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 × 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

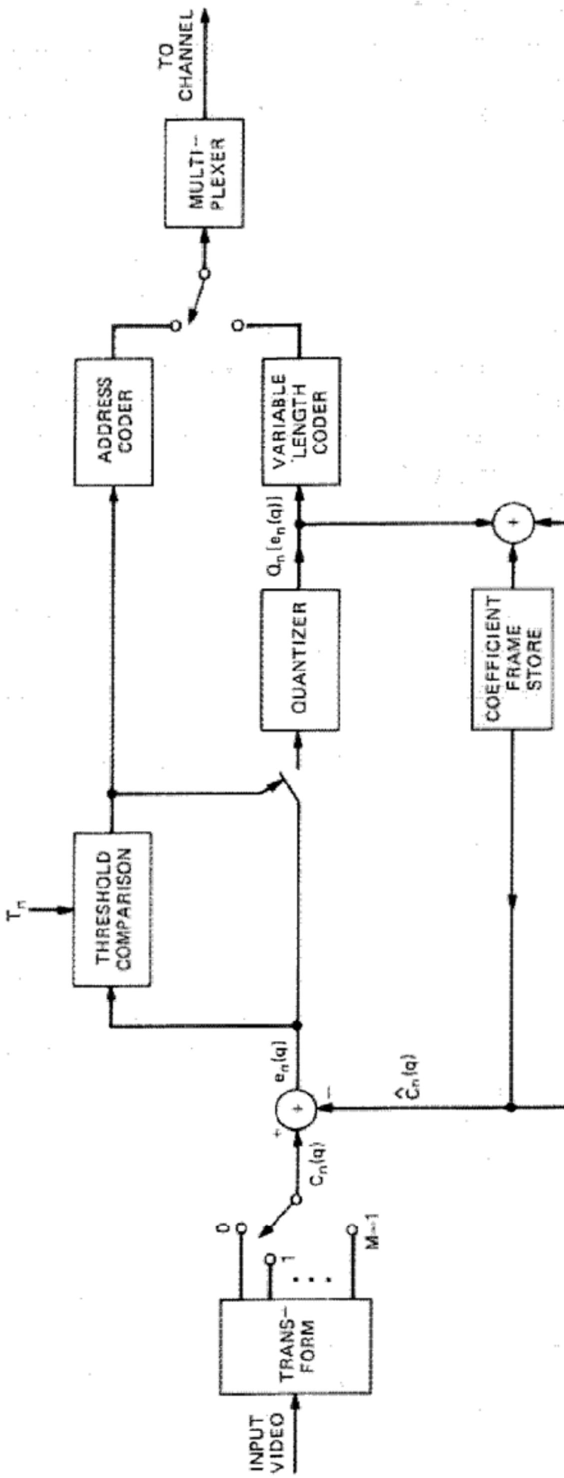

Fig. 1—Block diagram of hybrid transform/DPCM interframe image encoder.

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 $\{c_k\}_{k=0,\ldots,M-1}$ and $\{\bar{c}_k\}_{,k=0,\ldots,M-1}$ 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.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.