THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 58, No. 7, September 1979
Printed in U.S.A.

Transform Domain Motion Estimation

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

(Manuscript received March 21, 1979)

This paper introduces an algorithm for estimating the displacement of moving objects in a television scene from spatial transform coefficients of successive frames. The algorithm works recursively in such a way that the displacement estimates are updated from coefficient to coefficient. A promising application of this algorithm is in motion-compensated interframe hybrid transform-DPCM image coding. We give a statistical analysis of the transform domain displacement estimation algorithm and prove its convergence under certain realistic conditions. An analytical derivation is presented that gives sufficient conditions for the rate of convergence of the algorithm to be independent of the transform type. This result is supported by a number of simulation examples using Hadamard, Haar, and Slant transforms. We also describe an extension of the algorithm that adaptively updates displacement estimation according to the local features of the moving objects. Simulation results demonstrate that the adaptive displacement estimation algorithm has good convergence properties in estimating displacement even for very noisy images.

I. INTRODUCTION

The coefficient-recursive algorithm described in this paper estimates the displacement of objects in a television scene. It is a generalization of a pel-recursive displacement estimation algorithm recently introduced by Netravali and Robbins.[1,2] Coefficient-recursive displacement estimation has potential application in hybrid transform-DPCM[3,4] interframe image coders of the type discussed by Reader,[5] Roese,[6] and Jones.[7] The performance of a hybrid transform-DPCM interframe coder using coefficient recursive motion compensation is described in a companion paper.[8]

Before defining the coefficient-recursive displacement estimation algorithm, it is useful to first describe pel-recursive displacement

estimation. Let $I(\mathbf{x}_k, t)$ denote the intensity of a scene at the kth sample point $\mathbf{x}_k$ of a scan line, and let $I(\mathbf{x}_k, t-\tau)$ denote the intensity at the same spatial location in the previous frame. If the scene consists of an object that is undergoing pure translation, then, neglecting background,

$$I(\mathbf{x}_k, t) = I(\mathbf{x}_k - \mathbf{D}, t-\tau), \tag{1}$$

where $\mathbf{D}$ is the displacement of the object in one frame interval τ. Pel-recursive displacement estimation attempts to estimate $\mathbf{D}$ by minimizing the squared value of the displaced frame difference,

$$DFD(\mathbf{x}_k, \hat{\mathbf{D}}) = I(\mathbf{x}_k, t) - I(\mathbf{x}_k - \hat{\mathbf{D}}, t-\tau), \tag{2}$$

recursively with k using a steepest descent algorithm of the form:

$$\hat{\mathbf{D}}_{k+1} = \hat{\mathbf{D}}_k - \tfrac{1}{2}\,\epsilon\,\nabla_{\hat{\mathbf{D}}_k}[DFD(\mathbf{x}_k, \hat{\mathbf{D}}_k)]^2, \tag{3a}$$

where $\nabla_{\hat{\mathbf{D}}_k}$ is the two-dimensional gradient operator with respect to $\hat{\mathbf{D}}_k$. Carrying out this operation in (3a) and using (2) yields

$$\hat{\mathbf{D}}_{k+1} = \hat{\mathbf{D}}_k - \epsilon\, DFD(\mathbf{x}_k, \hat{\mathbf{D}}_k)\, \nabla I(\mathbf{x}_k - \hat{\mathbf{D}}_k, t-\tau), \tag{3b}$$

where $\nabla = \nabla_x$ is the two-dimensional spatial gradient operator with respect to horizontal and vertical coordinates x_1 and x_2 in $\mathbf{x} = (x_1, x_2)^T$:

$$\nabla I(\mathbf{x}_k - \hat{\mathbf{D}}_k \cdot t-\tau) = \begin{bmatrix} \dfrac{\partial}{\partial x_1} \\ \dfrac{\partial}{\partial x_2} \end{bmatrix} I(\mathbf{x}, t-\tau)\}_{\mathbf{x}=\mathbf{x}_k-\hat{\mathbf{D}}_k}. \tag{4}$$

Superscript T denotes transpose of a vector or matrix. The pel-domain interframe coder of Netravali and Robbins predicts intensity $I(\mathbf{x}_k, t)$ by the displaced previous frame intensity $I(\mathbf{x}_k - \hat{\mathbf{D}}_k, t-\tau)$ using interpolation for nonintegral $\hat{\mathbf{D}}_k$. If the magnitude of the prediction error exceeds a predetermined threshold, the coder transmits a quantized version of $DFD(\mathbf{x}_k, \hat{\mathbf{D}}_k)$ and address information to the receiver. Both receiver and transmitter then update $\hat{\mathbf{D}}_k$ according to (3b) using this quantized version. Netravali and Robbins[2] found that a coder using this algorithm consistently obtained bit rates that were 30 to 60 percent lower than those obtained by "frame-difference" prediction, which is commonly used in interframe coders.

In an interframe hybrid transform-DPCM coder,[5-7] individual frames of video are partitioned into blocks having dimension N_r rows by N_c columns, and a two-dimensional transform is performed on each block to produce a set of coefficients. The transform coefficients of the qth block of the present frame are predicted by the corresponding coeffi-

cients of the qth block of the previous (reference) frame and, if the prediction error is sufficiently high, the quantized prediction errors are transmitted to the receiver. These quantized errors add as correction terms to the coefficients predicted by the decoder, which inverse-transforms the result to obtain the decoded image. This process repeats with both coder and decoder predicting the transform coefficients of the next frame by the coefficients of the decoded frame, as illustrated in Fig. 1. In this type of codec, data compression is achieved both by the redundancy reduction implicit in the prediction process and by the fact that some coefficients can be reproduced with low precision (or totally omitted) without visibly degrading the reconstructed image. An advantage of interframe hybrid transform-DPCM coding over conventional (3-dimensional block) interframe transform coding[9] is the fact that the hybrid coder requires only a single frame of storage while the conventional transform coder requires many.

In a motion-compensated hybrid transform-DPCM coder of the type envisioned (Fig. 2), the nth coefficient of the qth present-frame block would be predicted by the nth coefficient of the *displaced* qth block of the previous frame where the displacement is a recursively updated estimate of frame-to-frame translation of the moving object. In this paper, we introduce and analyze a displacement estimation technique that operates recursively on coefficients in a manner analogous to the way (3) operates on pels.

Section II of this paper defines the coefficient-recursive displacement estimation algorithm for any real linear transform and gives illustrative simulation results using a separable 2-row by 8-column (2×8) transform block. A statistical analysis of the algorithm is given in Section III. In the analysis of Section III, a single frame is modeled as an image drawn at random from a stationary and ergodic ensemble of images. This random sample is assumed to be undergoing pure translation from frame to frame. An important result of this analysis is stated in Assertion 3 of Section 3.2, which says that, under certain conditions, the convergence properties of the coefficient-recursive displacement estimation are independent of the transform used. Section III presents simulation results that support this claim using Hadamard, Haar, and Slant transforms. Section IV describes an adaptive version of the coefficient-recursive algorithm and presents simulation results that indicate that this version can be used to some advantage in displacement estimation for noisy images. Illustrative simulation results are shown here using a 2×4 cosine transform block.

The algorithms discussed in this paper are local in nature and, as such, can estimate the individual frame-to-frame displacements of several objects that may be present in the television scene. However, we emphasize that all results presented here apply to objects undergoing pure translation; other types of motion are applicable to this study

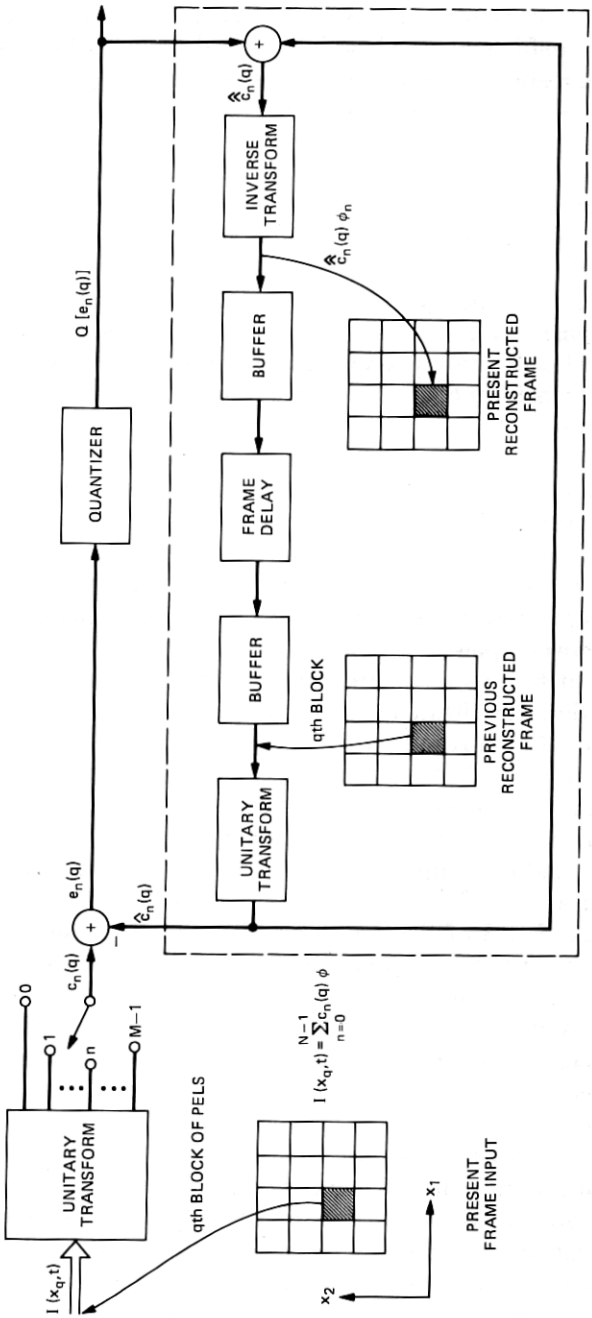

Fig. 1—Simplified diagram of hybrid transform-DPCM codec. The transform/inverse transform operations inside the dashed box are included for conceptual purposes.

to the extent that these can be approximated by pure translation over the spatial dimensions of a transform block. Background uncovered by moving objects is also ignored throughout this paper. In spite of the approximations involved, simulation results to be described in Ref. 8 show that the coefficient-recursive displacement algorithm studied here can be substantially beneficial when used in an interframe hybrid-DPCM codec operating on real-life scenes.

II. COEFFICIENT-RECURSIVE DISPLACEMENT ESTIMATION

Let a field of video be partitioned into rectangular blocks of pels, each having dimension N_r rows and N_c columns ($N_r \times N_c$). Let $\mathbf{x}_q = (x_{1q}, x_{2q})^T$ denote the coordinates 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 $q = 0, 1, 2, \cdots$. We number the $N = N_r N_c$ pel intensities of block q in a column-scanning fashion and denote them by a column vector $I(\mathbf{x}_q, t)$. Let the N component vector ϕ_n be the nth basis vector of a nonzero but otherwise arbitrary real linear transform, and denote the nth coefficient of the qth block of this transform in the present frame by $c_n(q)$, where

$$c_n(q) = \mathbf{I}^T(\mathbf{x}_q, t)\phi_n \tag{5}$$

and n is numbered from 0 to $N - 1$. The displaced previous frame value of this coefficient is

$$\hat{c}_n(q, \hat{\mathbf{D}}) = \mathbf{I}^T(\mathbf{x}_q - \hat{\mathbf{D}}, t - \tau)\phi_n, \tag{6}$$

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. Prediction of $c_n(q)$ of (5) by $\hat{c}_n(q, \hat{\mathbf{D}})$ of (6) results in coefficient prediction error

$$e_n(q, \hat{\mathbf{D}}) = [\mathbf{I}(\mathbf{x}_q, t) - \mathbf{I}(\mathbf{x}_q - \hat{\mathbf{D}}, t - \tau)]^T\phi_n. \tag{7}$$

The algorithm defined in this section attempts to decrease the squared-prediction errors $e(q, \hat{\mathbf{D}})^2$ in a coefficient-recursive manner by steepest descent iteration of the form

$$\begin{aligned}\hat{\mathbf{D}}_{n+1}(q) &= \hat{\mathbf{D}}_n(q) - \frac{\epsilon}{2}\nabla_{\hat{\mathbf{D}}_n(q)}e_n^2(q, \hat{\mathbf{D}}_n(q)) \\ &= \hat{\mathbf{D}}_n(q) - \epsilon e_n(q, \hat{\mathbf{D}}_n(q))\mathbf{G}_n(q)\end{aligned} \tag{8a}$$

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

$$\begin{aligned}\hat{\mathbf{D}}_0(q) &= \hat{\mathbf{D}}_{M-1}(q - 1) \\ &\quad - \epsilon e_{M-1}(q - 1, \hat{\mathbf{D}}_{M-1}(q - 1))\mathbf{G}_{M-1}(q - 1).\end{aligned} \tag{8b}$$

In (8), $\mathbf{G}_n(q)$ is the *coefficient gradient vector*

$$\mathbf{G}_n(q) = \nabla \mathbf{I}^T(\mathbf{x}_q - \hat{\mathbf{D}}_n(q), t - \tau)\phi_n. \tag{9}$$

Note that (8) operates upon coefficients 0 through $M - 1$, where for generality we assume $M \leq N$. Iteration in (8) progresses as follows. The initial displacement estimate of the qth block ($q > 1$), $\hat{\mathbf{D}}_0(q)$ is formed by updating the final displacement estimate $\hat{\mathbf{D}}_{M-1}(q-1)$ of the previous block as in (8b). The next displacement estimate of the qth block $\hat{\mathbf{D}}_1(q)$ is formed from (8a) with $n = 0$. Iteration progresses in the qth block by (8a) with $n = 1, 2, \cdots, M - 2$, resulting finally in displacement estimate $\hat{\mathbf{D}}_{M-1}(q)$ which, when updated in (8b) (with $q \rightarrow q + 1$), forms the initial displacement estimate $\hat{\mathbf{D}}_0(q+1)$ of block $q + 1$. This iteration procedure continues along all horizontal blocks of raster. The procedure is started in the $q = 0$ block with an arbitrarily chosen initial displacement estimate $\hat{\mathbf{D}}_0(0)$ followed by iterations of (8a) for $n = 0, 1, \cdots, M - 2$ and $q = 0$. In the sequel we assume that $\hat{\mathbf{D}}_0(0)$ is zero.

The envisioned motion-compensated interframe hybrid transform-DPCM coder transmits a quantized version of 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 threshold, thereby enabling the decoder to update its displacement estimate $\hat{\mathbf{D}}_n(q)$ as in (8) as well as correcting its prediction $\hat{c}_n(q, \hat{\mathbf{D}}_n(q))$ of coefficient $c_n(q)$. Both encoder and decoder use the updated displacement estimate in predicting the next coefficient, and the process continues. A simplified block diagram of the system that omits the thresholding operations is given in Fig. 2.

In the sequel, it is convenient to rewrite (8) in a form that explicitly describes the iteration convention. This can be done by defining a single index i,

$$i = qM + n; \qquad i = 0, 1, 2, \cdots \tag{10a}$$

that equals the total number of iterations of (8) that have occurred in iterating from $\hat{\mathbf{D}}_0(0)$ to $\hat{\mathbf{D}}_n(q)$. Quantities q and n are related to i by

$$n = ((i)) \tag{10b}$$

$$q = [[i]], \tag{10c}$$

where we use the notation $((i))$ to denote i modulo M and $[[i]]$ to denote the integer part of i/M.

Using (10), we set $\hat{\mathbf{D}}_i \triangleq \hat{\mathbf{D}}_n(q)$ and rewrite (8) as

$$\hat{\mathbf{D}}_{i+1} = \hat{\mathbf{D}}_i - \epsilon e_{((i))}([[i]]), \hat{\mathbf{D}}_i)\mathbf{G}_{((i))}([[i]]) \tag{11}$$

with $i = 0, 1, 2, \cdots$. Note that the Netravali-Robbins pel-recursive

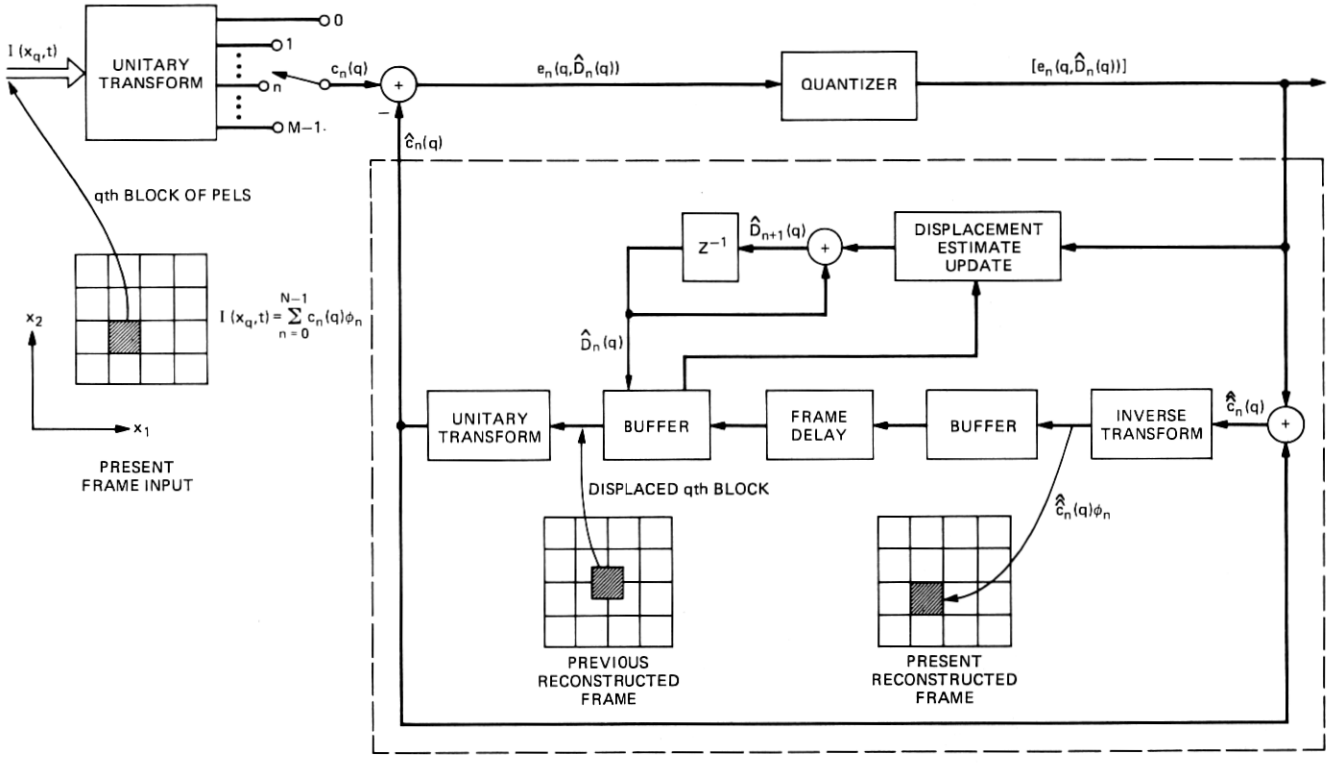

Fig. 2—Simplified diagram of motion-compensated hybrid transform-DPCM codec.

displacement estimation algorithm (3b) is a special case of (10) that results for a transform "block" having dimension 1 pel by 1 pel and single "basis vector" $\phi_0 = 1$.

We emphasize that recursions (8) and (11) were derived with the objective of decreasing the squared coefficient prediction errors of a hybrid transform-DPCM codec. As shown in Section III, coefficient prediction error is related to displacement estimation error (approximately) by a dot product between the displacement estimation error vector and vector $\mathbf{G}_n(q)$ that describes the spatial rates of change of the coefficient estimate $\hat{c}_n(q, \hat{\mathbf{D}}_n(q))$ with respect to small displacements of the block. Therefore, only the component of displacement estimation error in the direction of $\mathbf{G}_n(q)$ contributes to coefficient prediction error, and it is this component that is relevant in evaluating the performance of (8) or (11). For this reason, experimental results given in this paper refer to the component of displacement estimation error measured in the direction of its corresponding coefficient gradient $\mathbf{G}_n(q)$.

Experimental illustrations of the behavior of (11) are given in Figs. 3 through 5 where the moving object was the synthetically generated pattern of Fig. 6, displaced 2 pels in the horizontal direction each frame interval τ. This is a radial cosine function having a radius of 60 pels, and peak-to-peak amplitude 220 (out of an intensity range 0 to 255) at its center, decreasing to 130 at the circumference. The period P decreases with radial distance R starting with a period of 20 pels at center to 10 pels at the circumference. The pattern is described mathematically by the intensity function

$$f(R) = 100 \exp(-0.01\, R)\cos(2\pi R/P) + 128; \quad 0 \le R \le 60, \qquad (12a)$$

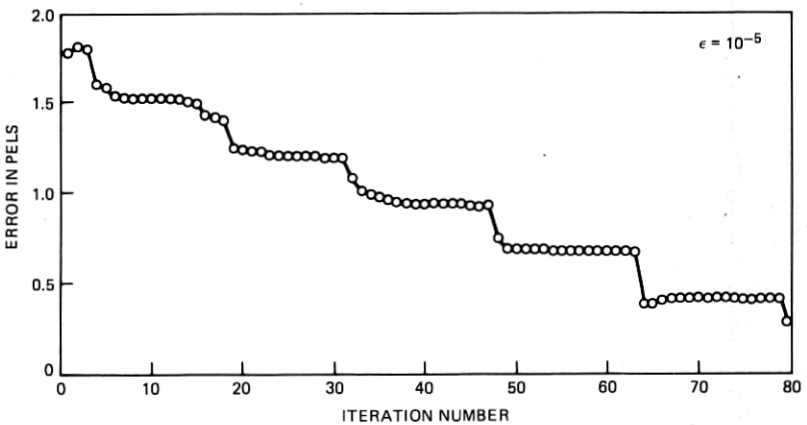

Fig. 3—Single-line convergence results using 2 × 8 separable Hadamard transform with $\epsilon = 10^{-5}$.

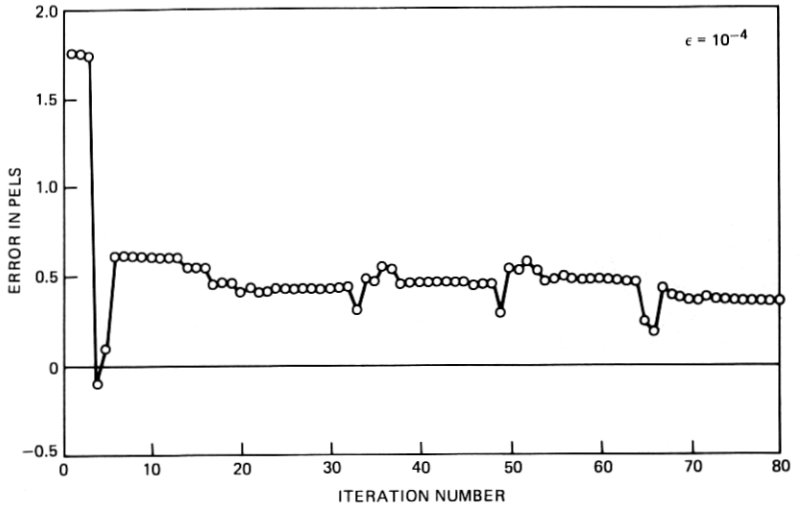

Fig. 4—Single-line convergence results using 2 × 8 separable Hadamard transform with $\epsilon = 10^{-4}$.

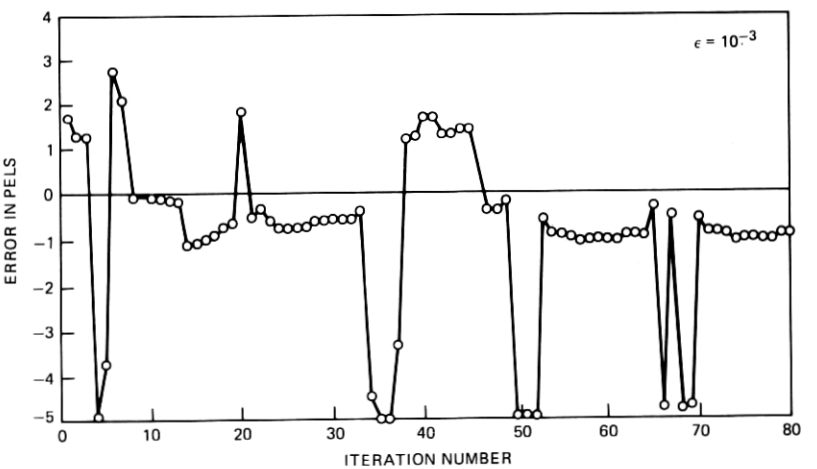

Fig. 5—Single-line convergence results using 2 × 8 separable Hadamard transform with $\epsilon = 10^{-3}$.

where

$$P = (1 - R/60.)\ 10 + 10. \tag{12b}$$

This function is displayed on a 256 × 256 element raster in two interlaced fields of 128 lines each. In applying (11), the spatial transforms were taken over a single field with coefficient prediction performed from the corresponding field separated in time by a frame

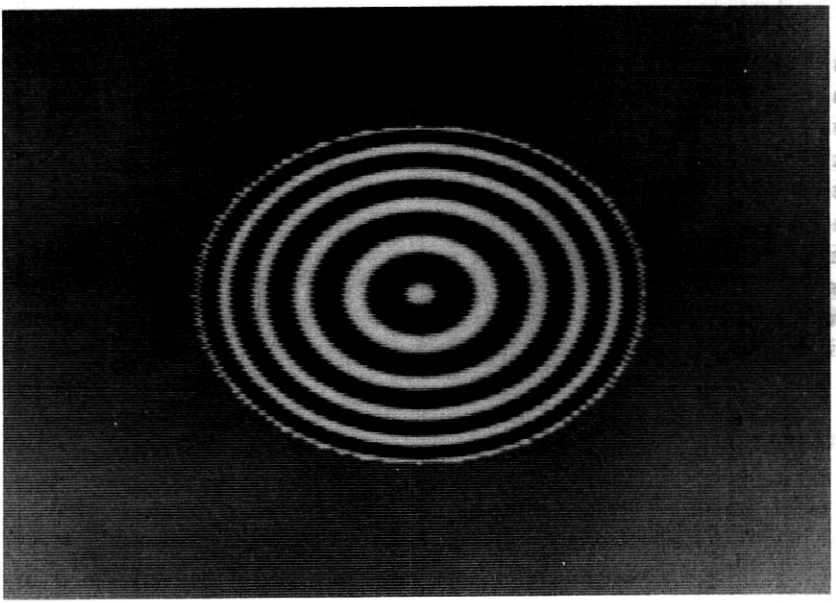

Fig. 6—Synthetic image used in simulations. This image is described by eq. (11) of Section II.

interval τ. Figures 3 through 5 show displacement estimation error in the direction of the spatial gradient of the corresponding coefficient versus iteration for a sequence of 2 × 8 blocks located 10 field lines above the center of the figure. Iteration was initiated with a zero displacement estimate approximately 7 pels within the pattern for each horizontal sequence. In these examples and all others presented in this paper, the two-dimensional transforms concerned were separable transforms of the form[10]

$$C = V[I]H, \tag{13}$$

where C is the $N_r \times N_c$ coefficient matrix, $[I]$ is the $N_r \times N_c$ matrix of pels, and V and H are unitary matrices having dimensions $N_r \times N_r$ and $N_c \times N_c$, respectively. Coefficient $c_n(q)$ of (5) is the nth column scanned coefficient of matrix C, with $[I]$ taken to be the qth pixel block of the present frame. In Figs. 3 through 5, V and H were the normalized sequency-ordered 2 × 2 and 8 × 8 Hadamard matrices of Fig. 7, and iteration of (11) progressed through all $M = 16$ coefficients in a block (i.e., $M = N$). Figure 3 illustrates the behavior of (11) for $\epsilon = 10^{-5}$. It can be seen by inspection of this figure that displacement estimation error tends to decrease roughly in a series of steps of 16 iterations. Iterations 1 to 6, 17 to 22, etc., corresponding to the first few (low sequency) vectors in $\{\phi_n\}$ tend to affect error significantly, while the

iterations corresponding to the higher sequency basis vectors do not. This type of behavior is scene-dependent and is investigated in the analysis of Section III. Figures 4 and 5 illustrate convergence for increased values of ϵ. In general, convergence rate increases gradually with increasing ϵ up to a point after which oscillations occur. Convergence to within 0.5-pel error was achieved for this particular example in two or three iterations for $\epsilon = 10^{-4}$ while the recursion became oscillatory (and eventually unstable as in Fig. 5) at values exceeding this. All these results are scene-dependent and, to some extent, dependent upon the row position of the sequence of blocks. Because of this, the behavior of (11) will be examined from a statistical viewpoint in Section III.

III. PROPERTIES OF COEFFICIENT-RECURSIVE DISPLACEMENT ESTIMATION

The displacement estimation procedure defined by (8) is a nonlinear recursion relation whose dynamic behavior is complicated by the fact that the error $e_n(q, \hat{\mathbf{D}}_n(q))^2$ is generally a multimodal function of $\hat{\mathbf{D}}_n(q)$, having global minimum at $\hat{\mathbf{D}}_n(q) = \mathbf{D}$ and local minima elsewhere. Convergent solutions to eq. (8) can, therefore, exist at displacement estimates other than the true displacement $\mathbf{D}$. Subsequent analysis will restrict consideration to the case in which the displacement estimate is sufficiently close to the true displacement that $e^2(q, \hat{\mathbf{D}})$ can be approximated as a quadratic function of $\hat{\mathbf{D}} - \mathbf{D}$. Under this restriction, the anomalous solutions can be ignored, and (8) reduces to an approximately linear stochastic recurrence relation.

Section 3.1 derives the linearized approximation to (8) and the associated quadratic error expression. The dynamic behavior of the coefficient-recursive displacement estimator is analyzed in Section 3.2. An important result of this section is that, for ϵ sufficiently small, the block-to-block convergence rate of mean displacement estimation error resulting from (8) is independent of the transform used for any unitary transform.

$$\begin{bmatrix} 0.707 & 0.707 \\ 0.707 & -0.707 \end{bmatrix}$$

(a)

$$\begin{bmatrix} 0.353 & 0.353 & 0.353 & 0.353 & 0.353 & 0.353 & 0.353 & 0.353 \\ 0.353 & 0.353 & 0.353 & 0.353 & -0.353 & -0.353 & -0.353 & -0.353 \\ 0.353 & 0.353 & -0.353 & -0.353 & -0.353 & -0.353 & 0.353 & 0.353 \\ 0.353 & 0.353 & -0.353 & -0.353 & 0.353 & 0.353 & -0.353 & -0.353 \\ 0.353 & -0.353 & -0.353 & 0.353 & 0.353 & -0.353 & -0.353 & 0.353 \\ 0.353 & -0.353 & -0.353 & 0.353 & -0.353 & 0.353 & 0.353 & -0.353 \\ 0.353 & -0.353 & 0.353 & -0.353 & -0.353 & 0.353 & -0.353 & 0.353 \end{bmatrix}$$

(b)

Fig. 7—Sequency-ordered Hadamard matrices. (a) 2×2. (b) 8×8.

3.1 Linear analysis

Assume that the pel intensities are samples of an object that is undergoing pure translation $\mathbf{D}$ from frame to frame as in (1) so that, neglecting background,

$$\mathbf{I}(\mathbf{x}_q, t) = \mathbf{I}(\mathbf{x}_q - \mathbf{D}, t - \tau). \tag{14}$$

For Euclidean norm $\| \hat{\mathbf{D}}_n(q) - \mathbf{D} \|$ sufficiently small, eq. (7) becomes, by Taylor's expansion about $\mathbf{D} - \hat{\mathbf{D}}(q)$, to a linear approximation

$$\begin{aligned} e_n(q, \hat{\mathbf{D}}_n(q)) &= [\mathbf{I}(\mathbf{x}_q - \mathbf{D}, t - \tau) - \mathbf{I}(\mathbf{x}_q - \hat{\mathbf{D}}_n(q), t - \tau)]^T \phi_n \\ &\approx \mathbf{G}_n^T(q)\, \boldsymbol{\Delta}_n(q), \end{aligned} \tag{15}$$

where $\mathbf{G}_n(q)$ is given by (9) and $\boldsymbol{\Delta}_n(q)$ is the displacement estimation error $\boldsymbol{\Delta}_n(q) = \hat{\mathbf{D}}_n(q) - \mathbf{D}$. Using the approximation in (15), we can approximate the squared coefficient estimate error by

$$e_n^2(q, \hat{\mathbf{D}}_n(q)) \approx \boldsymbol{\Delta}_n^T(q)\, [\mathbf{G}_n(q)\, \mathbf{G}_n^T(q)] \boldsymbol{\Delta}_n(q), \tag{16}$$

which is a quadratic function of the horizontal and vertical components of $\boldsymbol{\Delta}_n(q)$.

In terms of approximation (15), (8) assumes the form

$$\boldsymbol{\Delta}_{n+1}(q) = [U - \epsilon\, \mathbf{G}_n(q) \mathbf{G}_n^T(q)] \boldsymbol{\Delta}_n(q) \tag{17a}$$

with

$$\boldsymbol{\Delta}_0(q) = [U - \epsilon\, \mathbf{G}_{M-1}(q-1) \mathbf{G}_{M-1}^T (q-1)] \boldsymbol{\Delta}_{M-1}(q-1), \tag{17b}$$

where U is the 2×2 identity matrix. Similarly, (11) becomes

$$\boldsymbol{\Delta}_{i+1} = [U - \epsilon\, \mathbf{G}_{((i))}([[i]]) \mathbf{G}_{((i))}^T([[i]])] \boldsymbol{\Delta}_i, \quad i = 0, 1, \cdots, \tag{18}$$

where $\boldsymbol{\Delta}_i \triangleq \boldsymbol{\Delta}_n(q)$.

Considering the image as a random process, (18) is a stochastic recurrence relation. Equations similar to, but somewhat simpler than, (18) have appeared in the problem of adaptive tap gain adjustment of automatic channel equalizers. Unfortunately, a complete statistical description of the behavior of these simpler equations has not yet been obtained. The difficulty in analyzing these equations is that their solution depends upon products of matrices that are statistically dependent. It has been found, however, that useful approximate results can be obtained by treating the dependent matrices as if they were actually independent.[11] We use this method in Section 3.2 to analyze (18). As shown in Section 3.2 and Appendix A, there is some analytical justification for this approach because of certain properties of the transforms conventionally used in image coding. Further justification is given in the asymptotic analysis of Appendix B.

3.2 Statistical analysis

This section studies the behavior of mean displacement estimation error under the assumption that the sequence of gradient vectors $\mathbf{G}_{((i))}([[i]])$ entering (18) are statistically independent ("independence assumption"). Note that, if the sequence of $\mathbf{G}_{((i))}([[i]])$ in (18) are independent, then $\boldsymbol{\Delta}_i$ will be independent of the matrix premultiplying it. This follows from the fact that $\boldsymbol{\Delta}_i$ is determined by the $\mathbf{G}_{((j))}([[j]])$, $j < i$, which, by assumption, are independent of $\mathbf{G}_{((i))}([[i]])$. The ensemble mean of (18) (denoted by the overhead bar) then becomes

$$\bar{\boldsymbol{\Delta}}_{i+1} = [U - \epsilon\overline{\mathbf{G}_{((i))}\mathbf{G}_{((i))}^T}]\bar{\boldsymbol{\Delta}}_i, \quad i = 0, 1, 2, \cdots. \tag{19}$$

In writing (19), we have used (9) and assumed a stationary ergodic image ensemble. The matrix $\overline{\mathbf{G}_n(q)\mathbf{G}_n^T(q)}$ in this case will not depend upon $\mathbf{x}_q - \hat{\mathbf{D}}_n(q)$ and is written simply as $\overline{\mathbf{G}_{((i))}\mathbf{G}_{((i))}^T}$. The matrix $\overline{\mathbf{G}_{((i))}\mathbf{G}_{((i))}^T}$ is periodic in i with period M, having values specified by $\overline{\mathbf{G}_n\mathbf{G}_n^T}$, $n = 0, 1, \cdots M-1$. Alternative expressions for $\overline{\mathbf{G}_n\mathbf{G}_n^T}$ are

$$\overline{\mathbf{G}_n\mathbf{G}_n^T} = \overline{\nabla\mathbf{I}^T(\mathbf{x})\boldsymbol{\phi}_n\boldsymbol{\phi}_n^T\nabla^T\mathbf{I}(\mathbf{x})} \tag{20a}$$

and

$$\overline{\mathbf{G}_n\mathbf{G}_n^T} = \begin{bmatrix} \boldsymbol{\phi}_n^T R_1 \boldsymbol{\phi}_n & \boldsymbol{\phi}_n^T R_{12}\boldsymbol{\phi}_n \\ \boldsymbol{\phi}_n^T R_{12}\boldsymbol{\phi}_n & \boldsymbol{\phi}_n^T R_2 \boldsymbol{\phi}_n \end{bmatrix}, \tag{20b}$$

where R_1, R_2 and R_{12} are auto- and cross-correlation matrices of $\mathbf{I}/\partial x_1$ and $\mathbf{I}/\partial x_2$, and

$$\nabla^T = \left[\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}\right].$$

Equation (19) can be interpreted in terms of an optimum (Wiener) displacement estimator. Consider the mean square nth coefficient prediction error resulting from a *given* displacement error $\boldsymbol{\Delta}_n(q) = \boldsymbol{\delta}$. To the linear approximation of (15), this is

$$F_n(\boldsymbol{\delta}) = \boldsymbol{\delta}^T\overline{\mathbf{G}_n\mathbf{G}_n^T}\boldsymbol{\delta}, \tag{21}$$

which is a quadratic function of the components of $\boldsymbol{\delta}$. A steepest descent algorithm for arriving at the minimum of the $F_n(\boldsymbol{\delta})$ for $n = 0, 1, \cdots, M-1$ is

$$\boldsymbol{\delta}_{i+1} = [U - \epsilon\overline{\mathbf{G}_{((i))}\mathbf{G}_{((i))}}]\boldsymbol{\delta}_i, \quad i = 0, 1, 2, \cdots. \tag{22}$$

Comparing (19) and (22), we see that, under the assumption of mutually independent $\mathbf{G}_{((i))}([[i]])$, the mean displacement estimation error $\bar{\boldsymbol{\Delta}}_i$ satisfies a recursion (22) that minimizes mean-square prediction

error of coefficients $n = 0, 1, \cdots, M-1$. The convergence of (19) and (22) is established below in Assertion 1.

We emphasize that recursion (19) describes the progression of mean displacement estimation error under the assumption of independent $\mathbf{G}_{((i))}([[i]])$. The conventional approach to transform image coding is to choose basis vectors $\{\phi_n\}$ so that the transmitted coefficients (or the prediction errors of these coefficients) are as mutually "independent as possible."[12] This is best achieved by choosing the basis vectors $\{\phi_n\}$ of the transform to be the eigenvectors of the covariance matrix (i.e., the Karhunen-Loeve basis) of the block of pels. This basis results in transform coefficients that are *linearly* independent (uncorrelated) within the transform block, although dependency among coefficients from block to block may persist. Other transforms, such as the cosine or Hadamard transforms can be viewed as practical approximations to the Karhunen-Loeve transform. Assuming that the Karhunen-Loeve-basis vectors result in coefficient *gradients* that are linearly independent as well, this would help justify the application of independence theory in describing the behavior of (18). In Appendix A we show that this assumption in indeed correct for the stochastic image model most widely applied in image processing analyses. Some insight into the behavior of (18) for dependent $\mathbf{G}_{((i))}([[i]])$ is given by the analysis in Appendix B.

Assertion 1: *Under the independence assumption, mean displacement estimation error in (19) converges to zero if and only if the eigenvalues of* Ψ *are inside the unit circle, where*

$$\Psi = \prod_{n=0}^{M-1} [U - \epsilon \overline{\mathbf{G}_n \mathbf{G}_n^T}]. \tag{23}$$

In our product notation (23), matrix $U - \epsilon \overline{G_0 G_0^T}$ *is premultiplied by* $U - \epsilon \overline{G_1 G_1^T}$, *etc.*

Proof: This can easily be shown by iterating (19) from $i = 0$ to $i = qM + n - 1$:

$$\begin{aligned} \bar{\Delta}_{qM+n} &= \left[\prod_{i=0}^{qM+n-1} [U - \epsilon \overline{\mathbf{G}_{((i))} \mathbf{G}_{((i))}^T}] \right] \bar{\Delta}_0 \\ &= [U - \epsilon \overline{\mathbf{G}_{n-1} \mathbf{G}_{n-1}^T}] \cdots [U - \epsilon \overline{\mathbf{G}_0 \mathbf{G}_0^T}] \Psi^q \bar{\Delta}_0. \end{aligned} \tag{24}$$

Therefore, the behavior of $\bar{\Delta}_{qM+n}$ as q increases depends upon the matrix Ψ as Ψ^q, and Assertion 1 follows. QED

A useful sufficient condition for convergence of (19) is given in the following.

Assertion 2: *Under the independence assumption and for any normalized set of basis vectors, mean displacement estimation error in (19) is bounded by*

$$\| \bar{\Delta}_{qM+n} \| \leq \kappa^q \| \bar{\Delta}_0 \| \tag{25}$$

where κ, $(0 < \kappa < 1)$ *is the maximum eigenvalue of matrices* $U - \epsilon G_n G_n^T$, $n = 0, 1, \cdots M-1$, *and*

$$0 < \epsilon < \frac{2}{N[\overline{(\partial I/\partial x_1)^2} + \overline{(\partial I/\partial x_2)^2}]}. \tag{26}$$

Proof: Using the Schwartz inequality on (24):

$$\| \bar{\Delta}_{qM+n} \| \leq \| U - \epsilon \overline{\mathbf{G}_{n-1}\mathbf{G}_{n-1}^T} \| \cdots \| U - \epsilon \overline{\mathbf{G}_0 \mathbf{G}_0^T} \| \| \Psi \|^q \| \Delta_0 \|, \tag{27}$$

where $\| (\cdot) \|$ for a symmetric matrix is the magnitude of the maximum magnitude eigenvalue. Similarly,

$$\| \Psi \| \leq \prod_{n=0}^{M-1} \| U - \epsilon \overline{\mathbf{G}_n \mathbf{G}_n^T} \|. \tag{28}$$

The eigenvalues of $U - \epsilon \overline{\mathbf{G}_n \mathbf{G}_n^T}$ have the form $1 - \epsilon \lambda_n^{(i)}$ where the $\lambda_n^{(i)}$, $i = 1, 2$, are the (nonnegative) eigenvalues of $\overline{\mathbf{G}_n \mathbf{G}_n^T}$. Let κ be the largest of the norms in (28) and let $\lambda_{\max}$ be the largest of the $\lambda_n^{(i)}$ for $0 \leq n \leq M-1$. Then excluding trivial cases, each matrix norm in (27) and (28) will be less than unity for $0 < \epsilon < 2/\lambda_{\max}$, and at least q of the norms in (27) is κ. Therefore,

$$\| \bar{\Delta}_{qM+n} \| \leq \kappa^q \| \bar{\Delta}_0 \| \tag{29}$$

for $0 < \epsilon < 2/\lambda_{\max}$.

Assume that $\lambda_{\max}$ corresponds to $\overline{\mathbf{G}_n \mathbf{G}_n^T}$ for $n = k$. Then using (20) and the fact that the eigenvalues of a nonnegative-definite matrix are bounded by the trace of the matrix gives the chain of inequalities

$$\begin{aligned} \lambda_{\max} &\leq Tr[\overline{\mathbf{G}_k \mathbf{G}_k^T}] \\ &\leq \phi_k^T R_1 \phi_k + \phi_k^T R_2 \phi_k \\ &\leq N[\overline{(\partial I/\partial x_1)^2} + \overline{(\partial I/\partial x_2)^2}]. \end{aligned} \tag{30}$$

Therefore ϵ in the range (26) guarantees $0 < \epsilon < 2/\lambda_{\max}$ which, in turn, guarantees (29). QED

Allowing for variations in scene statistics, a conservative choice of ϵ would be somewhat less than $2/\lambda_{\max}$. In this event, the following assertion applies.

Assertion 3: *If iteration is taken over all coefficients (i.e.,* $M = N$*) of any complete orthonormal basis* $\{\phi_n\}$, *and if* ϵ *is small compared to* $2/\lambda_{\max}$ *where* $\lambda_{\max}$ *is the maximum eigenvalue of matrices* $\{\mathbf{G}_n \mathbf{G}_n^T;\ n = 0. \cdots, N-1\}$, *then the block-to-block convergence of* $\bar{\Delta}$, *is independent of the particular basis set used. Furthermore, the convergence rate is independent of block dimensions* N_r *and* N_c.

Proof: The block-to-block dynamics of $\bar{\Delta}_i$ is determined by the matrix Ψ of (23). For $0 < \epsilon \ll 2/\lambda_{\max}$, and for $M = N$, Ψ can be approximated by

$$\Psi = U - \epsilon \sum_{n=0}^{N-1} \overline{\mathbf{G}_n \mathbf{G}_n^T}. \tag{31}$$

From (20a),

$$\sum_{n=0}^{N-1} \overline{\mathbf{G}_n \mathbf{G}_n^T} = \overline{\nabla \mathbf{I}^T(\mathbf{x}) \left[\sum_{n=0}^{N-1} \phi_n \phi_n^T \right] \nabla^T \mathbf{I}(\mathbf{x})}. \tag{32}$$

But since $\{\phi_n\}$ is a complete orthonormal set, $\sum_{n=0}^{N-1} \phi_n \phi_n^T = U$, the N-by-N identity matrix, and

$$\Psi = U - \epsilon \overline{\nabla \mathbf{I}^T(\mathbf{x}) \nabla^T \mathbf{I}(\mathbf{x})}, \tag{33}$$

which is independent of $\{\phi_n\}$.

The expectation in (33) is given by

$$\overline{\nabla \mathbf{I}^T(\mathbf{x}) \nabla^T \mathbf{I}(\mathbf{x})} = \begin{Bmatrix} \sum \overline{(\partial I/\partial x_1)^2} & \sum \overline{(\partial I/\partial x_1)(\partial I/\partial x_2)} \\ \sum \overline{(\partial I/\partial x_1)(\partial I/\partial x_2)} & \sum \overline{(\partial I/\partial x_2)^2} \end{Bmatrix}, \tag{34}$$

where the summations are over the N pels of a block. By stationarity, the expectation of each term in (34) is a constant and (33) becomes

$$\begin{aligned} \Psi &= U - \epsilon N \Gamma \\ &\approx (U - \epsilon \Gamma)^N, \end{aligned} \tag{35}$$

where

$$\Gamma = \begin{bmatrix} \overline{(\partial I/\partial x_1)^2} & \overline{(\partial I/\partial x_1)(\partial I/\partial x_2)} \\ \overline{(\partial I/\partial x_1)(\partial I/\partial x_2)} & \overline{(\partial I/\partial x_2)^2} \end{bmatrix}. \tag{36}$$

Therefore q block-to-block iterations of (19) premultiplies Δ_0 by $(U - \epsilon\Gamma)^{qN}$, which is a function only of the total number of iterations qN and is independent of basis and block dimension. QED

Experimental evidence of Assertion 3 is shown in Figs. 8a to 8d. Figure 8a shows the relevant component of displacement estimation error versus iteration number averaged over the interior of the moving cosine pattern of Fig. 6 for a pel-recursive (unity) 1×1 transform and a 1×8 Hadamard transform of the type in Fig. 7b using $\epsilon = 5 \times 10^{-5}$. For each scan line entering the average, iteration of (11) was initiated with displacement estimate $\hat{\mathbf{D}} = \mathbf{0}$ just inside the circumference of the pattern. In spite of the disparity between block size and transform type, the block-to-block convergence rate (measured over spans of eight iterations) of the Hadamard estimator closely matches that of the pel-recursive estimator. Although Assertion 2 concerns average displacement error, we found that it often applied as well to individual scan lines as shown in Figs. 8b to 8d. These figures show the relevant

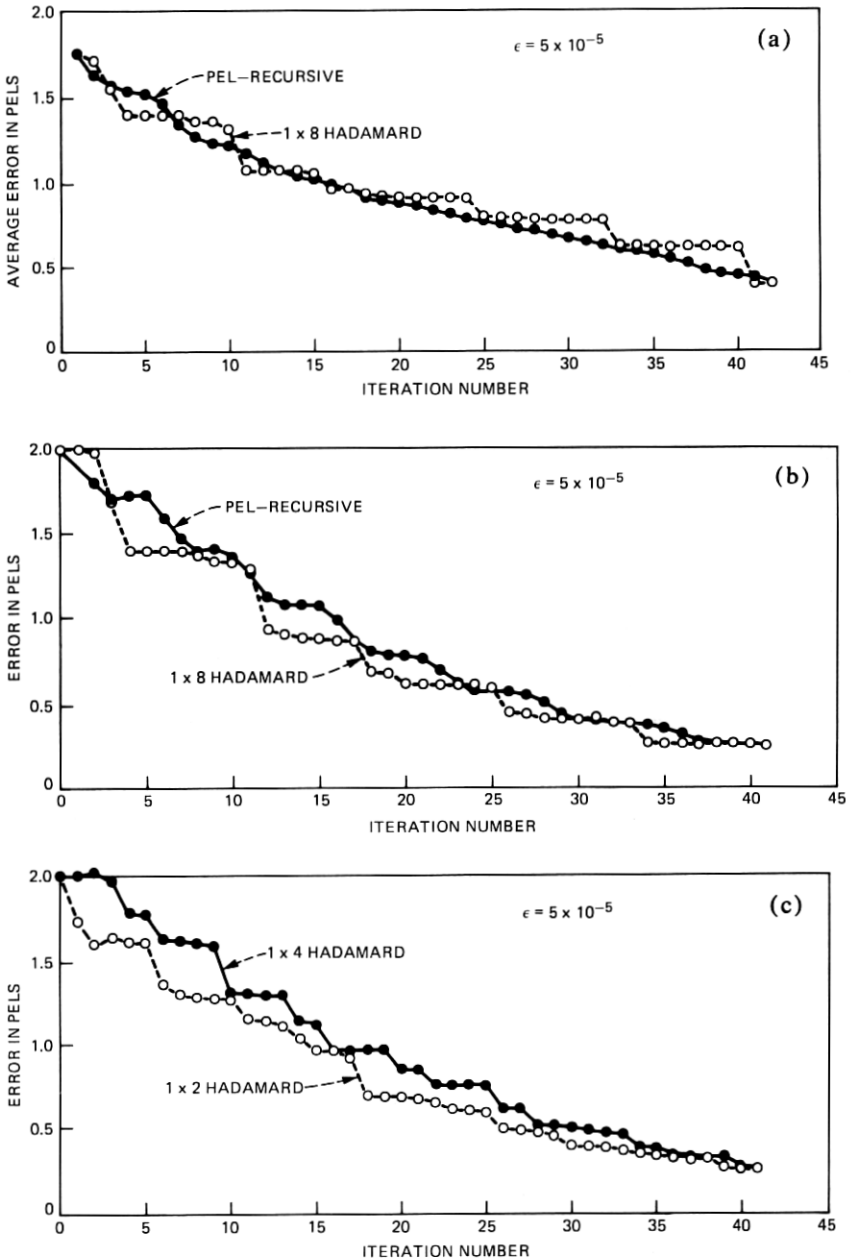

Fig. 8—Convergence results for various transforms with $\epsilon = 5 \times 10^{-5}$. (a) Pel-recursive and Hadamard 1×8 transform average relevant displacement estimation errors vs iteration number. (b) Pel-recursive and Hadamard 1×8 transform relevant displacement estimation errors for scan line through middle of moving cosine pattern. (c) Hadamard 1×4 and 1×2 transform relevant displacement estimation errors for scan line through middle of moving cosine pattern (*continued*).

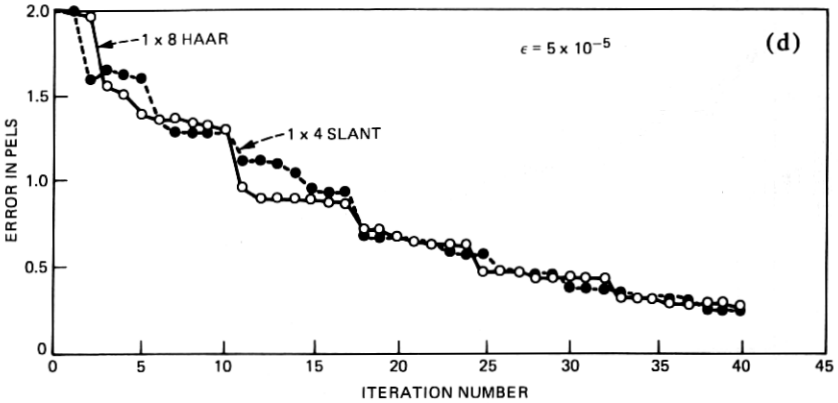

Fig. 8—(continued) (d) Haar 1 × 8 and Slant 1 × 4 transform relevant displacement estimation errors for scan line through middle of moving cosine pattern.

component of displacement estimation error versus iteration number for the scan line running through the middle of the moving cosine pattern. Hadamard 1 × 2, 1 × 4, 1 × 8, Slant 1 × 4, Haar 1 × 8, and pel-recursive estimators are all seen to have similar convergence rates with $\epsilon = 5 \times 10^{-5}$ when measured over the appropriate span of iterations. This also applies to the cosine transform (not shown), which was found to behave similarly to the Hadamard transform.

Figures 9a to 9c compare convergence of the 1 × 8 Hadamard block and pel-recursive displacement estimators as ϵ increases. The image data in this case was also the middle scan line of the moving cosine pattern. It can be seen that the convergence rates of the Hadamard and pel-recursive estimators are in rough agreement for increasing ϵ up to the point where oscillations occur (Fig. 9c). Note that oscillations occur in the Hadamard estimator before occurring in the pel-recursive estimator. This behavior was found to be the case for other transform types as well.

Although the block-to-block convergence rate of transform domain displacement estimators is substantially independent of the transform type, this is clearly not the case for *within* block convergence rate, as evidenced by Figs. 8 and 9. An explanation of this is given from the form of Ψ in (23), in which particular basis vector ϕ_n contributes a matrix factor of the form $[U - \epsilon \overline{\mathbf{G}_n \mathbf{G}_n^T}]$. This contribution of ϕ_n to reducing average displacement estimation error depends upon the eigenvalues of $\overline{\mathbf{G}_n \mathbf{G}_n^T}$ which are a measure of the statistical "match" between ϕ_n and the spatial rates of change of the scene.

It is possible to vary ϵ with n ($\epsilon = \epsilon_n$) to partially compensate for differences among the eigenvalues of matrices $\overline{\mathbf{G}_n \mathbf{G}_n^T}$. However, this technique can also make the algorithm move sensitive to noise that

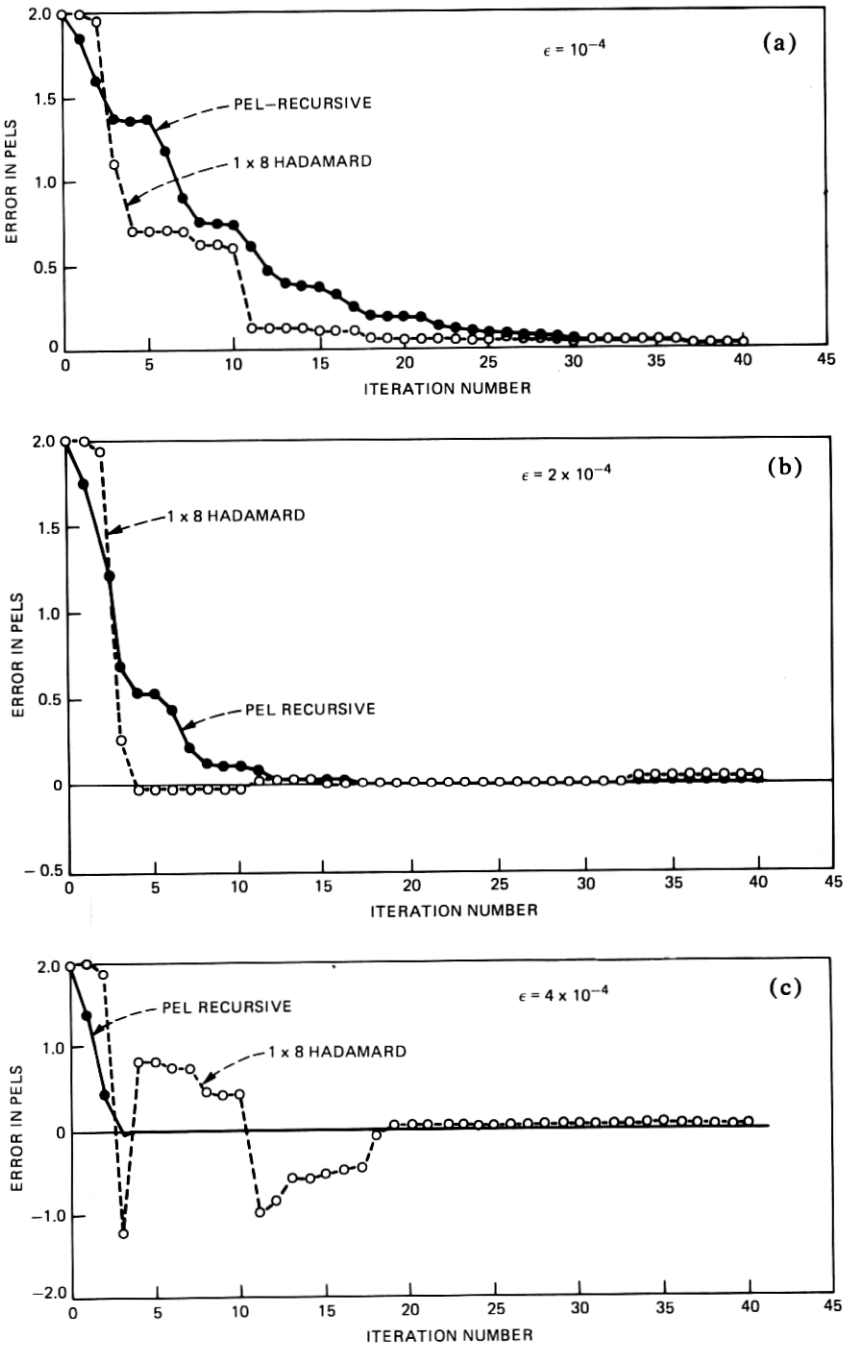

Fig. 9—Convergence results for pel-recursive and 1 × 8 Hadamard displacement estimators. (a) $\epsilon = 1 \times 10^{-4}$. (b) $\epsilon = 2 \times 10^{-4}$. (c) $\epsilon = 4 \times 10^{-4}$.

may be present in the image data. Section IV gives another approach that appears to have particularly good noise rejection properties.

IV. ADAPTATION

This section shows how the coefficient displacement estimation algorithm of Sections II and III can be improved by adaptively updating the displacement estimate according to the local features of the image. This is a technique that is not possible for the pel-recursive estimation. Adaptation in displacement estimation is motivated by the recognition that single frames of video are neither noiseless nor best described as stationary processes. Simulation results using noise-corrupted versions of the radial cosine object of Fig. 6 demonstrate that an adaptive algorithm of the type described here can have better convergence properties than either pel-recursive or nonadaptive coefficient recursive displacement estimation.

4.1 Preliminaries

The potential advantage of adaptation in (8) can be seen by considering the simple example of the moving edge scene of Fig. 10. This edge has constant slope $g = 3.8$ intensity increments per pel-to-pel distance over a width of 50 horizontal pel intervals, and velocity 2.7

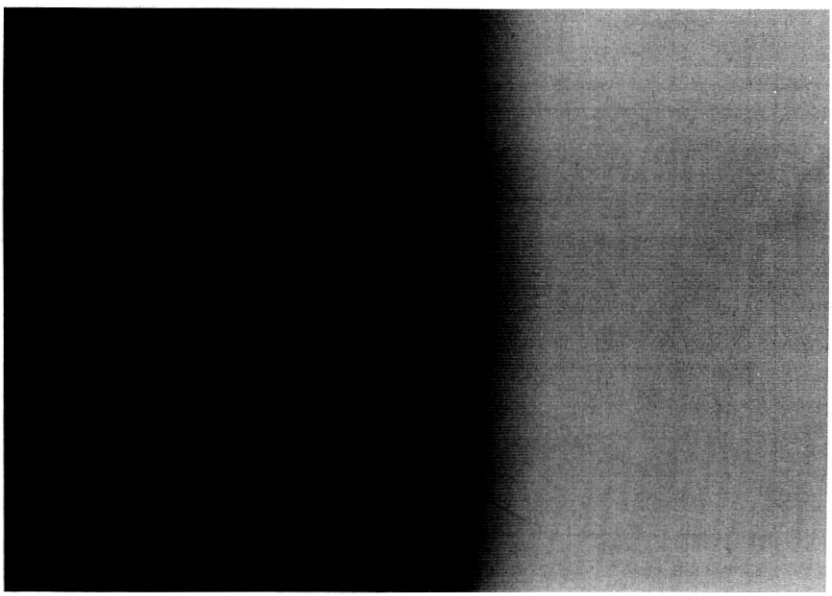

Fig. 10—Synthetic moving edge pattern.

pels horizontal per frame. Within the width of the edge, $\mathbf{G}_n(q)$ of (9) is given by

$$\mathbf{G}_n(q) = \begin{bmatrix} g\ g\ g\ \cdots\ g \\ 0\ 0\ 0\ \cdots\ 0 \end{bmatrix} \phi_n. \tag{37}$$

For a sequency-ordered basis set $\{\phi_n\}$, $\mathbf{G}_n(q)$ will be zero for all ac-basis vectors with the result that the corresponding displacement estimate updates will be determined solely by noise that may be present on the edge. Updating displacement estimate by iterating over these basis vectors can only increase estimation error. The dc-basis vector $(1/\sqrt{N})$, however, results in

$$\mathbf{G}_n(q) = \begin{pmatrix} g\sqrt{N} \\ 0 \end{pmatrix}, \tag{38}$$

which provides signal-dependent terms in (18) proportional to $\sqrt{N}$. This suggests that, for this example, it may be better to iterate repeatedly (say M times) over the dc-basis vector than to sequence through the M-basis vectors. We have not been able to analyze rigorously the performance of such dc-basis iteration on the moving edge in the presence of additive noise. In Appendix C we assume that the additive noise is white with power σ_w^2, and invoke certain assumptions regarding the independence of noise terms entering the recurrence relation. The result is the following approximate expressions for the horizontal component of displacement estimation error mean $\eta_\Delta(i)$ and steady-state variance σ_Δ^2.

$$\eta_\Delta(i) \sim \Delta(0)(1-\beta)^i; \quad i = 0, 1, 2, \cdots \tag{39a}$$

$$\sigma_\Delta^2 \sim 2\alpha(1+\alpha)\beta^2 \left\{ \frac{1}{2\beta - \beta^2} + \frac{\beta^2\alpha}{1-\beta^2\alpha} \right\}, \tag{39b}$$

where $\beta = \epsilon N g^2$ and $\alpha = \sigma_w^2/Ng^2$.

Not too surprisingly, expressions (39) indicate that, for constant rate of convergence (i.e., constant β), the steady-state displacement estimate error variance decreases inversely with block size N. This points out a possible advantage of transform domain displacement estimation compared to pel-recursive displacement estimation where the dc-basis block size is constrained to have dimension $N = 1$. Figures 11 through 13 show experimental and theoretical behavior of the horizontal component of displacement estimation error for pel-recursive and 2×8 dc-basis iteration on the moving edge. In obtaining the sample mean $\hat{\eta}_\Delta(i)$ and sample variance $\hat{\sigma}_\Delta^2$, averages were taken over 128 field lines.

Experimental and theoretical results are seen to compare favorably in the pel-recursive estimator case (Fig. 11) at a signal-to-noise ratio (SNR) = 45 dB. In Fig. 12, which describes displacement errors in the

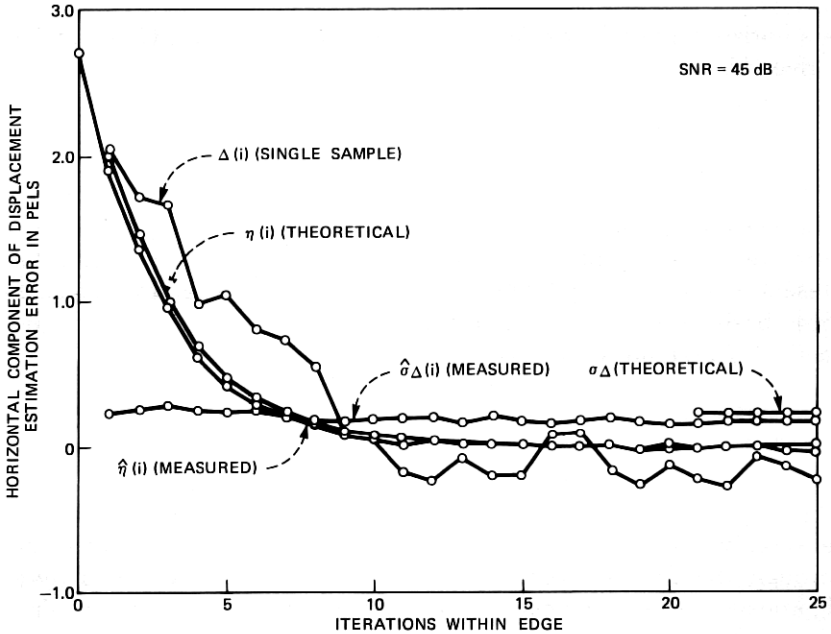

Fig. 11—Horizontal component of displacement estimation error for noisy moving edge (SNR = 45 dB) using pel-recursive displacement estimation. $\epsilon = 0.02$.

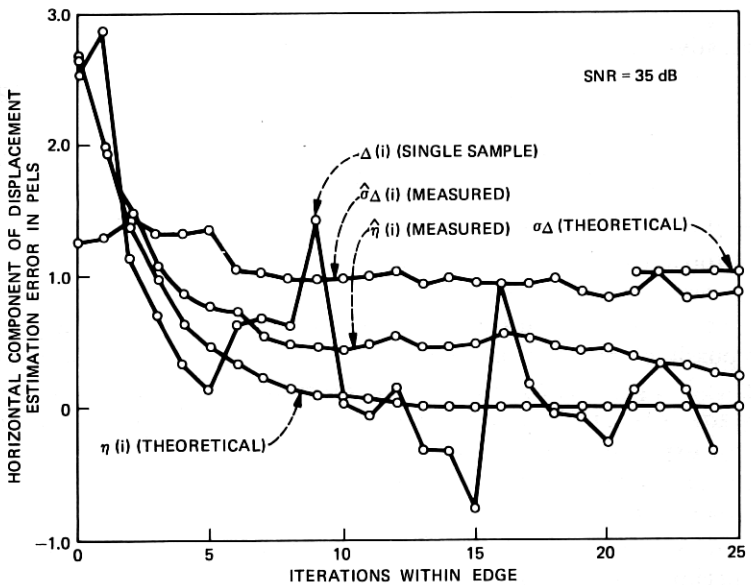

Fig. 12—Horizontal component of displacement estimation error for noisy moving edge (SNR = 35 dB) using pel-recursive displacement estimation. $\epsilon = 0.02$.

pel-recursive estimator at 35 dB, there is approximate agreement between theory and experimental data. This is also the case for the 2 × 8 dc-basis results of Fig. 13. From the experimental data in Figs. 12 and 13, it can be seen that the "dc-iteration" is more effective than pel-recursive estimation in combating the effects of noise in the displacement estimation of a moving edge.

The theoretical and experimental behavior of steady-state displacement error variance $\sigma_\Delta^2(\infty)$ versus β is plotted in Fig. 14 for pel-recursive displacement estimate for the moving edge pattern with SNR = 35 dB. The range $1 < \beta < 2$ represents oscillatory convergence with $\sigma_\Delta^2(\infty)$ increasing rapidly as β approaches 2. Note the trade-off between convergence rate and accuracy evidenced by Fig. 14 and eq. (39a).

4.2 Adaptation algorithm

The adaptive displacement estimation algorithm proposed here updates the displacement estimate (8) using that basis vector ϕ_{m0} whose projection onto the computed coefficient gradient of the reference frame has maximum amplitude. At each iteration step (n, q), the magnitude of the coefficient gradient vector $\| \mathbf{G}_n(q) \|$ is computed from

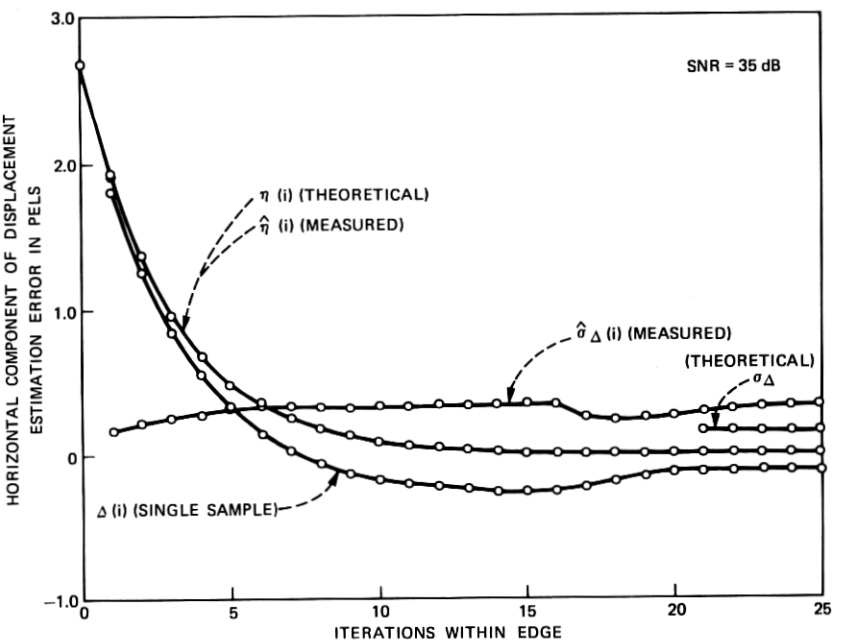

Fig. 13—Horizontal component of displacement estimation error for noisy moving edge (SNR = 35 dB) using 2 × 8 dc basis iteration. $\epsilon = 0.00125$. The factor ϵ was adjusted in this experiment to result in an identical average convergence rate as that of Figs. 11 and 12.

the *noisy* previous frame data for each vector in the given basis set $\{\phi_m;\ m = 0, 1, \cdots M - 1\}$. The particular basis ϕ_{m0} maximizing this quantity is then used for the displacement update.

Figure 15 compares the performance of adaptive and nonadaptive displacement estimation using a separable 2×4 cosine transform and $\epsilon = 10^{-5}$ for the test image of Fig. 16. This image consists of the moving

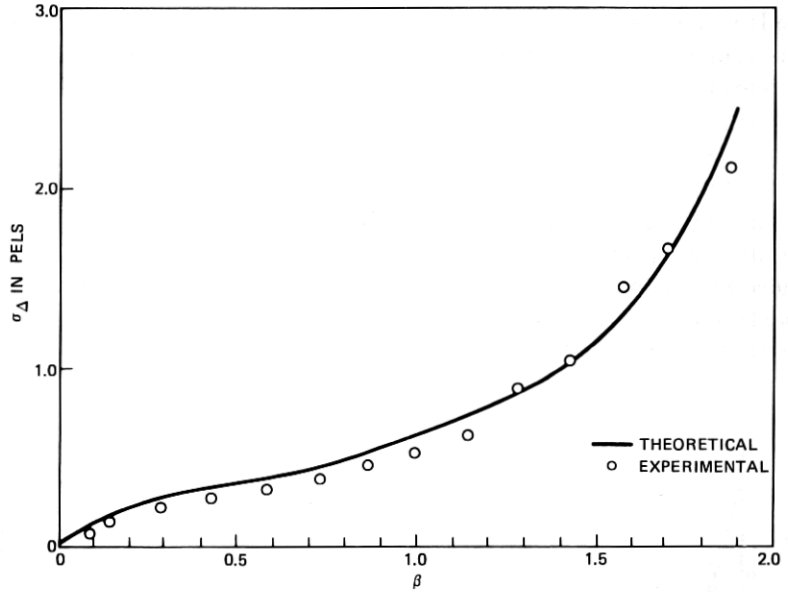

Fig. 14—Estimation error standard deviation vs rate parameter β: pel-recursive at SNR = 45.

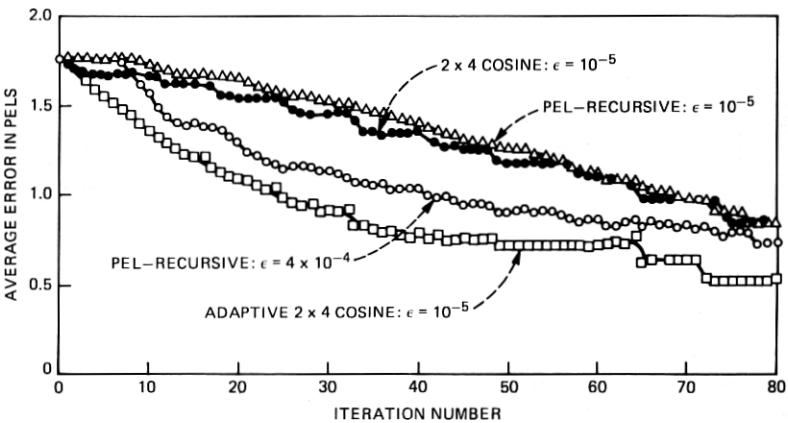

Fig. 15—Displacement estimation errors for noisy synthetic image of Fig. 16.

object of Fig. 6 corrupted by additive white Gaussian noise at 20-dB SNR. Also shown are results for pel-recursive displacement estimation at $\epsilon = 10^{-5}$ and a convergence-rate-optimized $\epsilon = 4 \times 10^{-5}$. The pel-recursive scanning pattern was chosen according to Fig. 17 to match the rate of progression of the pel-recursive algorithm along individual scan lines to that of the 2×4 transform algorithm. (For example, after

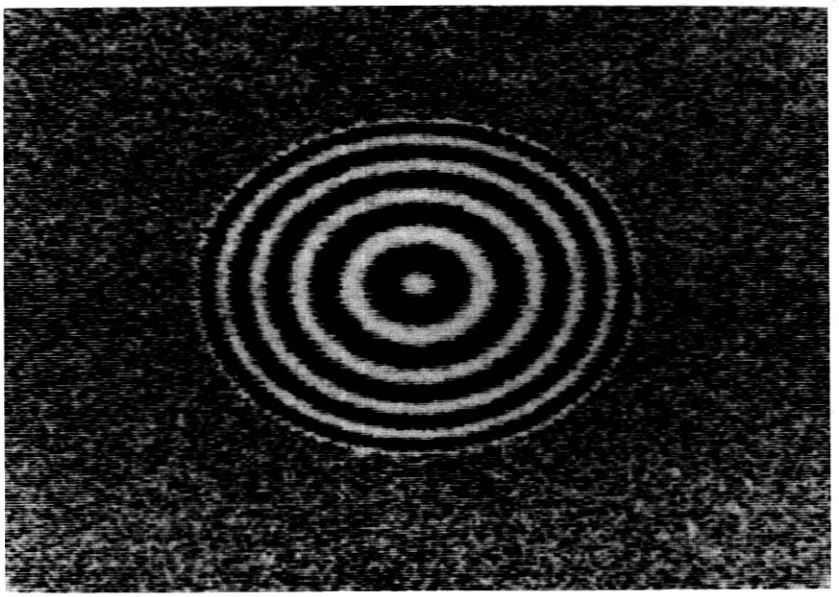

Fig. 16—Noisy test image. SNR = 20 dB.

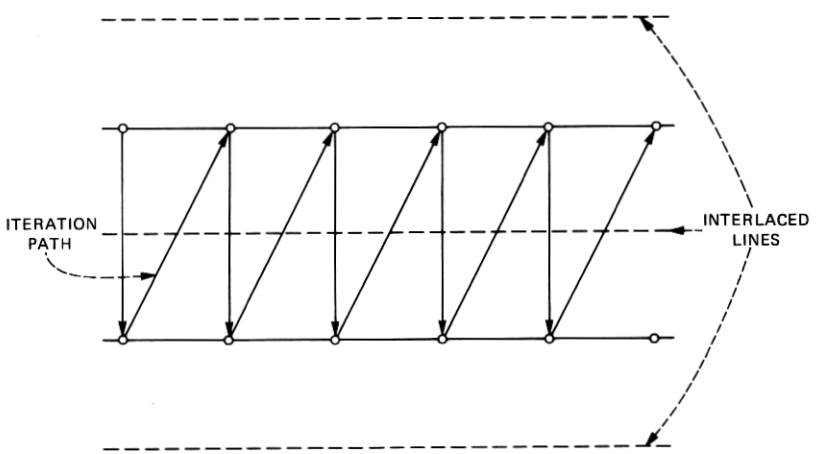

Fig. 17—Pel-recursive scanning pattern applicable to test results of Fig. 15.

40 iterations, both pel-recursive and coefficient-recursive displacement estimators will have traversed 20 columns of a given scan line.) All results of Fig. 15 are averages of relevant displacement error over the interior of the moving object. It can be seen from Fig. 15 that the adaptive 2×4 cosine displacement estimator has clearly superior average error convergence rate than either the pel-recursive or nonadaptive cosine displacement estimators. No choice of ϵ was found that could improve the convergence rate of the pel-recursive estimator beyond that shown for $\epsilon = 4 \times 10^{-5}$. We also computed the experimental standard deviations of relevant displacement estimation error. Nonadaptive cosine and pel-recursive estimators had experimental error standard deviations of 0.50 and 0.51, respectively, at $\epsilon = 10^{-5}$. The error standard deviation of the adaptive coefficient-recursive displacement estimator was 0.54, while that for the rate-optimized pel-recursive estimator was an inferior 0.67.

The above results demonstrate a potential advantage of adaptive coefficient-recursive displacement estimation over both pel-recursive and nonadaptive coefficient-recursive displacement estimation. It remains to be established, however, whether the adaptive scheme of this section will improve the performance of a motion-compensated hybrid transform-DPCM coder.

V. SUMMARY

This paper has introduced a coefficient-recursive displacement estimator having potential application in motion-compensated interframe hybrid transform-DPCM image coders. The convergence of the mean displacement estimate to the true displacement was established in Assertions 1 and 2 using assumptions that are supported by the analyses of Appendices A and B. Assertion C described conditions under which the rate of convergence of mean displacement estimation error is independent of the transform block size and type. An extension of the coefficient recursive algorithm was given in Section IV and shown by simulation to have improved convergence properties in the displacement estimation of noisy objects.

APPENDIX A

This appendix verifies a statement in Section III concerning the orthogonality of coefficient vectors $\mathbf{G}_n(q)$ $n = 0, 1, \cdots M - 1$ for a separable Markov image model. This model is described as follows. Let $I_{m,n}$ denote the intensity of the pel located in the mth row and nth column of the raster. Then for the Markov image model treated here:

$$\overline{I_{m,n}I_{i,j}} = \sigma^2 \rho_r^{|m-i|} \rho_c^{|n-j|}, \tag{40}$$

with $0 < |\rho_r| < 1$, $0 < |\rho_c| < 1$. In (40), σ^2 is the intensity variance, and ρ_r and ρ_c are the correlation coefficients of adjacent pels between rows and columns, respectively. If the pel intensities in a N_r row by N_c column block are indexed in column scan fashion and denoted by a column vector $\mathbf{I}$, then it can be shown that

$$\overline{I \cdot I^T} = \sigma^2 M_r \times M_c, \tag{41}$$

where $\times$ denotes the Kronecker product,[13] and M_r and M_c are, respectively, $N_r \times N_r$ and $N_c \times N_c$ Toeplitz matrices having ijth entries

$$[M_r]_{ij} = \rho_r^{|i-j|}$$

and

$$[\mathbf{M}_c]_{ij} = \rho_c^{|i-j|}. \tag{42}$$

Covariance models of this form have been widely applied in image processing studies.[14-18] Expressions for the eigenvalues and eigenvectors of M_r (or M_c) are given in Ref. 18.

We now apply this model to a study of the covariance properties of $G_n(q)$, $n = 0, 1, \cdots M-1$. From (9) we have

$$\overline{\mathbf{G}_n(q)\mathbf{G}_m^T(q)} = \begin{bmatrix} \phi_n^T R_1 \phi_m & \phi_n^T R_{12} \phi_m \\ \phi_n^T R_{12} \phi_m & \phi_n^T R_2 \phi_m \end{bmatrix}. \tag{43}$$

As described in Section III, matrices R_1, R_2, *and* R_{12} are auto- and cross-correlation matrices of the spatial derivatives of I. For the discrete image model specified by (40), we compute these spatial derivatives as the corresponding spatial differences. For example, derivatives in the row direction of the raster are given by $\overline{(I_{m,n} - I_{m,n-1})(I_{i,j} - I_{i,j-i})}$. Expanding this product and using (40) gives

$$\overline{(I_{m,n} - I_{m,n-1})(I_{i,j} - I_{i,j-i})} = \sigma^2 \rho_r^{|m-i|} [\alpha_c \rho_c^{|n-j|} + \beta_c \delta_{nj}], \tag{44}$$

where $\alpha_c = 2 - \rho_c^{-1} - \rho_c$, $\beta_c = \rho_c^{-1} - \rho_c$, and δ_{nj} is the Kronecker delta function. By comparing (40) and (41) with (44), we have

$$R_1 = \sigma^2 M_r \times [\alpha_c M_c + \beta_c U], \tag{45}$$

where U is an identity matrix.

Similarly, with $\alpha_r = 2 - \rho_r^{-1} - \rho_r$ and $\beta_r = \rho_r^{-1} - \rho_r$:

$$R_2 = \sigma^2 [\alpha_r M_r + \beta_r U] \times M_c \tag{46}$$

and

$$R_{12} = \sigma^2 [\alpha_r M_r + \beta_r U] \times [\alpha_c M_c + \beta_c U]. \tag{47}$$

As shown in Ref. 13, the eigenvectors of a Kronecker product $A \times B$ have the form

$$\begin{bmatrix} x_1^i \mathbf{y}^j \\ x_2^i \mathbf{y}^j \\ \cdot \\ \cdot \\ \cdot \\ x_N^i \mathbf{y}^j \end{bmatrix},$$

where x_k^i, $k = 1, 2, \cdots, N$, denotes the components of the eigenvector $\mathbf{x}^i$ of A and $\mathbf{y}^j$ denotes an eigenvector of B. Since any vector is an eigenvector of an identity matrix, it follows that the eigenvectors of R_1, R_2, and R_{12} above are in fact identical to the eigenvectors of the image covariance matrix (41). The normalized eigenvectors form a complete *orthonormal* set and are considered to be the optimum bases for transform coding image blocks modeled by (41). Selecting $\{\phi_n\}$ to be this set of eigenvectors, it follows that all terms in the matrix of (43) will be identically zero for $n \neq m$, which establishes the statistical orthogonality of the $\mathbf{G}_n(q)$.

APPENDIX B

This appendix shows that mean displacement estimation error for dependent $\mathbf{G}_{((i))}([[i]])$ is approximately given by recursion (19) for small ϵ.

Iterating (18) yields

$$\mathbf{\Delta}_{i+1} = \left(\prod_{j=0}^{i} [U - \epsilon \mathbf{G}_{((j))}([[j]]) \mathbf{G}_{((j))}^T([[j]])] \right) \mathbf{\Delta}_0. \tag{48}$$

The matrix product premultiplying $\mathbf{\Delta}_0$ in (48) is a function of ϵ. Taylor's expansion of this function about $\epsilon = 0$ yields (for fixed i):

$$\mathbf{\Delta}_{i+1} = \left(U - \epsilon \prod_{j=0}^{i} \mathbf{G}_{((j))}([[j]]) \mathbf{G}_{((j))}^T([[j]]) \right) \mathbf{\Delta}_0 + O(\epsilon^2) \tag{49}$$

so that

$$\bar{\mathbf{\Delta}}_{i+1} = \left(U - \epsilon \sum_{j=0}^{i} \overline{\mathbf{G}_{((j))} \mathbf{G}_{((j))}^T} \right) \bar{\mathbf{\Delta}}_0 + \bar{O}(\epsilon^2). \tag{50}$$

On the other hand, repeating the steps of (48) and (49) on (19) gives (for the independent $\mathbf{G}_{((j))}([[i]])$ case):

$$\bar{\mathbf{\Delta}}_{i+1} = \left(I - \epsilon \sum_{j=0}^{i} \overline{\mathbf{G}_{((j))} \mathbf{G}_{((j))}^T} \right) \bar{\mathbf{\Delta}}_0 + O'(\epsilon^2). \tag{51}$$

Comparing (50) and (51) we have the result that mean estimation errors for the dependent and independent cases are equal to an

approximation $\bar{O}(\epsilon^2) - O'(\epsilon^2)$, which can be neglected for sufficiently small ϵ.

APPENDIX C

This appendix briefly sketches the steps leading to (39).

We consider the successive frames of video to consist of a moving edge contaminated by independent noise:

$$I(\mathbf{x}, \tau) = g(x_1 - D) + w(\mathbf{x}, t)$$
$$I(\mathbf{x}, t - \tau) = gx_1 + w(\mathbf{x}, t - \tau), \tag{52}$$

where the noise $w(\cdot, \cdot)$ is white with respect to both pel-to-pel and frame-to-frame dimensions. An analysis similar to that leading to (18) then yields

$$\Delta_{i+1} = \Delta_i - \epsilon[\mathbf{G}_n(q) + \nabla[\mathbf{W}^T(\mathbf{x}_{[[i]]} - \hat{\mathbf{D}}_{i,t} - \tau)\phi_n]]$$
$$\cdot[\mathbf{G}_n^T(q)\Delta_i + \mathbf{W}^T(\mathbf{x}_{[[i]]}, t)\phi_n - \mathbf{W}^T(\mathbf{x}_{[[i]]} - \hat{\mathbf{D}}_i, t - \tau)\phi_n]. \tag{53}$$

We consider iterating (53) repeatedly, using dc-basis vector $\phi_0 = 1/\sqrt{N}$ and $\mathbf{G}_n(q)$ given by (38). Let $\Delta(i)$ denote the displacement error in the horizontal direction at iteration i; $Z_1(i)$ denote the difference of the two noise terms in the final term of (53); and $Z_2(i)$ denote the horizontal component of the noise gradient term in (53). Then the horizontal component of (53) becomes

$$\Delta(i + 1) = \Delta(i) - \epsilon Ng^2[\Delta(i) + Z_1(i)/\sqrt{N}g][1 + Z_2(i)/\sqrt{N}g]. \tag{54}$$

Defining $\beta = \epsilon Ng^2$, $\gamma = 1/\sqrt{N}g$ and rearranging terms yield

$$\Delta(i + 1) = [1 - \beta(1 + \gamma Z_2(i))]\Delta(i) - \beta\gamma(1 + \gamma Z_2(i))Z_1(i). \tag{55}$$

Note that each noise term in (55) is multiplied by a factor γ which, for given β, decreases as $1/\sqrt{N}$. Neglecting dependencies between $\{Z_2(i)\}$, and $\{Z_1(i)\}$, this equation is linear with respect to input Z_1 and output Δ. The solution has the form

$$\Delta(i) = \Delta(0) \prod_{j=1}^{i-1} [1 - \beta(1 + \gamma Z_2(j))] + \sum_{k=0}^{i} h(i, k)Z_1(k), \tag{56}$$

where $h(i, k)$ is the response of (55) for $\Delta(0) = 0$ and input $Z_1(i) = \delta_{ik}$. By assuming that $\{Z_1(i)\}$ and $\{Z_2(i)\}$ are mutually independent white sequences, the mean and variance of $\Delta(i)$ can now be derived by a tedious but conventional analysis, resulting in (39).

REFERENCES

1. A. N. Netravali and J. D. Robbins, "Motion-Compensated Television Coding: Part I," B.S.T.J., *58*, No. 3 (March 1979), pp. 631–670.

2. J. D. Robbins and A. N. Netravali, "Interframe Coding Using Movement Compensation," Int. Conf. on Commun., June 1979.
3. D. O. Reudink, unpublished work.
4. A. Habibi, "Hybrid Coding of Pictorial Data," IEEE Trans. Commun., *COM-22*, No. 5 (May 1974), pp. 614–624.
5. C. Reader, "Orthogonal Transform Coding of Still and Moving Pictures," Ph.D. dissertation, The University of Sussex, United Kingdom, 1974.
6. 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.
7. H. W. Jones, "A Conditional Replenishment Hadamard Video Compressor," SPIE, *119*, Applications of Digital Image Processing, 1977, pp. 91–98.
8. A. N. Netravali and J. A. Stuller, "Motion-Compensated Transform Coding," B.S.T.J., this issue, pp. 1703–1718.
9. S. C. Knauer, "Real-Time Video Compression Algorithm for Hadamard Transform Processing," Proceedings of SPIE, *66* (August 1975), pp. 58–69.
10. H. C. Andrews and C. L. Patterson, "Outer Product Expansions and Their Uses in Digital Image Processing," American Math. Monthly. *82*, No. 1 (January 1974), pp. 1–13.
11. B. Widrow, "Adaptive Filters," in *Aspects of Network and System Theory*, R. Kalman and N. DeClaris, eds., New York: Holt, Rinehart, and Winston, 1971, pp. 563–587.
12. P. A. Wintz, "Transform Picture Coding," Proc. IEEE, *60* (July 1972), pp. 809–820.
13. R. Bellman, *Introduction to Matrix Analysis*, 2nd ed., New York: McGraw-Hill, 1970, pp. 235.
14. A. Arcese, P. H. Mengert, and E. W. Trombini, "Image Detection through Bipolar Correlation," IEEE Trans. Inform. Theory, *IT-16*, No. 5 (September, 1970), pp. 534–541.
15. A. K. Jain and E. Angel, "Image Restoration, Modelling, and Reduction of Dimensionality," IEEE Trans. Computers, *C-21* (July 1972), pp. 636–641.
16. J. A. Stuller and B. Kurz, "Two-Dimensional Markov Representations of Sampled Images," IEEE Trans. Commun., *COM-24* (October 1976), pp. 1148–1152.
17. J. A. Stuller and B. Kurz, "Interframe Sequential Picture Coding," IEEE Trans. Commun., *COM-25*, No. 5 (May 1977), pp. 485–495.
18. W. K. Pratt, "Generalized Wiener Filtering Computation Techniques," IEEE Trans. Computers, *C-21*, No. 7 (July 1972), pp. 636–641.