Displacement Measurement and Its Application in Interframe Image Coding

JASWANT R. JAIN, MEMBER, IEEE, AND ANIL K. JAIN

Abstract—**A new technique for estimating interframe displacement of small blocks with minimum mean square error is presented. An efficient algorithm for searching the direction of displacement has been described. The results of applying the technique to two sets of images are presented which show 8–10 dB improvement in interframe variance reduction due to motion compensation. The motion compensation is applied for analysis and design of a hybrid coding scheme and the results show a factor of two gain at low bit rates.**

I. INTRODUCTION

A LARGE number of image transmission and storage applications, e.g., teleconferencing, videotelephone, television and satellite image transmission, medical imaging for computer aided tomography and angiocardiography, etc., contain images of moving objects. The motion captured in such a multiframe sequence of images includes translation and rotation of objects with respect to the camera.

For interframe image coding, large levels of compression could be achieved if only one knew the trajectories traversed by the various objects. Then one could simply code the initial frame together with the trajectory information of each pixel. In practice, a significant component of the motion in a scene can be approximated by piecewise translation of several areas of a frame with respect to a reference frame. Utilization of the knowledge of motion or displacement of pixels in successive frames for image coding is called motion compensation.

Displacement measurement and motion compensation have been applied for interframe image data compression with improved results [1]-[8]. Limb and Murphy [1] and Rocca, Brofferio *et al.* [2]-[4] have considered techniques for estimating translation of a block of pixels. Netravali and Robbins [5] take the approach of predicting the displacement of each pixel recursively from its neighboring pixels which have already been coded. In this paper we present a method of displacement measurement which estimates displacement on a block by block basis. Application of this method in interframe hybrid coding (with and without frame skipping and interpolation) is shown. The results presented here are based on [6] and [7] and the algorithm developed is quite different from other displacement measurement techniques including the one reported in [8]. A detailed bibliography and discussion of other interframe coding techniques are given in [9] and [10].

A new technique for displacement measurement is described in Section II. This technique is based on an efficient 2-dimensional search procedure. The results of applying this technique for measurement of the displacement on two sets of images are reported in Section III. We find 8-10 dB reduction in variance of the interframe difference signal as a result of motion compensation. Motion compensation is applied for analysis and design of interframe hybrid coding methods in Section IV. Summary and conclusions are presented in Section V. In the Appendix we give a proof of the convergence of the search algorithm.

II. A DISPLACEMENT MEASUREMENT ALGORITHM

In this section, we describe a method of measuring interframe motion for digitized images. First, we approximate the interframe motion by piecewise translation of one or more areas of a frame relative to a reference frame. The segmentation of an image into areas, each of which is undergoing approximately the same translation, and the measurement of the magnitude and the direction of the translation of each area is a difficult task. Cafforio and Rocca [3] describe a method for segmentation and measurement of the displacement of a single moving object in a stationary background. Then, extension of the method to more than one moving object has also been shown. The method becomes increasingly complex as the number of moving areas increases and the size of the image grows larger, since the information concerning segmentation as well as translation is to be coded. Coding of segments with arbitrary boundaries increases the complexity as well as the length of the code to be utilized.

A simpler method is to segment an image into fixed size, small rectangular blocks and to assume that each of these areas is undergoing independent translation. If these areas are small enough, rotation, zooming, etc., of larger objects can be closely approximated by piecewise translation of these smaller areas. It avoids the problem of coding the segmentation information and only the displacement vector of each block needs to be coded.

A method which has been used for the measurement of displacement between two given images, particularly for aerial guidance, is area correlation [11], [12]. This consists of calculating the cross-correlation function of the two images. The location of the peak of the correlation function gives the displacement vector. The cross-correlation function is usually calculated via the fast Fourier transform (FFT). To improve the accuracy of this method some high-pass filtering (which

Manuscript received January 5, 1981; revised August 28, 1981. This paper was presented at the Picture Coding Symposium, Ipswich, England, July 1979.

J. R. Jain is with the Defense Division, Systems Control, Inc., Palo Alto, CA 94304.

A. K. Jain is with the Signal and Image Processing Laboratory, Department of Electrical and Computer Engineering, University of California, Davis, CA 95616.

0090-6778/81/1200-1799$00.75 © 1981 IEEE

could be done in the spatial domain [11] or the Fourier domain [12]) of the images is useful. We have found that the accuracy of the area correlation method is poor when the block size is small and the blocks are not undergoing pure translation. The method presented here performs significantly better under most circumstances for interframe image displacement estimation. This method requires a search for the *direction of minimum distortion* (or DMD) and is described below.

First, the image is divided into smaller rectangular areas, which we call subblocks. Let U be an $M \times N$ size subblock of an image and U_R be an $(M + 2p) \times (N + 2p)$ size subblock of a reference (neighboring) image, centered at the same spatial location as U, where p is the maximum displacement allowed in either direction in integer number of pixels.

Let us define a mean distortion function between U and U_R as

$$D(i,j) = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} g(u(m,n) - u_R(m+i, n+j)), \quad -p \leqslant i,j \leqslant p \tag{1}$$

where $g(x)$ is a given positive and increasing distortion function of x, e.g., $g(x) = x^2$ would correspond to $D(i,j)$ as mean square error function. The direction of minimum distortion is given by (i,j), such that $D(i,j)$ is minimum.

One difficulty with finding the DMD as stated above is that it requires evaluation of $D(i,j)$ for $(2p + 1) \times (2p + 1)$ directions, and even for motions up to 5 pixels along either side of the axes one has to search 121 directions. A solution to overcome this difficulty is found by making the assumption that if

$$D_0(q,l) = \min_{i,j} \{D(i,j)\} \tag{2}$$

then for $m = i - q$, $n = j - l$, the functions

$$\begin{aligned} D_1(|m|,|n|) &= D(i,j) - D_0(q,l), \quad m \geqslant 0, \quad n \geqslant 0 \\ D_2(|m|,|n|) &= D(i,j) - D_0(q,l), \quad m \geqslant 0, \quad n \leqslant 0 \\ D_3(|m|,|n|) &= D(i,j) - D_0(q,l), \quad m \leqslant 0, \quad n \leqslant 0 \\ D_4(|m|,|n|) &= D(i,j) - D_0(q,l), \quad m \leqslant 0, \quad n \geqslant 0 \end{aligned}$$

are increasing functions of both $|m|$ and $|n|$, i.e., for $1 \leqslant k \leqslant 4$,

$$\begin{aligned} D_k(|m|,|n|) &< D_k(|m'|,|n'|), \\ &\text{if } |m| < |m'| \text{ and } |n| \leqslant |n'| \\ &\text{or } |m| \leqslant |m'| \text{ and } |n| < |n'| \end{aligned} \tag{3}$$

where $|\cdot|$ represents the absolute value. It means that the distortion function monotonically increases as we move away from the DMD along any direction in each of the four quadrants. For $g(x) = x^2$, the above would be satisfied if the covariance function of images is a decreasing function of displacement in each of the quadrants. Most image covariance functions satisfy this condition, at least in a neighborhood extending to several pixels. Fig. 1 shows the locations of various distortion functions. We point out that the condition in (3) does not require $D_4(m_1, n_1)$ to be less than $D_4(m_2, n_2)$ or vice versa, even if they are equidistant from D_0. All that it requires is that $D_4(m_2, n_1)$ be less than $D_4(m_1, n_1)$ and $D_4(m_2, n_2)$ (see Fig. 1).

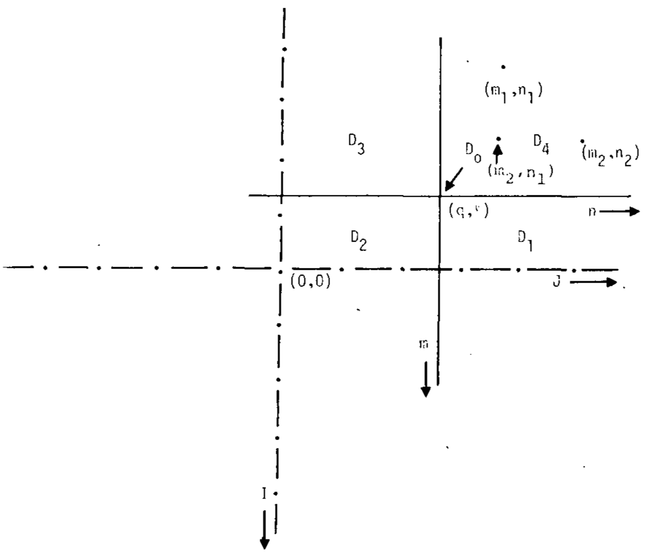


Fig. 1. Illustration of distortion functions.

With the above assumption, we use a 2-D directed search method, which is an extension of the binary or logarithm search [13] in one dimension. The search is accomplished by successively reducing the area of search. Each step consists of searching five locations which contain the center of the area, and the midpoints between the center and the four boundaries of the area along the axes passing through the center. This procedure continues until the plane of search reduces to a 3×3 size. In the final step all the nine locations are searched and the location corresponding to the minimum is the DMD. The algorithm is given below.

For any integer $m > 0$, we define

$$\begin{aligned} N(m) &= \{(i,j); \quad -m \leqslant i,j \leqslant m\} \\ M(m) &= \{(0,0), (m,0), (0,m), (-m,0), (0,-m)\}. \end{aligned} \tag{4}$$

A 2-D Logarithmic Search Procedure for DMD:

Step 1: (initialization)

$$\begin{aligned} D(i,j) &= \infty \qquad (i,j) \notin N(p) \\ n' &= \lfloor \log_2 p \rfloor \\ n &= \max \cdot \{2, 2^{n'-1}\} \\ q &= l = 0 \text{ (or an initial guess for DMD)} \end{aligned}$$

where $\lfloor \cdot \rfloor$ is a lower integer truncation function.

Step 2: $M'(n) \leftarrow M(n)$.

Step 3: Find $(i,j) \in M'(n)$ such that $D(i + q, j + l)$ is

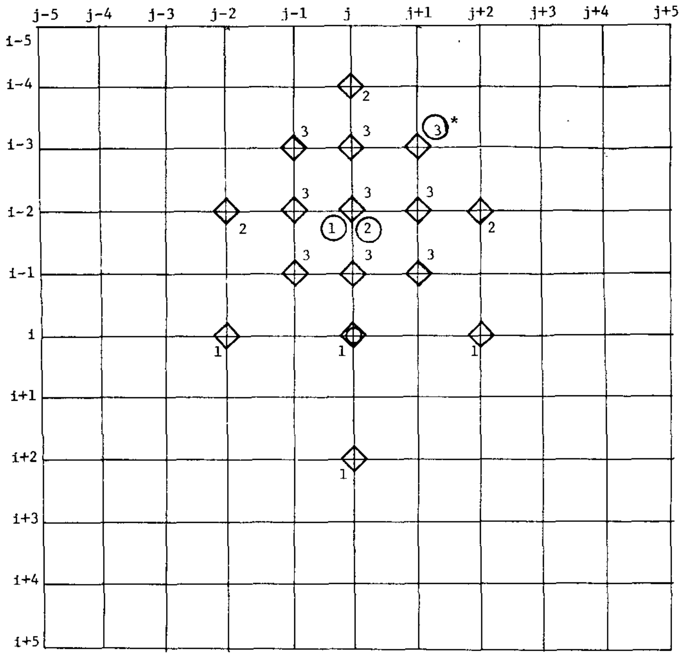


Fig. 2. A 2-D logarithmic search procedure for the direction of minimum distortion. The figure shows the concept of the 2-D logarithmic search to find a pixel in another frame which is registered with respect to the pixel (i, j) of a given frame, such that the mean-square error over a block defined around (i, j) is minimized. The search is done step by step with ◇ indicating the directions searched at a step number marked. The numbers circled show the optimum directions for that search step and the * shows the final optimum direction, $(i - 3, j + 1)$ in the above example. This procedure requires only searching 13 to 21 locations for the above grid as opposed to 121 total possibilities.

minimum. If $i = 0$ and $j = 0$, go to Step 5; otherwise go to Step 4.

Step 4: $q \leftarrow q + i, l \leftarrow l + j$; $\mathcal{M}'(n) \leftarrow \mathcal{M}'(n) - (-i, -j)$; go to Step 3.

Step 5: $n \leftarrow n/2$. If $n = 1$, go to Step 6; otherwise, go to Step 2.

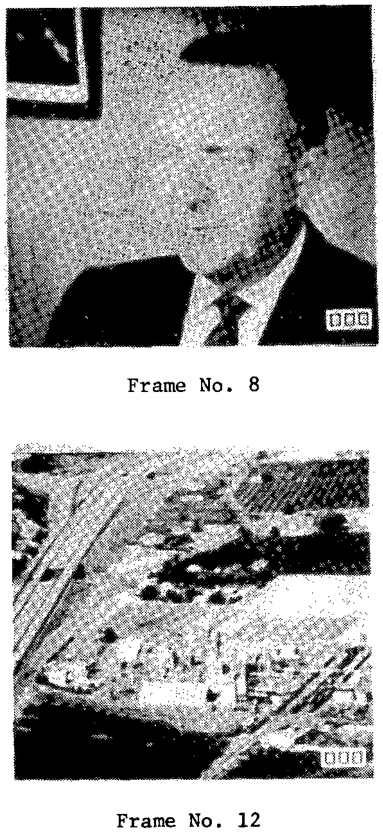
Step 6: Find $(i, j) \in \mathcal{N}(1)$ such that $D(i + q, j + l)$ is minimum: $q \leftarrow i + q, l \leftarrow l + j$. (q, l) then gives the DMD.

Fig. 2 illustrates the search procedure for $p = 5$. A proof of the search algorithm is given in the Appendix.

If the direction of minimum distortion lies outside $\mathcal{N}(p)$, the algorithm would converge to a point on the boundary which is closest to the DMD if the condition of (3) is satisfied.

III. EFFECTS OF MOTION COMPENSATION ON INTERFRAME VARIANCE

The method of DMD displacement measurement discussed above was applied to the "Cronkite" and "Chemical Plant" image sets, each containing 16 frames of 256 X 256 pixels quantized uniformly to 8 bits. Fig. 3 shows typical frames of these data sets. These were obtained from digitization of 24 frame-per-second motion pictures. The former were obtained with fixed camera position and the latter were photographed from a helicopter. The distortion function $g(x) = x^2$ was used so that the DMD would correspond to minimum mean square error in registration of the subblocks. A subblock size of 16 X 16 was chosen. The sizes 16 X 16 and 32 X 32 were found to be good compromises between accuracy of piecewise translational approximation of the motion, the cost of transmitting displacement vectors, and the complexity of data compression schemes using motion measurement.


Frame No. 8

Frame No. 12

Fig. 3. Frames of the original "Cronkite" (on the top) and "Chemical Plant" (on the bottom) data sets.

For the multiframe data, when the reference image is a neighboring frame of the image relative to which motion measurement is done, the variance of the interframe differences is called interframe variance (IFV). Once the DMD for a subblock has been found, the area of the reference image in the direction of the DMD is taken as the motion compensated estimate of the subblock. By collecting all the motion compensated estimates from the reference frame, one obtains the motion compensated reference frame. If (q, l) is the displacement vector of the DMD, then $D(q, l)$ is the IFV with motion compensation and $D(0, 0)$ is the IFV without motion compensation for that subblock. These quantities for a frame are obtained by averaging them over all the subblocks.

The effect of motion compensation can be indicated by the improvement in the signal-to-noise ratio (SNR) if the frame differences are encoded. We define

$$\text{SNR} = 10 \log_{10} \frac{(\text{peak-to-peak value of the signal})^2}{(\text{variance of error})} \text{ dB.} \tag{5}$$

The improvement in SNR also represents the reduction in IFV due to motion compensation. Table I summarizes these results. An interesting and important observation is that there

TABLE I
EFFECT OF MOTION COMPENSATION OF INTERFRAME CODING OF FRAME DIFFERENCES

DATA	FRAME NO.	SNR IN DECIBELS		
		WITHOUT MOTION COMPENSATION	WITH MOTION COMPENSATION	IMPROVEMENT
CRONKITE	6	29.90	35.88	5.98
	7	25.26	35.68	10.42
	8	26.18	36.30	10.12
	9	26.03	36.26	10.23
CHEMICAL PLANT	11	16.66	26.77	10.11
	12	16.90	26.69	9.79
	13	17.53	26.56	9.03

is a large frame-to-frame variation in the IFV without motion compensation due to variation in motion activity as a function of time (4.64 dB between frames 6 and 7 of "Cronkite" image set). After motion compensation this variation becomes negligible giving a nearly constant variance. This means that we could use a stationary prediction model for pixels along their motion trajectory.

Effect of Displacement on Temporal Correlation

The foregoing algorithm measures displacement up to an integer number of pixels per frame. Higher accuracy (up to a fraction of a pixel displacement) [1], [3]-[5] can be useful in further lowering the variance of the prediction error, resulting in higher levels of compression. In theory, it is possible to achieve higher accuracy by interpolating between the displacements of a pixel over several frames. This will generally increase the frame storage requirements as well as the complexity of the method.

A simpler and yet effective alternative is to use a prediction model along the estimated motion trajectory where the prediction rule is designed to account for the motion uncertainty.

Let $u(x, y, t)$, a zero-mean random variable, denote the intensity of the pixel at location (x, y). Let each image be a sample of a 2-D homogeneous stationary random field whose covariance is given by

$$E[u(x', y', t)u(x + x', y + y', t)] = \sigma^2 R(|x|, |y|) \tag{6}$$

where $E[\cdot]$ denotes the expectation and σ^2 is the variance of $u(x, y, t)$. If $(x + dx, y + dy)$ is the new location of the pixel at time $t + dt$, then the trajectory of motion is given by

$$u(x, y, t) = u(x + dx, y + dy, t + dt). \tag{7}$$

Let $\overline{dx}$ and $\overline{dy}$ be the estimates of dx and dy, respectively, and let

$$\tilde{dx} = dx - \overline{dx}, \quad \tilde{dy} = dy - \overline{dy} \tag{8}$$

be the motion estimation error. The motion compensated interframe estimate u^c is given by

$$u^c(x + \overline{dx}, y + \overline{dy}, t + dt) = u(x, y, t). \tag{9}$$

The temporal correlation after motion compensation is given by

$$\alpha = \frac{E[u^c(x + \overline{dx}, y + \overline{dy}, t + dt)u(x + \overline{dx}, y + \overline{dy}, t + dt)]}{E[u^2(x + \overline{dx}, y + \overline{dy}, dt)]}$$

$$= \frac{1}{\sigma^2} E[u(x, y, t)u(x + \overline{dx}, y + \overline{dy}, t + dt)]. \tag{10}$$

Assuming $\tilde{dx}$, $\tilde{dy}$ to be independent random variables and using (7) and (8) we obtain

$$\alpha = E[R(|\tilde{dx}|, |\tilde{dy}|)]. \tag{11}$$

For small x, y most image covariance functions can be approximated by a function linear in $|x|$, $|y|$, giving

$$\alpha = R(E|\tilde{dx}|, E|\tilde{dy}|). \tag{12}$$

Thus, from the distributions of $\tilde{dx}$ and $\tilde{dy}$ one can obtain the temporal correlation. The motion compensated interframe variance is defined as

$$\xi^2 = E[\{u(x + \overline{dx}, y + \overline{dy}, t + dt) - u^c(x + \overline{dx}, y + \overline{dy}, t + dt)\}^2] \tag{13}$$

which can be simplified via (6), (9), and (10) to yield

$$\xi^2 = 2\sigma^2(1 - \alpha) \tag{14}$$

or

$$\alpha = 1 - \frac{\xi^2}{2\sigma^2}. \tag{15}$$

This relation is useful in estimating the temporal correlation from the motion compensated IFV measurement. Now, if we use a motion compensated predictor defined as

$$\hat{u}(x + \overline{dx}, y + \overline{dy}, t + dt) = \alpha u(x, y, t) \tag{16}$$

then it is easy to show that for given displacement estimates $\overline{dx}, \overline{dy}$, the $\hat{u}$ given above is the optimum mean square estimate of $u(x + \overline{dx}, y + \overline{dy}, t + dt)$ and the associated variance of the motion compensated prediction error is given by

$$\beta^2 = E[\{u(x + \overline{dx}, y + \overline{dy}, t + dt) - \hat{u}(x + \overline{dx}, y + \overline{dy}, t + dt)\}^2] \tag{17}$$

$$= \sigma^2(1 - \alpha^2). \tag{18}$$

Since α is a correlation parameter, $|\alpha| \leqslant 1$. Comparing (14) and (18) we conclude that $\beta^2 \leqslant \xi^2$. This means that the motion compensated prediction rule (16) will further improve the performance of an interframe predictive coder, especially if α is not very close to 1. Since we found the motion compensated IFV to be nearly constant, a constant value of the temporal correlation can be used for all the pixels. In the sequel we will find the above results useful in analyzing the performance of interframe coders.

Equation (12) is also useful for measuring the average displacement uncertainties $E[|dx|]$ and $E[|dy|]$. For example, let

$$R(|x|, |y|) = \rho^{\sqrt{x^2+y^2}}. \tag{19}$$

For $\rho = 0.96$, $\alpha = 0.99$ and assuming $E[\tilde{dx}] = E[\tilde{dy}] \triangleq h$ we obtain from (12)

$$\alpha = \rho^h \tag{20}$$

or

$$h = \frac{\log \alpha}{\log \rho} = \frac{\log [1-(1-\alpha)]}{\log [1-(1-\rho)]} \cong \frac{1-\alpha}{1-\rho} \tag{21}$$

which gives $h = 0.25$. Thus, if we assume $\tilde{dx}$, $\tilde{dy}$ to be uniformly distributed, the maximum error in displacement measurement would be ±0.5 pixel.

IV. APPLICATION TO DATA COMPRESSION

In this section we will show the usefulness of the estimated motion in data compression for transmission or storage of images.

Frame Skipping

Frame skipping (temporal subsampling) is one of the simplest methods of data compression for interframe motion images. For simplicity of discussion, suppose only the alternate frames are skipped. (Our discussion can be easily extended to other cases.) With no knowledge of motion trajectory of the pixels, a skipped frame is generally reproduced either by repeating the preceding frame or by interpolation between the preceding and the following frames. Both of these methods have serious effects on the quality of motion reproduction. The former results in jerkiness in the reproduction of the motion [Figs. 4(a) and 5(a)] and the latter in blurring of the moving areas [Figs. 4(c) and 5(c)].

Let u_{2k} be a subblock of the $(2K)$th frame where frames $2, 4, \cdots, 2k, \cdots$ have been skipped. Then u_{2k}^*, the reproduced value of u_{2k}, is obtained (without motion compensation) as follows.

Frame Repetition

$$u_{2k}^*(m,n) = u_{2k-1}(m,n) \tag{22}$$

Frame Interpolation

$$u_{2k}^*(m,n) = \tfrac{1}{2}\{u_{2k-1}(m,n) + u_{2k+1}(m,n)\}. \tag{23}$$

The disadvantages of frame repetition and interpolation can be overcome by predicting or interpolating the pixels of the skipped frame along its motion trajectory. Thus, with motion compensation, (22) and (23) are replaced by

$$u_{2k}^*(m,n) = u_{2k-1}(m+q, n+l) \tag{24}$$

and

$$u_{2k}^*(m,n) = \tfrac{1}{2}\{u_{2k-1}(m+q, n+l) + u_{2k+1}(m+q', n+l')\} \tag{25}$$

respectively, where (q, l) and (q', l') are the coordinates of the displacement vectors of u_{2k} relative to the preceding and the following frames, respectively.

Figs. 4 and 5 show the results of data compression via frame skipping on typical frames of the two image sets. With motion compensation the SNR improves by about 10 dB, which is quite significant.

Motion Compensated Interframe Hybrid Coding

Once the motion trajectories of all the pixels are known, various image coding techniques [9], [10] can be applied by adapting them along these trajectories. For example, in predictive techniques (DPCM or hybrid coding) the predictor will use pixels that lie on the motion trajectory. For three-dimensional transform coding, one would select spatial blocks which lie on the motion trajectory.

We have applied the motion compensation in conjunction with the interframe *hybrid coding*, which has been discussed by Habibi, Roese *et al.* and Jain and Wang [14]-[17]. Here a method of analysis for motion compensated hybrid coding and some supporting experimental results are presented.

Fig. 6(a) and (b) shows the block diagrams for interframe hybrid coding without and with motion compensation, respectively. Each transform coefficient is coded in the temporal dimension by a first-order DPCM. Let U_k denote a 16 X 16 block; then V_k given by

$$V_k = \Psi U_k \psi^T \tag{26}$$

is its two-dimensional cosine transform, where Ψ denotes the 16 X 16 unitary cosine transform matrix.

The DPCM loops in Fig. 6 are based on the assumption that the transform coefficients $V_k(m,n)$ can be represented by a first-order Markov sequence with correlation α along the estimated motion trajectory. If $\sigma_v^2(m,n)$ denotes the variance of $V_k(m,n)$ and the differential signal $e_k(m,n)$ is quantized by its Max quantizer, then its variance is given by [10], [17]

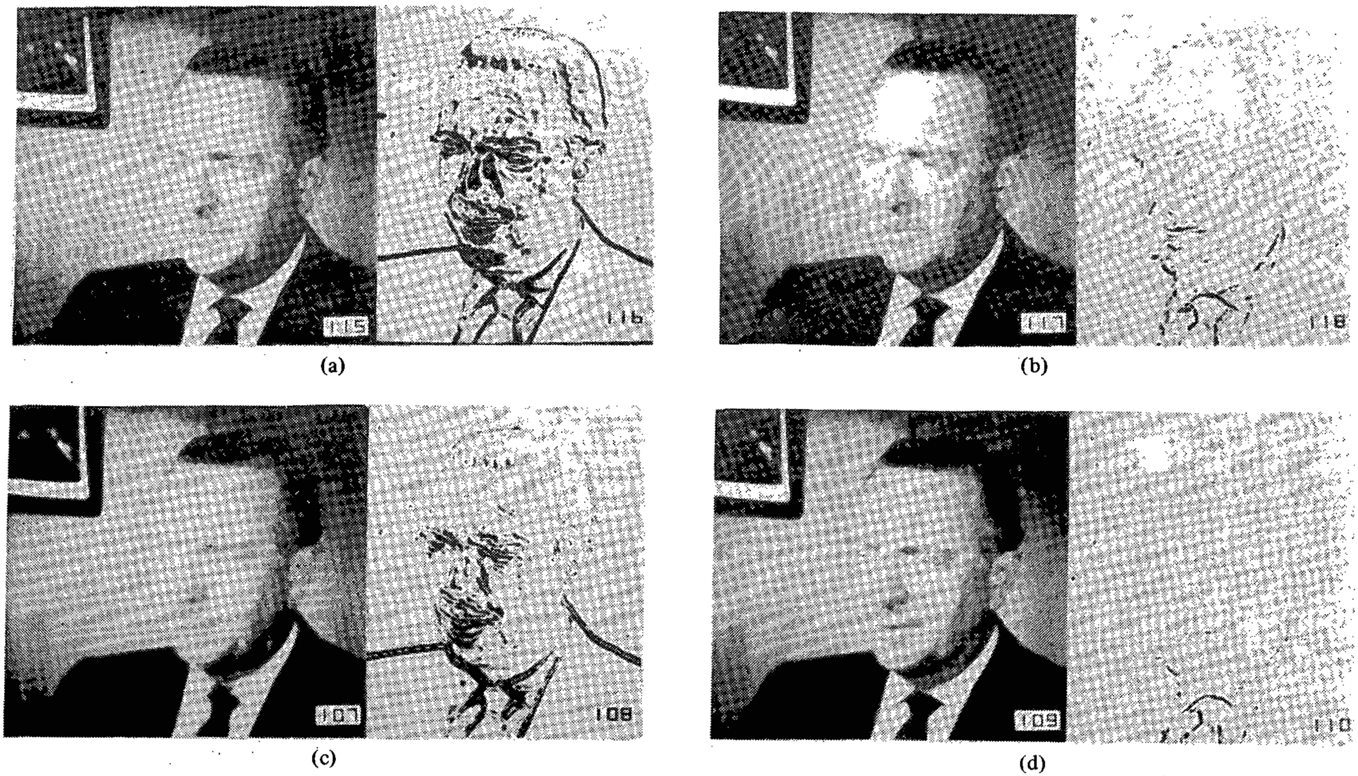
$$\sigma_e^2(m,n) = \sigma_v^2(m,n)(1-\alpha^2)/[1-\alpha^2 f(b_{m,n})] \tag{27}$$

where $f(x)$ is the mean square distortion function of an x-bit Max quantizer whose input has unit variance. Assuming noise-free transmission, the average mean square distortion for an $N \times N$ block is given by

$$D = \frac{1}{N^2} \sum\sum_{m,n=1}^{N} \sigma_e^2(m,n) f(b_{m,n}). \tag{28}$$

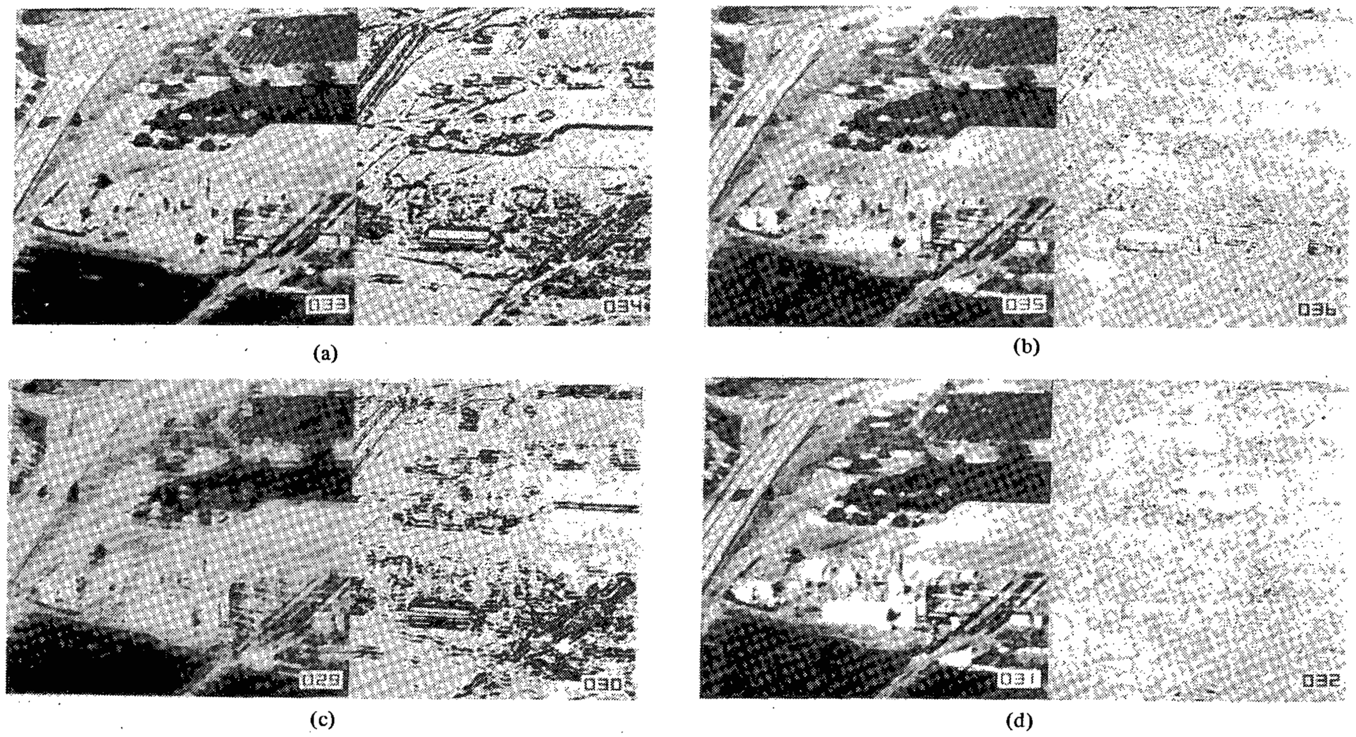
If the average bit rate for a block is b per pixel so that

$$\frac{1}{N^2} \sum\sum_{m,n} b_{m,n} = b \tag{29}$$



(a) (b) (c) (d)

Fig. 4. Data compression by frame skipping. Resulting images and absolute value of the error frame repetition and frame interpolation of "Cronkite" image frame 8. The error is amplified by a factor of 10 and truncated to 255. An inverse bit assignment is used (255 corresponds to black) for the error images. (a) Frame repetition along temporal axis, SNR = 26.17 dB. (b) Frame repetition along motion trajectory, SNR = 36.31 dB. (c) Frame interpolation along temporal axis, SNR = 30.47 dB. (d) Frame interpolation along motion trajectory, SNR = 38.62 dB.



(a) (b) (c) (d)

Fig. 5. Data compression by frame skipping. Resulting images and absolute value of the error of frame repetition and frame interpolation of "Chemical Plant" frame 12. The error is amplified by a factor of 5 and truncated to 255. An inverse bit assignment is used for the error images. (a) Frame repetition along temporal axis, SNR = 16.90 dB. (b) Frame repetition along motion trajectory, SNR = 26.69 dB. (c) Frame interpolation along temporal axis, SNR = 19.34 dB. (d) Frame interpolation along motion trajectory, SNR = 29.56 dB.

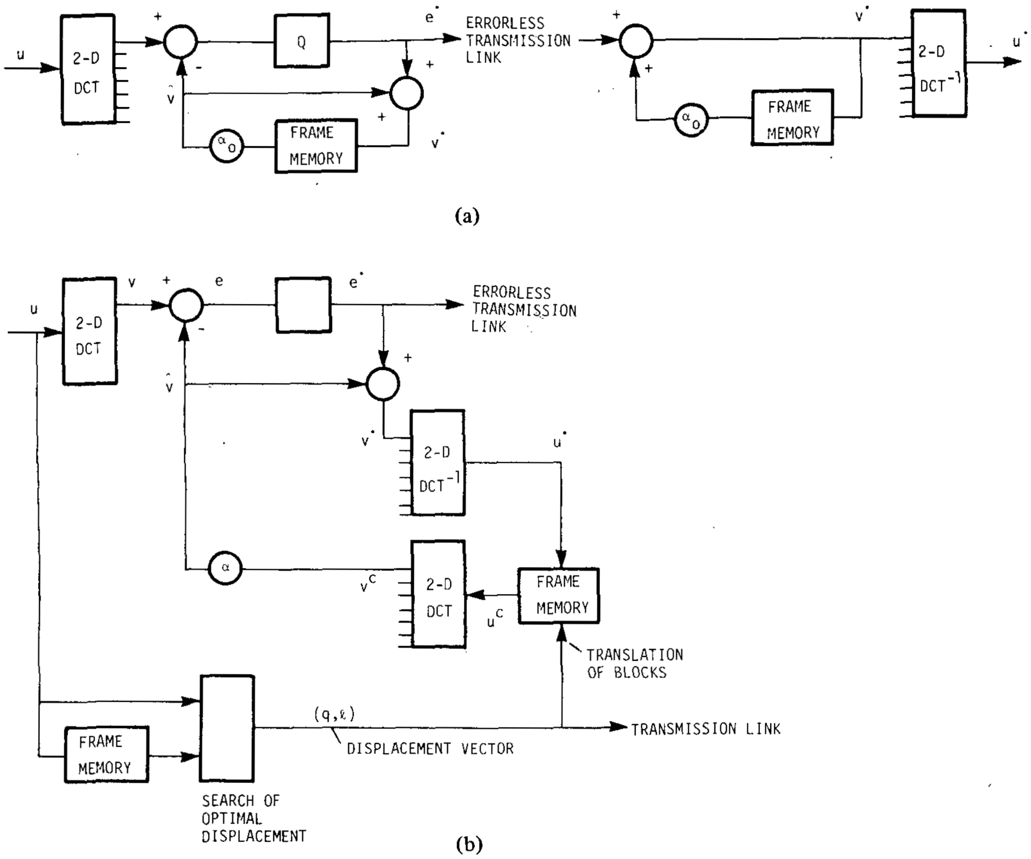


Fig. 6. Interframe hybrid coding. (a) Conventional. (b) With motion compensation.

then, given b, one can find integers $b_{m,n} \geqslant 0$ such that (28) is minimized. Now, the distortion rate function for motion compensated hybrid coding can be determined as follows.

Let $\tilde{d}x$, $\tilde{d}y$, which represent displacement uncertainty in pixels/frame, be identically distributed random variables. For the purpose of computing the distortion rate function we choose the following two distributions–Gaussian probability distribution $N(\mu, \sigma)$ which as mean μ and standard deviation σ, and a uniform distribution $B(p, q)$ defined in the interval (p, q). Given the probability distribution models for $\tilde{d}x$, $\tilde{d}y$ and a spatial covariance model, we can find the temporal correlation coefficient α via (12). In our study the isotropic covariance model of (19) was used (with $\rho = 0.95$ for "Cronkite" images and $\rho = 0.9$ for "Chemical Plant" images). Then for each fixed b, we minimize (28) with respect to $b_{m,n}$ such that $b_{m,n} \geqslant 0$ are integers and (29) is satisfied. This is done quite easily following the integer bit allocation algorithm given in [10], [17]. The pairs of values D, b then give the desired distortion rate function.

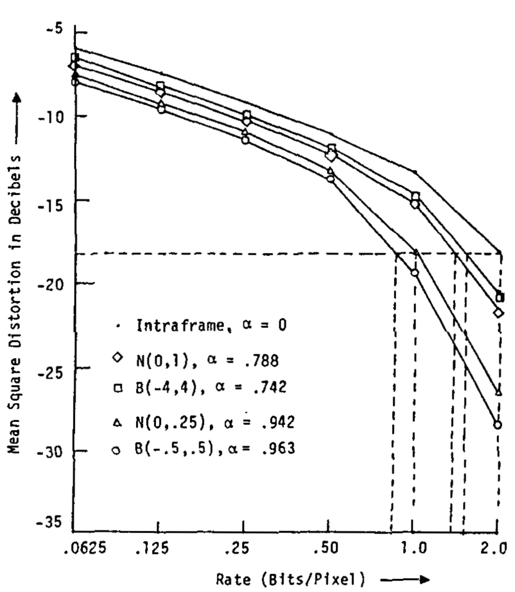


Fig. 7. Distortion-rate curves for hybrid interframe coding of images with isotropic intraframe covariance and for various distributions of interframe motion uncertainty.

Fig. 7 shows some distortion rate curves for various distributions of $\tilde{d}x$ and $\tilde{d}y$ for unit variance data with $\rho = 0.90$. The distributions corresponding to $B(-4, 4)$, $N(0, 1)$ can be considered as models for coding without motion compensation ($\tilde{d}x = dx$), and $B(-0.5, 0.5)$, $N(0, 0.25)$ for coding with motion compensation. The curve for $\alpha = 0$ corresponds to the intraframe transform coding. We notice that in the absence of motion compensation interframe hybrid coding does achieve some compression gain over the intraframe transform coding (about 2.5 dB at 2 bits/pixel). Motion compensation results in additional gain (about 7 dB). The actual gain achieved for a block will depend on the extent of displacement for that block and will vary from block to block since not all the blocks undergo the same displacement.

Experimental Results

An extensive simulation of several hybrid coding algorithms, viz. conventional, adaptive, and adaptive with motion compensation (with and without frame skipping and interpolation), were performed on the two data sets. While full details may be found in [7], here we summarize the key results and conclusions.

In the adaptive algorithm each block is classified into one of four classes based on the IFV (with or without motion compensation). In general it is a measure of combined spatial and temporal activity of the block under consideration. With motion compensation it is expected primarily to measure the spatial activity. For each class we define a spatial correlation model of the form

$$R(|x|, |y|) = \exp\left(-\sqrt{a_1{}^2x^2 + a_2{}^2y^2}\right) \quad (30)$$

where $e^{-a_1} \triangleq \rho_x$, $e^{-a_2} \triangleq \rho_y$ define the one pixel correlations along the x and y directions. The transform coefficient variances $\sigma_v{}^2(m, n)$ are determined from this model. The temporal correlation α is also determined from this model and the uniform probability density model $B(-0.5, 0.5)$ for the displacement uncertainties.

The algorithm could be operated in two modes, with and without skipping the alternate frames. The skipped frames are interpolated along the motion trajectories as described before. Table II gives typical parameters for different classes for the "Cronkite" data set. It was noted that the temporal correlation after motion compensation is reduced when frames are skipped. This means our ability to predict motion decreases as the temporal resolution decreases, a result which is expected. Table III shows that the overall SNR at an approximate bit rate of 0.25 bits/pixel/frame is the same with and without frame skipping. However, the former has a better spatial reproduction while the latter has better temporal reproduction. In Table IV we give the same results without motion compensation. Some of the coded images together with the error images are shown in Fig. 8. Comparison of the picture quality as well as the entries of Tables III and IV shows an improvement in bit rate due to motion compensation by a factor of two at low bit rates. The bit rates reported here include the overhead for transmitting the displacement vector and classification bits wherever applicable.

Remarks

1) The foregoing methods differ considerably from other methods including that of [8]. The displacement measurement algorithm and its utilization in predictive coding are basically different. Our experience shows that a displacement measurement based on low bit quantized values used for transmission results in significant degradation of the accuracy of the displacement vector. Quantization noise further reduces the temporal correlation, thereby lowering the achievable compression.

2) Although we used Fig. 6(b) for our motion compensated hybrid coding simulations, the complexity of our coder can be reduced by implementing it as shown in Fig. 9, where the DCT operations are performed on the prediction error. Compared to Fig. 6(b), we now have only one DCT and one inverse DCT operation at the transmitter. The realization of Fig. 9 is only applicable for a constant value of α for all the elements of V as used in our experiments. For a general case where α is variable one would need the added complexity of Fig. 6(b). We used the algorithm of Fig. 6(b) because it facilitated the use of a single computer program for simulating the several different hybrid coding algorithms that were compared. Conceptually, the algorithm of Fig. 6(b) is more illustrative in demonstrating the idea of motion compensation in interframe coding and also in developing the distortion rate analysis for this method.

TABLE II
PARAMETERS OF THE 4 CLASS ADAPTIVE HYBRID CODING SCHEME WITH MOTION COMPENSATION FOR "CRONKITE" IMAGES

CLASS NO.	ACTIVITY INDEX	PROBABILITY OF OCCURRENCE	CORRELATION PARAMETERS		
			ρ_x	ρ_y	α
1	0-10	.506	.985	.98	.996
2	10-20	.295	.955	.945	.99
3	20-50	.143	.91	.90	.97
4	50-	.057	.80	.78	.95

TABLE III
PERFORMANCE OF THE ADAPTIVE HYBRID CODING WITH MOTION COMPENSATION FOR "CRONKITE" IMAGE FRAMES 5–9; SKIPPED FRAMES ARE INTERPOLATED ALONG THE MOTION TRAJECTORY

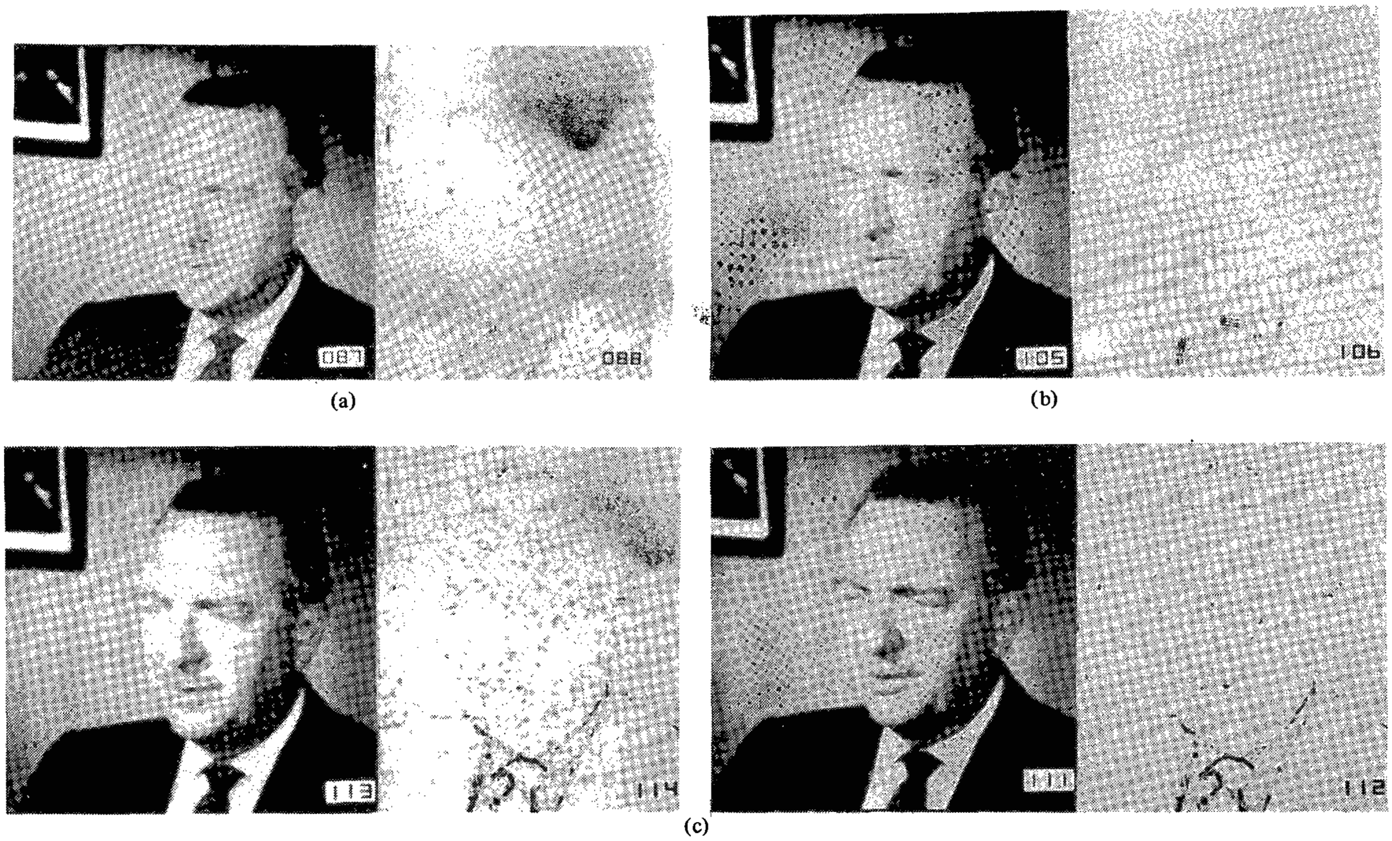
FRAMES SKIPPED?	AVERAGE BIT-RATE PER PIXEL	SIGNAL TO NOISE RATIO IN DECIBELS		
		CODED FRAMES	INTERPOLATED FRAMES	OVERALL
NO	.253	38.74	--	38.74
YES	.252	39.97	37.58	38.62
YES	.125	37.60	36.69	37.12

TABLE IV
PERFORMANCE OF ADAPTIVE HYBRID CODING WITHOUT MOTION COMPENSATION FOR "CRONKITE" IMAGES

S. N.	Overall Bit-rate	SNR in dB
1	.114	32.86
2	.25	36.52
3	.50	39.85

3) The adaptive hybrid coding algorithm described above is a variable bit rate scheme. The same algorithm used in a nonadaptive manner will become a fixed bit rate scheme, but its performance will not be as good. Our results indicate that without motion compensation the adaptive scheme has a bit rate roughly half that of the nonadaptive scheme [7]. The algorithm described here seems suitable for videotelephone, video conference, etc. For these applications the variable bit rate is not expected to result in very large buffer requirement.

4) Although we have primarily used motion compensation with hybrid coding, it can be used in conjunction with three-dimensional DPCM as well as with three-dimensional transform coding. With DPCM, the algorithm would differ



(a)

(b)

(c)

Fig. 8. Coded images and error images resulting from adaptive hybrid coding. Error images are obtained in the same manner as in Fig. 4. (a) No motion compensation. Bit-rate/pixel = 0.25, SNR = 36.78 dB. (b) Motion compensation without frame skipping. Bit-rate/pixel = 0.25, SNR = 38.51 dB. (c) Motion compensation with alternate frame skipping and interpolation of skipped frames along motion trajectory (interpolated images). (i) Bit-rate/pixel = 0.125, SNR = 36.70 dB. (ii) Bit-rate/pixel = 0.25, SNR = 37.59 dB.

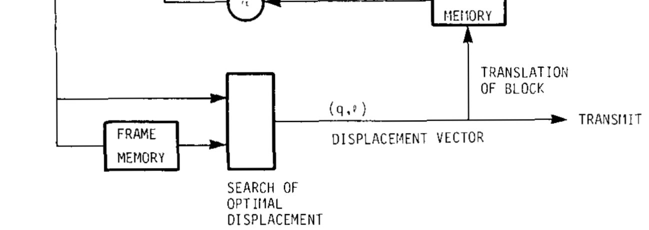


Fig. 9. Simplified interframe hybrid coding with motion compensation.

from that of Fig. 9 only in that the two-dimensional prediction error sequence $\epsilon_k(m, n)$ will be coded by a two-dimensional DPCM coder (rather than the transform coder used in Fig. 9). This will reduce the coder complexity as well as its performance much in the same way as two-dimensional DPCM does with respect to two-dimensional transform coding. Use of the displacement vector in three-dimensional transform coding is possible but would increase the complexity too much. Compared to three-dimensional adaptive transform coding (using a classification method similar to the one used here), the motion compensated adaptive hybrid coding has lower complexity and gives higher SNR.

V. SUMMARY AND CONCLUSIONS

A new displacement measurement technique for interframe coding has been presented. The technique is based on dividing a frame into smaller rectangular blocks and finding the direction of minimum distortion (DMD) for each block. An efficient search technique over a 2-D plane has been devised for finding the DMD. Use of the displacement vector was shown in analyzing and designing interframe motion compensated hybrid coders. Motion compensated hybrid coders can be useful in achieving low bit rates for transmission or storage of interframe images. Bit rate reduction by approximately a factor of two seems possible by using motion compensation.

APPENDIX

A PROOF OF THE DMD SEARCH ALGORITHM

A proof of the convergence of the DMD search algorithm described in Section II can be given as follows. Our initial estimate of DMD ($q = 0, l = 0$) is successively incremented by (i, j) in Step 3. For any given search-step size n, there are three cases for the value of (i, j) found in Step 3: 1) $i \neq 0, j = 0$, 2) $i = 0, j \neq 0$; and 3) $i = 0, j = 0$.

For cases 1) and 2) the straight lines $I = q$ and $J = l$, respectively, divide the current region of search into two regions, one in which the DMD lies and the other in which it does not lie. This is guaranteed by the condition of (3). The region in which the DMD does not lie is excluded from further search by $M'(n) \leftarrow M' - (-i, -j)$ in Step 4. After having gone through Steps 3 and 4 repeatedly, the procedure will come to case 3) because we are searching over a finite region and we are successively reducing the region of search under cases 1) and 2).

Under case 3) the DMD is guaranteed to lie between the region bounded by the lines $I = q - n, I = q + n, J = l - n$, and $J = l + n$ due to the condition of (3). Therefore, the search-step size n is reduced to $n/2$ in Step 5, and the procedure is repeated till $n = 1$. For $n = 1$ the DMD is guaranteed to lie within $(q \pm 1, l \pm 1)$ and Step 6 thus concludes the search by searching over all the nine points in this region.

REFERENCES

[1] J. O. Limb and J. A. Murphy, "Measuring the speed of moving objects from television signals," *IEEE Trans. Commun.*, vol. COM-23, pp. 474–478, Apr. 1975.

[2] F. Rocca and S. Zanoletti, "Bandwidth reduction via movement compensation on a model of the random video process," *IEEE Trans. Commun.*, vol. COM-20, pp. 960–965, Oct. 1972.

[3] C. Cafforio and F. Rocca, "Method for measuring small displacements of television images," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 573–579, Sept. 1976.

[4] S. Brofferio and F. Rocca, "Interframe redundancy reduction of video signals generated by translating objects," *IEEE Trans. Commun.*, vol. COM-25, pp. 448–455, Apr. 1977.

[5] A. N. Netravali and J. D. Robbins, "Motion-compensated television coding: Part I," *Bell Syst. Tech. J.*, vol. 58, pp. 631–670, Mar. 1979.

[6] J. R. Jain and A. K. Jain, "Adaptive techniques for motion compensation in interframe coding," presented at the Picture Coding Symp., Ipswich, England, 1979.

[7] ——, "Interframe adaptive data compression techniques for images," Signal and Image Processing Lab., Dep. Elec. Comput. Eng., Univ. of California at Davis, Tech. Rep., Aug. 1979.

[8] A. N. Netravali and J. A. Stuller, "Motion compensated transform coding," *Bell Syst. Tech. J.*, vol. 58, pp. 1703–1718, Sept. 1979.

[9] A. N. Netravali and J. O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. 68, pp. 366–406, Mar. 1980.

[10] A. K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, pp. 349–389, Mar. 1981.

[11] W. K. Pratt, "Correlation techniques of image registration," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-10, pp. 353–358, May 1974.

[12] A. K. Jain and J. R. Jain, "Radar image modeling and processing for real-time RF simulation," Dep. Elec. Eng., State Univ. of New York at Buffalo, Final Rep., ARO Grant DAAG2977G004, Feb. 1978.

[13] D. E. Knuth, *Searching and Sorting, Vol. 3, The Art of Computer Programming*. Reading, MA: Addison-Wesley, 1973.

[14] A. Habibi, "Hybrid coding of pictorial data," *IEEE Trans. Commun.*, vol. COM-22, pp. 614–624, May 1974.

[15] J. A. Roese, "Interframe coding of digital images using transform and transform/predictive techniques," Image Processing Inst., Univ. of Southern California, Los Angeles, USCIPI Rep. 700, June 1976.

[16] J. A. Roese *et al.*, "Interframe cosine transform image coding," *IEEE Trans. Commun.*, vol. COM-25, pp. 1329–1338, Nov. 1977.

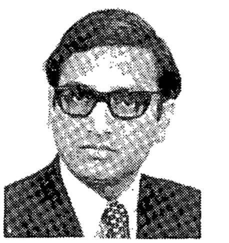
[17] A. K. Jain and S. H. Wang, "Stochastic image models and hybrid coding," State Univ. of New York at Buffalo, Final Rep., Contr. N00953-77-C-003MJE, Oct. 1977.

★

Jaswant R. Jain (S'73–M'74–S'79–M'79) was born in Rajasthan, India in 1949. He received the B.E. degree from the University of Jodhpur, Jodhpur, India, the M.E. degree from the Indian Institute of Science, Bangalore, India, and the Ph.D. degree from the State University of New York at Buffalo, all in electrical engineering, in 1972, 1974, and 1979, respectively.

During 1974–1975 he was a Graduate Student and Senior Research Assistant at the Advanced Center for Electronic Systems, Indian Institute of Technology, Kanpur, India. He was a Graduate Fellow at S.U.N.Y. Buffalo, where he worked on modeling, restoration, and data compression of images. Since August 1979, except for a brief period, he has been with Systems Control, Inc., Palo Alto, CA, where he is presently a Research Engineer. He has worked on optical sensing of ocean surfaces and is currently active in low bit rate speech compression. Between November 1980 and February 1981 he worked at National Semiconductor Corporation, Santa Clara, CA, in the area of speech recognition. His research interests are in the areas of digital signal processing, image processing, and speech compression and recognition.

★



Anil K. Jain was born in India on January 21, 1946. He received the B.Tech. (Hons.) degree from the Indian Institute of Technology, Kharagpur, India, in 1967 and the M.S. and Ph.D. degrees from the University of Rochester, Rochester, NY, in 1969 and 1970, respectively, all in electrical engineering.

From 1968 to 1970 he was a System Science Fellow at the University of Rochester, and from 1970 to 1971 he was a Post-Doctoral Fellow with the Department of Electrical Engineering and Computer Science, University of Southern California, Los Angeles. From 1971 to 1974 he was an Assistant Professor with the Department of Electrical Engineering and the Image Processing Institute at U.S.C. From 1974 to 1979 he was an Associate Professor at the State University of New York at Buffalo. During 1979–1980 he was an Acting Associate Professor at the University of California, Davis. Since 1979 he has been Professor of Electrical and Computer Engineering at U.C. Davis. He has been a Consultant for several private and federal organizations, and his research interests include digital signal and image processing, real-time computer systems, and pattern recognition.