Copyright © 1979 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 58, No. 7, September 1979
Printed in U.S.A.

Motion-Compensated Transform Coding

By A. N. NETRAVALI and J. A. STULLER

(Manuscript received April 10, 1979)

Interframe hybrid transform/DPCM coders encode television signals by taking a spatial transform of a block of picture elements in a frame and predictively coding the resulting coefficients using the corresponding coefficients of the spatial block at the same location in the previous frame. These coders can be made more efficient for scenes containing objects in translational motion by first estimating the translational displacement of objects and then using coefficients of a spatially displaced block in the previous frame for prediction. This paper presents simulation results for such motion-compensated transform coders using two algorithms for estimating displacements. The first algorithm, which is developed in a companion paper, recursively estimates the displacements from the previously transmitted transform coefficients, thereby eliminating the need to transmit the displacement estimates. The second algorithm, due to Limb and Murphy, estimates displacements by taking ratios of accummulated frame difference and spatial difference signals in a block. In this scheme, the displacement estimates are transmitted to the receiver. Computer simulations on two typical real-life sequences of frames show that motion-compensated coefficient prediction results in coder bit rates that are 20 to 40 percent lower than conventional interframe transform coders using "frame difference of coefficients." Comparisons of bit rates for approximately the same picture quality show that the two methods of displacement estimation are quite similar in performance with a slight preference for the scheme with recursive displacement estimation.

I. INTRODUCTION

Television signals, which are generated by scanning a scene 30 times a second, contain a significant amount of frame-to-frame redundancy. A large part of this redundancy can be removed by the technique of conditional replenishment.[1-5] In conditional replenishment, each frame is segmented into two parts: background, which consists of picture elements (pels) having intensities similar to the previous frame pels, and moving area, which consists of pels that differ significantly from the previous frame pels. Information is transmitted only about the moving area in the form of prediction errors and addresses of the moving area pels. Conditional replenishment schemes can be improved by estimating the displacement of objects in the scene and using the displacement estimate for predictive coding by taking differences of elements in the moving area with respect to appropriately displaced elements in the previous frame. Such schemes have been referred to as motion-compensated coding schemes.[6–11]

Transform domain methods have been widely discussed for bandwidth compression of still images or single frames.[12] They can also be used for coding of sequences of television frames by taking a two-dimensional spatial transform followed by predictive coding using corresponding coefficients from the spatial transform of the previous frame.[13–16] This type of hybrid coding[17] relieves the storage problems associated with the use of three-dimensional transform blocks. Such a scheme can be made more efficient for scenes containing objects in motion by using, for prediction, coefficients of blocks from the previous frame that are spatially displaced from the present frame block by an amount equal to the displacement of objects. As in the pel domain, the success of motion compensation in transform coders depends upon: (*i*) the amount of purely translational motion of objects in the scene, (*ii*) the ability of the displacement estimation algorithm to estimate the translation with an accuracy necessary for good prediction of the coefficients, and (*iii*) the robustness of the displacement estimation algorithm when the resolution of the transmitted picture is changed to match the coder bit rate to the channel rate.

In this paper, we use two previously published displacement estimation algorithms for motion-compensated transform coding. The first algorithm is an extension of a corresponding method in the pel domain.[10,11] It works recursively on the previously transmitted transform coefficients of the present as well as the previous frame. It therefore requires no separate transmission of the displacement estimate. This algorithm is discussed in detail in a companion paper,[18] where its properties are described both analytically and experimentally in terms of certain simple synthetically generated scenes. The other method of displacement estimation that we use is due to Limb and Murphy.[19] It estimates displacements in a block of pels using a ratio of accummulated frame difference and spatial difference signals from future as well as past data. These displacement estimates are nonrecursive and must be transmitted separately to the receiver. The present paper investigates the performance of the two displacement estimation algorithms in the context of interframe coders operating on real-life scenes that contain fairly complex (nontranslational) motion. Results are given here on the effects of various coder parameters such as block size, particular transform (Hadamard, cosine, etc.), and other parameters of the displacement estimators. The primary result of this paper is that the application of either recursive or nonrecursive motion estimation provides a 20 to 40 percent decrease in bit rate, compared to conventional, uncompensated hybrid transform/DPCM coding. We have found that the use of large block sizes in motion estimation degrades the coder performance. This may be a result of spatially nonuniform displacements being averaged over the transform block by the displacement estimator. Also, since the motion in real scenes is generally not uniform in rectangular blocks, as the block size is increased, only a fraction of elements in a block are compensable with a given displacement, and therefore transmitting coefficients of a larger block containing some compensable and some uncompensable pels becomes inefficient.

II. HYBRID TRANSFORM CODING WITHOUT MOTION COMPENSATION

In an interframe hybrid transform-DPCM coder, a field of video is partitioned into blocks having dimensions N_r rows by N_c columns, and a two-dimensional transform is performed on each block to obtain a set of coefficients. Transform coefficients of the qth block of the present frame are predicted by the corresponding coefficients of the qth block of the previously encoded frame, and, if the prediction error is above a specified threshold, the quantized prediction errors are transmitted to the receiver. These quantized errors are added to the coefficients predicted by the receiver, which inversely transforms the result to obtain an image for display at the receiver. A block diagram of an interframe hybrid transform-DPCM transmitter is shown in Fig. 1. Data compression is achieved both by the redundancy removal implicit in the prediction process and because some coefficients can be reproduced with low precision (or totally omitted from transmission) without visible degradation in the reconstructed picture.

The performance of the interframe hybrid transform-DPCM coder and the other coders described in later sections of this paper is evaluated in terms of bit rate for an acceptable subjective picture quality using two scenes, one called Judy and one Mike and Nadine. The coding degradation was judged in informal tests by the authors to be just perceptible from a viewing distance of six times the picture height. These scenes consist of 64 frames (2:1 interlaced fields) of 256 $\times$ 256 samples each, obtained at 30 times a second and sampled at Nyquist rate from a video signal of 1-MHz bandwidth. The scene Judy contains head-and-shoulders views of a person engaged in a rather

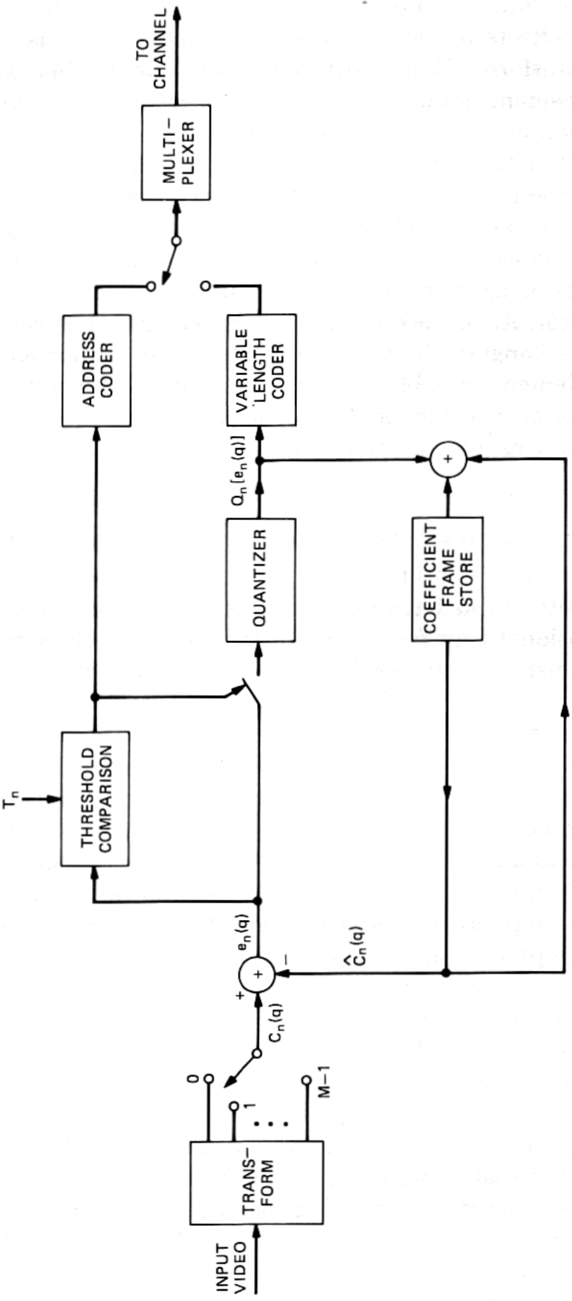

Fig. 1—Block diagram of hybrid transform/DPCM interframe image encoder.

active conversation. The portion of a frame classified as moving area varies from 15 to 51 percent. The motion is not strictly translational, and there are different parts of the scene moving differently (such as lips, eyes, and head). Four frames of this scene are shown in Fig. 4 of Ref. 10. The scene Mike and Nadine contains a panned full-body view of two people briskly walking around each other on a set with severe nonuniform and time-varying illumination. The percentage of a frame classified as moving area varied from 92 to 96 percent. Four frames from this sequence are shown in Fig. 5 of Ref. 10.

In our simulations of the interframe hybrid transform-DPCM (called conditional replenishment in the transform domain), the coefficients of two corresponding spatial blocks of the same field from two successive frames are compared, and if the difference is more than a threshold, the coefficient is transmitted. Thus, if $\{c_k\}_{k=0,\dots,M-1,}$ and $\{\bar{c}_k\}_{,k=0,\dots,M-1}$ are M selected coefficients (out of N coefficients in a block) of the present and coded previous frame blocks, respectively, then the quantized error, $Q_k[c_k - \bar{c}_k]$, is transmitted only if $|c_k - \bar{c}_k| \geq T_k$, where $Q_k[\cdot]$ is the quantizer for the kth coefficient, and T_k is the threshold. If c_k is not transmitted, then its value at the receiver is assumed to be $\bar{c}_k$. Thus the transmission consists of the quantized prediction error of the coefficients that were selected for transmission and the addresses of the coefficients that were dropped from the transmission. The information necessary to convey addresses of the coefficients selected for transmission was computed based on the run-length coding of runs of coefficients within a block and then from block to block. Parameters of the coder such as the number of coefficients that were entirely dropped from the transmission, the thresholds $\{T_k\}$ for selecting the transmitted coefficients, and the quantizer scales were adjusted* to produce pictures in which coding degradations were just perceptible. The entropies of the prediction errors and the run lengths specifying addresses of the transmitted coefficients are added to compute the total bit rate.

The results are shown in Fig. 2, in which the bit rate is plotted as a function of the frame number for 60 frames. In these simulations and those of the next section, the coder was initialized so that it used the unquantized original first frame for prediction of the second frame. For comparison, the results from Ref. 10 are reproduced for conditional replenishment in pel domain. The comparison shows that, in the transform domain, using a cosine transform on a 2×4 block, there is a reduction of about 10 percent in bit rate over that obtained in pel

* We do not claim that these adjustments resulted in an optimum set of parameters. However, a sufficiently large set of parameters was tried, giving us confidence that our results are not far from the optimum.

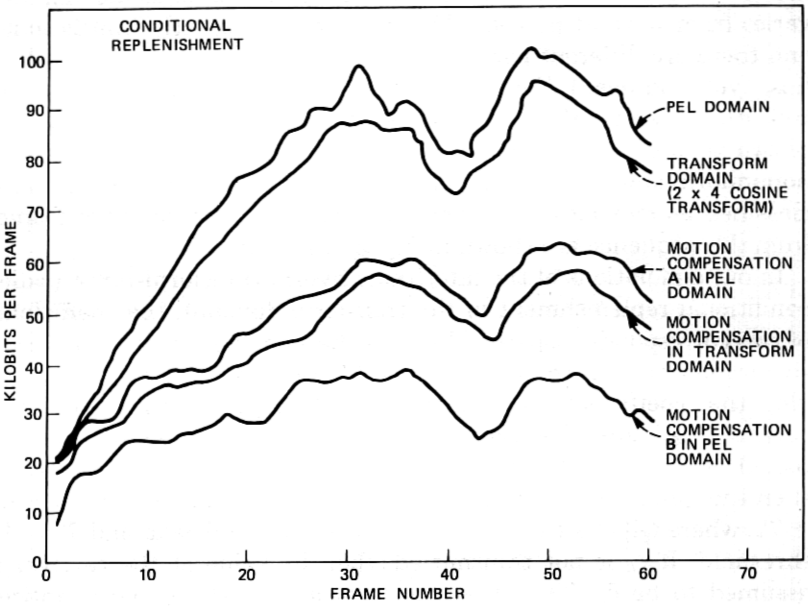

Fig. 2—Performance of conditional replenishment and motion-compensated transform coders. Kilobits/frame are plotted as a function of the frame number for a typical sequence containing active motion of a head-and-shoulders view.

domain.* For this particular case, we dropped the eighth coefficient entirely, and the prediction errors for the other seven coefficients were quantized with uniform quantization scales with step sizes of 3, 5, 5, 7, 7, 9, 9, respectively. The thresholds $\{T_k\}$ for predictability were chosen to be 1, 2, 2, 3, 3, 4, 4 (out of 255) for the seven coefficients, respectively.

We varied some parameters of the transform to evaluate the sensitivity of these results to the block size and the type of transform used. Some of these are shown in Fig. 3. It is seen from this figure that a one-dimensional cosine transform with four elements did worse than the conditional replenishment in the pel domain (between 5 to 10 percent). As the transform size was increased, the bit rate dropped; for 2×2 block and cosine transform, the results were similar to the conditional replenishment in pel domain; the 2×8 block using the cosine transform, on the other hand, did about 15 percent better than the conditional replenishment in pel domain. We also tried different transforms and found that for small block sizes they were equivalent to the cosine transform but, as the block size was increased, the cosine

* Of course, several other modifications can be made to improve the pel domain conditional replenishment. Our comparison is not meant to be a comparison between pel domain and transform domain coding in general.

transform behaved better than the other transforms. The results for the 2×8 block using the Hadamard transform basis were very similar to those of 2×4 block and cosine transform but were inferior to those of the 2×8 block and cosine transform.

Figure 4 shows the distribution of the bits required for addressing and for the transmission of the first coefficient. As is seen, the addressing bits are about 50 percent of the total bits. This is a significant increase in addressing requirement compared to the conditional replenishment in the pel domain, where the addressing accounts for only about 20 to 30 percent of the total bits. This may be a result of using only the prediction error corresponding to the coefficient being coded for deciding whether that coefficient should be transmitted. This may have made the decision to transmit a coefficient unnecessarily noisy. We did, however, try several methods of reducing the addressing bits, but none of these resulted in an overall bit rate reduction. In the fraction of the bits that are required to send the prediction errors, those for the first coefficient account for more than half, as shown in Fig. 4. Thus the addressing and the first coefficient take up around 80 percent of the total bits generated by the conditional replenishment coder.

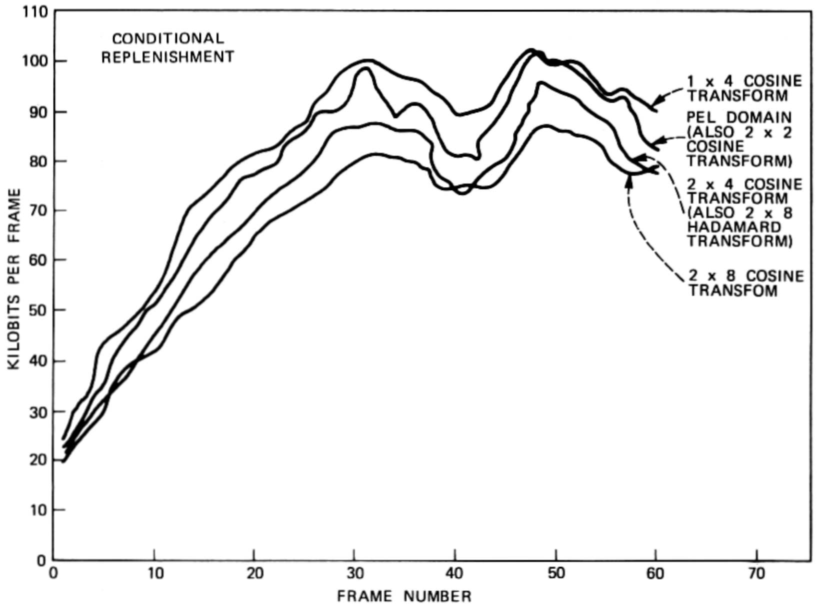

Fig. 3—Performance of conditional replenishment in the transform domain with various transforms.

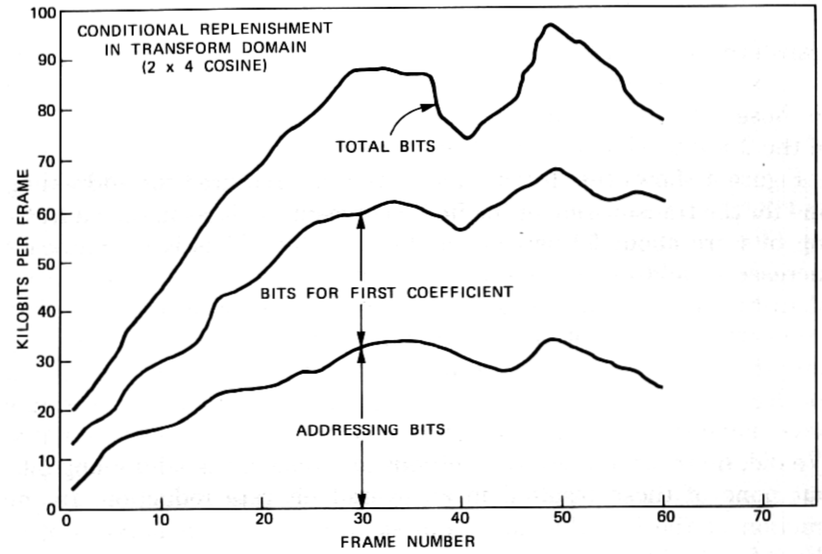

Fig. 4—Distribution of total compressed bits/frame into addressing information and that required for the transmission of the first coefficient.

III. MOTION COMPENSATION WITH RECURSIVE DISPLACEMENT ESTIMATION

In the motion-compensated hybrid transform-DPCM coder shown in Fig. 5, the nth coefficient of the qth present field block is predicted by the nth coefficient of either the displaced or the nondisplaced block of the previous frame, depending on which was better for the $(n - 1)$th coefficient, where the displacement is an estimate of the frame-to-frame translation of a moving object. The displacement estimation technique used in this section is identical to the one given in our companion paper.[18] We describe it as follows: Let $\mathbf{x}_q = (x_{1q}, x_{2q})^T$ denote the coordinate of the upper left-hand pel of the qth block, where the blocks in each row of blocks are numbered from left to right with $q = 0, 1, 2, \cdots$, and superscript T denotes the transpose of a vector or matrix. The pel intensities of block q in a column-scanning fashion are denoted by a column vector $\mathbf{I}(\mathbf{x}_q, t)$. Let the nth basis vector of the transform be denoted by $\boldsymbol{\phi}_n$, and, therefore, the nth coefficient of the qth block of the transform of the present frame can be written as

$$c_n(q) = \mathbf{I}^T(\mathbf{x}_q, t)\boldsymbol{\phi}_n. \tag{1}$$

The displaced previous frame value of this coefficient is

$$\hat{c}_n(q, \hat{\mathbf{D}}) = \mathbf{I}^T(\mathbf{x}_q - \hat{\mathbf{D}}, t - \tau)\boldsymbol{\phi}_n, \tag{2}$$

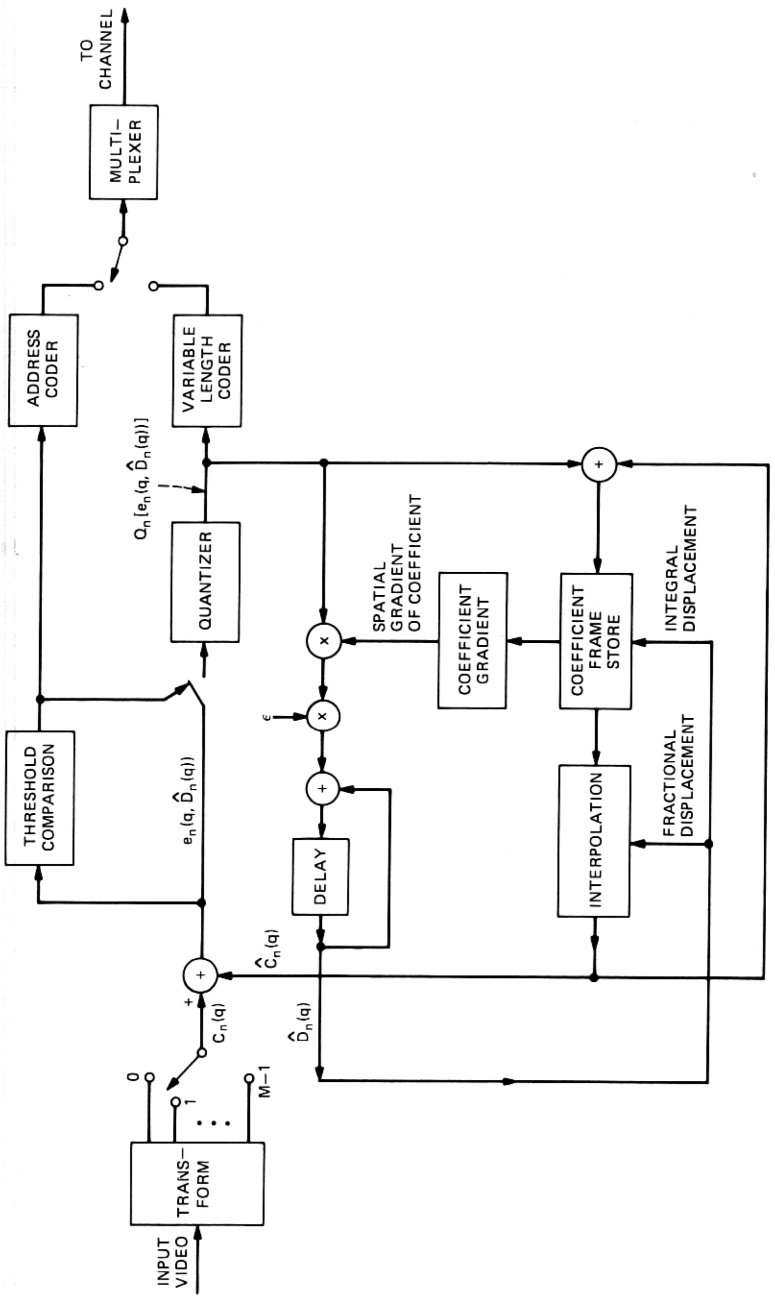

Fig. 5—Block diagram of motion-compensated transform/DPCM coder.

where $\mathbf{I}(\mathbf{x}_q - \hat{\mathbf{D}}, t - \tau)$ is the column vector of intensities of the displaced qth block of the previous frame and $\hat{\mathbf{D}}$ is the estimated displacement of the moving object. Computation of the elements in $\mathbf{I}(\mathbf{x}_q - \hat{\mathbf{D}}, t - \tau)$ generally requires an interpolation among the given previous-frame pel intensities. The displacement estimation algorithm attempts to minimize the prediction error in predicting $c_n(q)$ by $\hat{c}_n(q, \hat{\mathbf{D}})$ by the steepest descent iteration of the form

$$\begin{aligned}\hat{\mathbf{D}}_{n+1}(q) &= \hat{\mathbf{D}}_n(q) - \frac{\epsilon}{2}\nabla_{\hat{D}_n(q)} e_n^2(q, \hat{\mathbf{D}}_n(q)) \\ &= \hat{\mathbf{D}}_n(q) - \epsilon e_n(q, \hat{\mathbf{D}}_n(q)) \nabla\, \mathbf{I}^T(\mathbf{x}_q - \hat{\mathbf{D}}_n(q), t - \tau)\boldsymbol{\phi}_n \qquad (3)\end{aligned}$$

for $n = 0, 1, \cdots, M\text{–}2$ and $q = 0, 1, 2, \cdots$, with

$$\begin{aligned}\hat{\mathbf{D}}_0(q) = \hat{\mathbf{D}}_{M-1}&(q-1) \\ &- \epsilon e_{M-1}(q-1, \hat{\mathbf{D}}_{M-1}(q-1)) \\ &\quad \cdot \nabla\, \mathbf{I}^T(\mathbf{x}_{q-1} - \hat{\mathbf{D}}_{M-1}(q-1), t - \tau)\boldsymbol{\phi}_0, \qquad (4)\end{aligned}$$

where $e_n(q, \hat{D}_n(q))$ is the error in the prediction of $c_n(q)$ (i.e., $c_n(q) - \hat{c}_n(q, \hat{D}_n(q))$) and M is the number of displacement iterations performed per block. Thus the iteration proceeds by first assuming the initial displacement estimate of the qth block, $\hat{\mathbf{D}}_0(q)$, as an update from the final displacement estimate of the $q - 1$ block $\hat{D}_{M-1}(q - 1)$. The next displacement estimate of the qth block, $\hat{\mathbf{D}}_1(q)$, is formed from eq. (3) with $n = 0$. Iteration progresses in the qth block from coefficient to coefficient, resulting finally in displacement estimate $\hat{\mathbf{D}}_{M-1}(q)$. This iteration procedure continues along all horizontal blocks of the raster. The initial displacement of the leftmost block is assumed arbitrarily to be zero.

Such a motion-compensated transform encoder transmits a quantized version of the coefficient prediction error $e_n(q, \hat{\mathbf{D}}(q))$ to the receiver whenever the magnitude $| e_n(q, \hat{\mathbf{D}}(q) |$ exceeds a given threshold T_n, thereby enabling the decoder to update its displacement estimate $\hat{\mathbf{D}}_n(q)$ as in eqs. (3) and (4), as well as correcting its prediction of coefficient $c_n(q)$. Both the encoder and decoder use the updated displacement estimate in predicting the next coefficient, and the process continues. We note that, since only previously transmitted information is used for displacement updating, no separate transmission of displacement is necessary. A simplified block diagram of the hybrid motion-compensated coder is shown in Fig. 5.

The results of motion-compensated coding in the transform domain for the scene Judy are given in Fig. 2. In this figure, total bits per frame are plotted against the frame number. For purposes of comparison,*

* It should be noted that motion compensation in pel domain used intensities of the previous field rather than frame, whereas motion compensation in transform domain used intensities of the previous frame. It was found that, for the pel domain case (Ref. 10), previous field intensities give better results.

this figure also shows conditional replenishment in the pel domain as well as in the transform domain and the two motion compensation techniques of Refs. 10 and 11 in the pel domain. Motion compensation in the transform domain is about 20 to 40 percent better than conditional replenishment in the transform domain. Also, motion compensation in the transform domain is better than one of the pel domain motion compensation techniques by about 5 to 10 percent. This pel domain technique is described in Ref. 10. It segments a frame into three types of areas: background, compensable moving area, and uncompensable moving area, and, therefore, requires a significant amount of address transmission. Motion compensation in the transform domain results in about 20-percent higher bit rates compared to the second motion compensation technique in pel domain. In this second technique, which is described in Ref. 11, each frame is divided only in two parts, predictable and unpredictable, and thus transmission of moving area address information is eliminated.

Results of motion compensation in the transform domain which uses different types of transforms and block sizes are given in Fig. 6. This figure shows that the cosine transform with 2×4 block does the best. Increasing the block size increases the bit rate, perhaps as a result of the uncompensable area (i.e., the pels for which the prediction error is larger than threshold T_n) being in small isolated fragments. This result is in contrast with the result obtained with larger block sizes in conditional replenishment, where a larger block size, such as 2×8, gave better results than a smaller block size, such as 2×4. A one-dimensional transform, for example, the 1×4 block cosine transform, does worse than motion compensation A in the pel domain, whereas a 2×2 block using the cosine transform, on the other hand, is equivalent to motion compensation A in the pel domain.

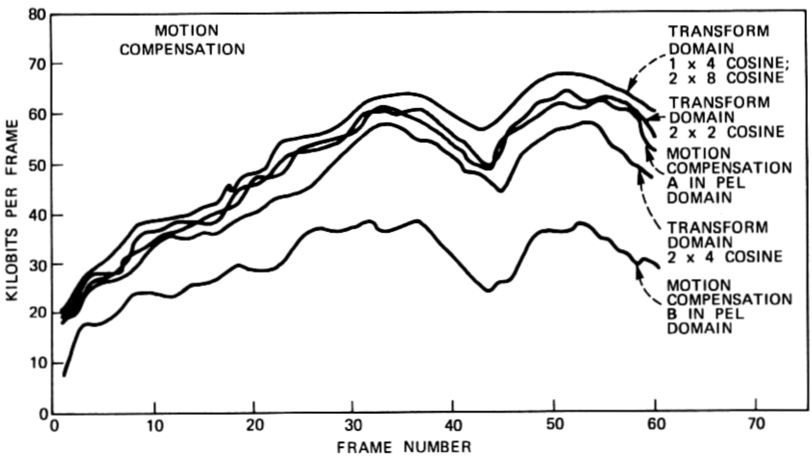

Fig. 6—Performance of motion-compensated coder with different transforms.

Figure 7 is a plot of the portion of the bits required for addressing and the error transmission for the first coefficient. It is seen that, with motion compensation in the transform domain, as in the pel domain, the addressing takes up a significant portion of the total bits. This portion varies from 40 to 60 percent of the total bits. The first coefficient transmission requires more than 50 percent of the bits required for transmission of the coefficients. Although the figure shows the results for 2 × 4 block and cosine transform, similar results were obtained for other transforms.

We found that more coefficients could be dropped altogether from transmission in the motion-compensated transform coder than in the conditional replenishment transform coder. For example, with a 2 × 4 block and the cosine transform, only five coefficients were needed in motion compensation, compared to the seven coefficients that were necessary for conditional replenishment. Unfortunately, however, the effect of dropping a larger number of coefficients did not result in a large bit-rate reduction, since the number of bits required for these coefficients was very small.

The results of Fig. 6 were obtained by adjusting the quantization scales and the predictability thresholds $\{T_n\}$ in such a way that coding degradations in pictures were just perceptible in informal viewing by the authors. The quantization scales that we used were from uniform quantizers with step sizes of 3, 5, 5, 7, 7 (for the first five coefficients of the 2 × 4 cosine transform), and the predictability thresholds were 2, 3, 3, 4, 4 (out of 255) for the first five coefficients. Coarser quantization of the higher order coefficients was possible in motion compensation,

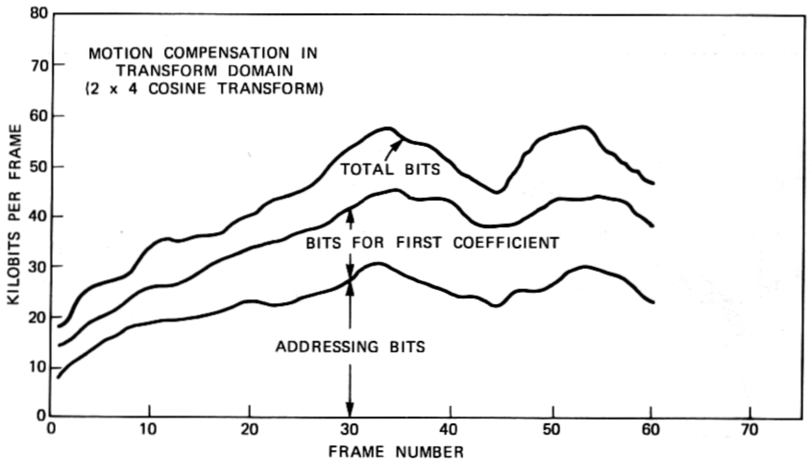

Fig. 7—Distribution of total compressed bits/frame into addressing information and that required for the transmission of the first coefficient.

compared to conditional replenishment, without significantly degrading picture quality. An increase of the predictability thresholds of the first coefficient resulted in rapid degradation of the picture quality. As the predictability threshold was increased, the block structure of the transform became clearly evident, and the frame-to-frame effects of the visibility of the block structure were found rather annoying. For higher order coefficients, however, the picture quality was not a sensitive function of the predictability thresholds. It appears that due to better prediction in motion-compensated transform coding, effects of coarser quantization and higher predictability thresholds are seen in smaller disjoint areas of the picture and, therefore, their visibility is lower.

The recursive displacement estimation algorithm of eq. (2) was also varied by changing ϵ and by changing the order of iteration of the coefficients within a block. We found that ϵ variation did not change the bit rates significantly as long as ϵ was within a decade of 0.0001. As expected,[18] much larger ϵ resulted in noisy estimates of displacement, whereas smaller values of ϵ took longer to converge. The order of iteration was varied by estimating displacement starting from the first coefficient to the fifth (for a 2×4 transform block and cosine transform with transmission of only five coefficients) or starting from the fifth coefficient to the first. This corresponded to iterating from low frequency to high frequency or vice versa. We found that going from high frequency to low frequency resulted in a smaller number of bits for transmission of the prediction error by about 5 to 10 percent. However, since amplitude bits account for only about 50 percent of the total bits, the overall reduction was only about 2 to 5 percent. Another variation that was tried consisted of iterating only the first (dc) coefficient for the entire block; that is, iterating the first coefficient five times instead of iterating all the five coefficients once. This variation resulted in performance which was very similar to the case in which all the coefficients were iterated. Iterating some other higher order coefficients five times (with no iteration of the first coefficient), however, was found to be quite inferior. Although all the above conclusions are based on the scene Judy, similar conclusions are true for the scene Mike and Nadine. In general, as in pel domain,[10, 11] the bit rate for Mike and Nadine was much higher than that for Judy. It varied between 170 and 200 kilobits per frame for conditional replenishment in the transform domain, compared to 150 to 175 kilobits per frame for motion-compensated transform coding.

IV. MOTION COMPENSATION WITH TRANSMITTED DISPLACEMENT

In this section, we give results of estimating displacement by a technique proposed by Limb and Murphy[19] and then use it for motion-compensated prediction. The displacement computation was done in "displacement blocks" with varying sizes, such that the transform block was an exact submultiple of the displacement block in both dimensions. Also, coded values of intensities of the previous frame were used to obviate the need of an additional frame store. Having computed the displacement for the displacement block by the Limb/Murphy algorithm, each transform coefficient within the transform block is predicted by using the displaced coefficient from the previous frame or the nondisplaced coefficient from the previous frame, depending on which was better for the previous coefficient of the same block. In this scheme, there is a tradeoff between the displacement block size and the total number of bits required for a given picture quality. A large displacement block size tends to average all the local variations of the displacement and, consequently, may not result in a good prediction; however, it requires less overhead for transmission of the displacement estimate. On the other hand, a small displacement block requires larger overhead but is potentially superior for displacement estimation in noiseless data. For real scenes, however, the quality of displacement estimation using small blocks might also suffer.

Our simulations used three sizes for the displacement blocks: 16 × 32 (i.e., 16 lines × 32 elements in the same field), 8 × 16, and 4 × 8. These blocks were approximately square, considering the interlace. Only a 2 × 4 transform block with the cosine transform was used. All the rest of the coder parameters were adjusted to generate pictures of approximately the same quality as before. For both the scenes, without accounting for bits required for transmission of displacement information, a displacement block size of 8 × 16 did the best in terms of bits per frame. For the scene Judy, displacement blocks of 16 × 32 and 4 × 8 resulted in bit rates that were higher by approximately 1000 bits per frame and 2000 bits per frame, respectively. For the scene Mike and Nadine, similar comparisons resulted in about 3000 bits per frame and 5000 bits per frame. Also without accounting for those bits necessary for transmission of displacement information, the 8 × 16 displacement block resulted in bit rates comparable to those of previous sections with recursive displacement estimation for Judy, but about 5000 to 7000 bits per frame higher for the scene Mike and Nadine. This, however, is a small percentage of the total bits transmitted per frame. As mentioned earlier, the schemes of this section require transmission of displacement information. We did not study any schemes to optimize transmission of this information. Assuming that each D_x and D_y can be specified by 8 bits, we would need 2024, 8096, and 32,384 bits per frame for 16 × 32, 8 × 16, and 4 × 8 displacement blocks, respectively. Clearly, considering the overall bit rate, displacement blocks of 8 × 16 and 16 × 32 are similar in performance, with a slight preference for the 16 × 32 block. The 4 × 8 displacement block is significantly worse, perhaps due to very noisy displacement estimates. Comparison of the techniques of this section with that of the previous section indicates a slight preference for recursive schemes.

V. CONCLUSIONS

We have developed two schemes for motion compensation in the transform domain. In the first scheme, displacement is estimated recursively from previously transmitted coefficients, whereas the second scheme estimates displacements in a block (generally larger than the transform block itself) using some past and future data, and, therefore, requires separate transmission of displacement information. We found that motion compensation resulted in bit rates which were 20 to 40 percent better than the conventional hybrid interframe coders, which use frame difference prediction. Motion compensation in the pel domain was superior to that in the transform domain for the particular coder structures we investigated. The two methods of displacement estimation were quite similar in performance with a slight preference for the scheme with recursive displacement estimation. None of these comparisons was based on hardware complexity, and it is possible that hardware considerations may change the preferences.

REFERENCES

1. F. W. Mounts, "A Video Encoding System Using Conditional Picture-Element Replenishment," B.S.T.J., *48*, No. 7 (September 1969), pp. 2545–2554.
2. B. G. Haskell, F. W. Mounts, and J. C. Candy, "Interframe Coding of Videotelephone Pictures," Proc. IEEE, *60*, No. 7 (July 1972), pp. 792–800.
3. J. O. Limb, R. F. W. Pease, and K. A. Walsh, "Combining Intraframe and Frame-to-Frame Coding for Television," B.S.T.J., *53*, No. 6 (August 1974), pp. 1137–1173.
4. T. Ishiguro, K. Iinuma, Y. Iijima, T. Koga, S. Azami, and T. Mune, "Composite Interframe Coding of NTSC Television Signals," 1976 Nat. Telecommun. Conf. Record, *1*, Dallas, Texas, November 1976, pp. 6.4–1 to 6.4–5.
5. B. G. Haskell, P. L. Gordon, R. L. Schmidt, and J. V. Scattaglia, "Interframe Coding of 525-Line Monochrome Television at 1.5Mbs," IEEE Trans. Commun., *COM-25* (October 1977), pp. 1461–1466.
6. F. Rocca, "Television Bandwidth Compression Utilizing Frame-to-Frame Correlation and Movement Compensation," Symposium on Picture Bandwidth Compression, M.I.T., Cambridge, Mass., 1969; Gordon and Breach, 1972.
7. B. G. Haskell and J. O. Limb, "Predictive Video Encoding Using Measured Subject Velocity," January 1972, U.S. Patent 3, 632, 865.
8. C. Cafforio and F. Rocca, "Methods for Measuring Small Displacements of Television Images," IEEE Trans. Inform. Theory, *IT-22*, No. 5 (September 1976), pp. 573–579.
9. B. G. Haskell, "Entropy Measurements for Nonadaptive and Adaptive, Frame-to-Frame, Linear Predictive Coding of Video Telephone Signals," B.S.T.J., *54*, No. 6 (August 1975), pp. 1155–1174.
10. A. N. Netravali and J. D. Robbins, "Motion-Compensated Television Coding: Part I," B.S.T.J., *58*, No. 3 (March 1979), pp. 629–668.
11. J. D. Robbins and A. N. Netravali, "Interframe Television Coding Using Movement Compensation," Int. Conf. Commun., Boston, Mass., June 1979.
12. P. A. Wintz, "Transform Picture Coding," Proc. IEEE, *60*, No. 7 (July 1972), pp. 809–820.

13. C. Reader, "Orthogonal Transform Coding of Still and Moving Pictures," Ph.D. dissertation, The University of Sussex, United Kingdom, 1974.
14. J. A. Roese, W. K. Pratt, and G. S. Robinson, "Interframe Cosine Transform Image Coding," IEEE Trans. Commun., *COM-25*, No. 11 (November 1977), pp. 1329–1339.
15. H. W. Jones, "A Conditional Replenishment Hadamard Video Compressor," SPIE, *119*, Applications of Digital Image Processing, 1977, pp. 91–98.
16. S. C. Knauer, "Real-Time Video Compression Algorithm for Hadamard Transform Processing," Proc. SPIE, *66* (August 1975), pp. 58–69.
17. A. Habibi, "Hybrid Coding of Pictorial Data," IEEE Trans. Commun., *COM-22*, No. 5 (May 1974), pp. 614–624.
18. J. A. Stuller and A. N. Netravali, "Transform Domain Motion Estimation," B.S.T.J., this issue, pp. 1673–1702.
19. J. O. Limb and J. A. Murphy, "Estimating the Velocity of Moving Images from Television Signals," Computer Graphics and Image Processing, *4*, 1975, pp. 311–327.