bit rates which are relatively low compared with PCM, the time required for correction can be quite long. For example, if 10 percent of the channel data is devoted to PCM, then at 200 kb/s all errors in the picture can be corrected in about 22 seconds. But with one error in 10^6 bits, for example, the average time between errors is 5 seconds. Thus, with this technique, errors are always present and, with differential coding of the type discussed previously, highly visible. Other error control techniques are obviously necessary.

Randomly occurring, isolated errors and occasional bursts of errors can be dealt with fairly easily by lowering the information bit rate and using forward-acting error correction codes. However, long bursts of errors in the bit stream present much more of a problem. Cluster addressing, variable word-length coding, and DPCM all serve to increase the vulnerability of the system to digital transmission errors. A long burst of errors would, in most instances, cause picture breakup for many seconds, until some updating procedure could restore the receiver frame memory to its proper state.

Although long bursts of digital transmission errors cannot easily be corrected, they can, in most cases, be detected fairly easily. The receiver could then switch to a frame repeat mode during that portion of the picture for which the frame memory is known to be in error, thus avoiding the picture breakup associated with free running operation. With no movement in the scene, errors would not affect picture quality. With movement, however, errors could, for example, cause the lower half of the picture to freeze for several seconds until it could be updated via PCM.

Recovery from transmission errors can be speeded up considerably if the transmitter can be made aware of their existence and general location by feedback from the receiver. The transmitter could then simply zero out the offending portion of its frame memory and send a control signal telling the receiver to do the same. The erroneous portion of the receiver frame memory will then be updated automatically.

Somewhat more complicated schemes can be devised that utilize feedback of error status and retransmission of incorrectly received data blocks. Some extra buffering is needed (the required amount depends on the channel delay), but erroneous data will not enter the receiver frame memory except on rare occasions when the error detection algorithm fails. With these techniques, periods of very noisy transmission simply cause the transmitter buffer to fill, which automatically invokes lower-resolution coding modes or frame repeating to match the rate of data generation with the currently available channel capacity.

X. SIMULATION OF THE CODER

A simulation of the coder was constructed to observe what the picture quality would be in an actual system in the absence of digital transmission errors. By and large, the simulator performed all the coding operations that would significantly affect picture quality; however, many shortcuts were taken.

Synchronization of the camera, PCM coder, and simulator was maintained through the same 2-MHz clock; thus, phase-locking and stability problems were sidestepped. Peak-signal-to-rms noise ratio of the input video signal was above 40 dB; thus, problems of analog transmission to the coder were not considered.

The field delays were obtained using a core memory configured as a tapped delay line. Most of the other circuitry was TTL or MOS. A buffer was not constructed. Instead, an up-down counter and threshold detection logic was used to implement the mode control features previously discussed. This approach also made construction of a variable word-length coder unnecessary, although presently available inexpensive solid-state ROMs make this a fairly easy task. The display was obtained by incorporating the field interpolation circuitry of Fig. 7 into the simulator. Normally, this logic is required only at the receiver.

For scenes of people engaged in conversation, it was necessary to delay the voice signal by about 100 ms to obtain a match with the moving lips. Most of this delay is due to the temporal filtering discussed previously. The remainder is due to the field delay between input and display (see Fig. 7). In fact, a completely satisfying match between voice and lips is not obtainable because of the blurring of moving areas caused by the temporal filtering.

XI. CONCLUSION

In this paper, a frame-to-frame coder for videotelephone signals is described that operates at a relatively low bit rate compared with previous coders (200 kb/s or 0.1 bit per pel for an original signal of 1 MHz). The coder was designed on the assumption that faithful rendition of large moving areas in a scene is not essential for effective interactive visual communication to take place. Whether or not this assumption holds in the majority of situations remains to be seen, but it is conceivable that if users are made aware of the considerable economic saving involved, they will put up with a certain amount of visible distortion in the display.

Graphics and scenes containing little or no movement are portrayed without degradation. Low-key face-to-face conversations contain detectable blurring of moving areas, but for many users this may not be highly objectionable. However, large moving areas are very visibly blurred, sometimes to the point of being nonrecognizable.

The one-way transmission delay of the coder is comparable to the nominally acceptable figure of 300 ms. If the delay of the digital transmission channel is also significant compared with this, as it would be, for example, on an earth satellite circuit, then interactive communication will be severely hampered. Also, special measures must be taken to deal with digital transmission errors. The data generated by the coder are in highly sensitive form. Thus, if some of them arrive incorrectly at the receiver, precautions must be taken to ensure that they do not corrupt legitimate information which has already been received.

The techniques described here apply also to higher resolution pictures, e.g., 525-line standard broadcast rate signals. Indeed, since moving areas do not require any more resolution than with videotelephone, the channel bit rate should not be very much higher either. Graphics and scenes containing no movement would be displayed with much higher resolution. However, the coder itself would also be more expensive.

Much work remains to be done before it will be known if the techniques described here are useful in providing an acceptable compromise between slow-scan facsimile transmission and full rendition of scenes containing movement. Coding for redundancy reduction will remain practical only if costs of logic and storage fall faster than costs of transmission. Also, the requirements of future visual communication systems may change drastically after users begin to learn how to use them effectively in their day-to-day lives.

XII. ACKNOWLEDGMENTS

We are grateful for the many suggestions and criticisms as well as for the technical support we have received from D. J. Connor, J. C. Candy, K. A. Walsh, and our colleagues in the Visual Communication Research Department.

APPENDIX

Here, analytical expressions are given for the power transfer P versus temporal frequency characteristics of Fig. 3. Let $z(t)$ be the light intensity falling on a point of the television camera target, $x(t)$ the output signal as that point is read out of the camera, and $y(t)$ the temporally filtered signal. T is the time between normal frames, i.e., 1/30 second.

Fig. 3a— Camera integration only.

$$x(t) = \frac{1}{T}\int_{t-T}^{t} z(s)ds \tag{1}$$

$$= \frac{1}{T}\int_{-\infty}^{t} z(s)ds - \frac{1}{T}\int_{-\infty}^{t-T} z(s)ds. \tag{2}$$

Taking Fourier transforms,

$$X(\omega) = \frac{1}{j\omega T} Z(\omega) - \frac{1}{j\omega T} Z(\omega)e^{-j\omega T} \tag{3}$$

$$P_a = \left|\frac{X(\omega)}{Z(\omega)}\right|^2 = \frac{2}{\omega^2 T^2}(1 - \cos \omega T). \tag{4}$$

Fig. 3b—Temporal averaging.

$$y(t) = \tfrac{1}{2}x(t) + \tfrac{1}{2}y(t - T). \tag{5}$$

Taking Fourier transforms,

$$Y(\omega) = \tfrac{1}{2}X(\omega) + \tfrac{1}{2}Y(\omega)e^{-j\omega T} \tag{6}$$

$$\left|\frac{Y(\omega)}{X(\omega)}\right|^2 = \frac{1}{(5 - 4 \cos \omega T)} \tag{7}$$

$$P_b = \left|\frac{Y(\omega)}{Z(\omega)}\right|^2 = \frac{2(1 - \cos \omega T)}{\omega^2 T^2 (5 - 4 \cos \omega T)}. \tag{8}$$

Fig. 3c—Temporal averaging and frame repeating.

$$y(t) = \tfrac{1}{2}x(t) + \tfrac{1}{2}y(t - 2T). \tag{9}$$

From (7),

$$\left|\frac{Y(\omega)}{X(\omega)}\right|^2 = \frac{1}{(5 - 4 \cos 2\omega T)} \tag{10}$$

$$P_c = \left|\frac{Y(\omega)}{Z(\omega)}\right|^2 = \frac{2(1 - \cos \omega T)}{\omega^2 T^2 (5 - 4 \cos 2\omega T)}. \tag{11}$$

REFERENCES

1. 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.
2. F. W. Mounts, "A Video Encoding System Using Conditional Picture-Element Replenishment," B.S.T.J., *48*, No. 7 (September 1969), pp. 2545–2554.
3. J. C. Candy, M. A. Franke, B. G. Haskell, and F. W. Mounts, "Transmitting Television as Clusters of Frame-to-Frame Differences," B.S.T.J., *50*, No. 6 (July–August 1971), pp. 1889–1917.
4. R. F. W. Pease and J. O. Limb, "Exchange of Spatial and Temporal Resolution in Television Coding," B.S.T.J., *50*, No. 1 (January 1971), pp. 191–200.
5. J. B. Millard, Y. C. Ching, and D. M. Henderson, private communication.
6. D. J. Connor, B. G. Haskell, and F. W. Mounts, "A Frame-to-Frame *Picturephone*® Coder for Signals Containing Differential Quantizing Noise," B.S.T.J., *52*, No. 1 (January 1973), pp. 35–51.
7. 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 (July–August 1974), pp. 1137–1173.
8. B. G. Haskell, "Entropy Measurements for Nonadaptive and Adaptive Frame-to-Frame Linear Predictive Coding of Videotelephone Signals," unpublished work.
9. D. J. Connor, private communication.
10. J. O. Limb and R. F. W. Pease, "A Simple Interframe Coder for Video Telephony," B.S.T.J., *50*, No. 6 (July–August 1971), pp. 1877–1888.