/

MOTION-COMPENSATED TRANSFORM CODING BASED ON BLOCK MOTION-TRACKING ALGORITHM

A. Puri*, H.-M. Hang*, and D. L. Schilling**

* AT&T Bell Laboratories
Holmdel, NJ 07733 USA

** The City College of New York
New York, NY 10031 USA

ABSTRACT

This paper presents studies on the motion-compensated transform coding schemes using the block-matching technique. First, we present an efficient block motion-tracking algorithm that provides good motion estimation while minimizing the computations. Next, we investigate the effectiveness of various orthogonal transforms on the motion-compensated frame differential signal. Finally, the results of coding a real image sequence using various motion-compensated transform coding algorithms are discussed.

INTRODUCTION

Many recent efforts towards efficient image sequence compression employ techniques to detect, estimate and compensate for motion of objects in a scene. The so-called *motion-compensated coding schemes* estimate the displacements of moving objects and only encode the pel differences between the current frame and the translated previous frame in the moving areas of the picture. Researchers have shown that this technique can increase the coding efficiency significantly [1]-[6].

One popular technique to estimate the motion displacement is the block matching approach [1]-[5],[8]. In such a technique each picture frame is first partitioned into fixed-size blocks and the estimation of displacement vectors is then performed on each block by a matching technique.

Our proposed block-matching motion-compensated coding scheme has three elements: (1) a *motion detector* which detects the moving blocks, (2) a *displacement estimator* which estimates the displacement vectors of moving blocks, and (3) a *data compression algorithm* which encodes the inter-frame differences after motion-compensation. In this paper we focus on the performance comparison of various orthogonal transforms for encoding motion-compensated frame differences. In order to reduce the computational complexity for motion estimation, we adopt a motion tracking algorithm in which an initial guess of the displacement is made from the previous frame displacements, and a small update is used to correct this guess.

We first measure statistics of an image sequence for motion tracking algorithm assuming perfect reconstruction. This ideal-case simulation is helpful in measuring the performance of the basic motion estimation algorithm. Three orthogonal transforms (DCT, DST and 1-D KLT) are then chosen as possible candidates for coding the motion-compensated frame-differential signal. A statistical comparison of transform domain behavior of motion-compensated frame differential signals for chosen transforms is then performed.

Next, a (theoretically) asymptotically-optimum quantizer is added into the coder loop to judge the real coding performance on the test pictures. Among the three transforms (DCT, DST and 1-D KLT) in our study, DST and KLT seem to have a small performance margin in the coded picture quality and compressed data bit rate.

MOTION DETECTION AND ESTIMATION

A motion detector is first used to classify a block to be *moving* or *non-moving* before any motion estimation can be done for that block. It screens the original image blocks for motion so that the displacement estimator, which usually requires much more computations than the motion detector, works only on a smaller number of blocks which are actually moving. In addition, it also helps in reducing the number of blocks that may be erroneously classified as moving by the displacement estimator, because of uncertainty, when blocks are in low detail region and/or have random camera noise. The motion detector is thus necessary to ensure a zero displacement for the unchanged region. The design of our motion detector is motivated by earlier works [7]. Two parameters, T_0 and N_0, are used. A pel at location (x,y) in frame k is called *moving* if $|s_k(x,y) - s_{k-1}(x,y)| > T_0$, where $s_k(x,y)$ and $s_{k-1}(x,y)$ represent pels in frames k and k-1 respectively. A block is called *moving* if the number of moving pels in that block is greater than or equal to N_0. In our experiment, the parameters, $T_0=3$ and $N_0=10$, work adequately for blocks of size 8×8 and 8-bit pels in the range 0-255.

The goal of a block-displacement estimator is to find the best match of a block from frame k in a suitable *match area* in the previous frame k-1. The (absolute) detectable displacement is assumed to be less than X_m along horizontal (x) axis and Y_m along vertical (y) axis. If the size of a reference block is $M\times N$, the match area is $(M+2X_m)\times(N+2Y_m)$ where X_m and Y_m assume discrete values for digitized images.

The above problem consists of seeking for a best match location by evaluating a matching criterion at every point in the displacement *search region* (SR) having $(2X_m+1)\times(2Y_m+1)$ points. In most practical cases, M is often chosen to be the same as N and $X_m=Y_m=p$ and hence SR has $(2p+1)^2$ points. Each point in SR is a candidate for being the correct displacement vector and will be called a *search point*.

A matching criterion called *number of thresholded absolute differences (NTAD)* is employed and is defined as [6]:

$$NTAD(u,v) = \sum_{y=1}^{M}\sum_{x=1}^{N}\left[f(T_0, |s_k(x,y) - s_{k-1}(x+u,y+v)|)\right]$$

$$\text{where } f(T_0,a) = \begin{cases} 1 & T_0 < a \\ 0 & T_0 \geq a \end{cases} \quad (1)$$

This criterion seems to perform slightly better perceptionally than the often used *mean absolute (frame) difference (MAD)*[1],[8].

If no prior information about the movement of object is known, every search point in SR is an equally likely candidate for the best-match. However, due to high temporal correlation in the movement between frames [4], [5],[8] the previous-frame displacement vectors in the neighborhood of the current block form a good initial estimate of the current displacement vector. In other words, a best-match can be obtained by a small search centered at the initial displacement estimate and this search need only be conducted on a small search region.

The motion tracking technique employed is similar to that of [4],[8]. Let $D_k(x,y)$ represent the displacement vector of the current block and $D_{k-1}(x,y)$ be the displacement estimate of the block at the same location in the previous-frame. Then,

$$D_k(x,y) = D_{k-1}(x,y) + U_k(x,y), \quad (2)$$

where $U_k(x,y)$ represents a small *update* or a correction term. The center of current SR is therefore shifted by the displacement vector of the previous-frame block $(D_{k-1}(x,y))$ at the same location before an update can be made. A small update on the initial estimate is usually sufficient to obtain the best estimate of motion for the current block. The update term is found by comparing the matching criteria (eqn. (1)) at all the positions in the small search region.

In order to evaluate the effectiveness of the motion estimation algorithms, we conduct a series of experiments on a real video sequence. The original Ms USA sequence is digitized from NTSC signals sampled at twice the color subcarrier frequency and 8 bits per pel. It is then converted to component form consisting of a luminance signal Y and chrominance signals U and V. A line of luminance contains 368 pels, and there are 240 such lines per field. Since our application is low-bit rate video-conferencing, we further reduce the spatial resolution by subsampling in the horizontal direction by a factor of 2. The image field now consists of 184 *pels* × 240 *lines*. This is converted into block format with block size of 8×8. Each field now contains 23 Y blocks in the horizontal direction and 30 Y blocks in the vertical direction, and will be referred to as a "frame" in the terms such as *frame difference* used in this paper. Performance of motion compensation algorithms is investigated at a temporal subsampling rate of 4:1, i.e., 15 fields (also referred to here as frames) per second. The displacements in the high-motion areas in the sequence still do not exceed 5 to 6 pels. This is typical of most movements in a video-conferencing environment.

From Fig. 1 we note that the entropy of the motion compensated frame difference signal is smaller than the uncompensated signal by as much as 35% (depending on the motion activity). Further, comparing the performance of the motion-tracking algorithm for $p=3$ to that of the simple search (exhaustive with zero initial estimate) for $p=6$, we also note that the two algorithms perform identically. Since the motion-tracking algorithm is computationally less complex while it maintains the performance of exhaustive search, it is employed in our inter-frame coding system.

ORTHOGONAL TRANSFORMS ON THE MOTION-COMPENSATED FRAME DIFFERENCES

The frame differences after motion compensation (motion-compensated frame difference, MCFD) have to be coded and sent to the receiver if their values are significant. Hence, we separate the MCFD signals into two categories: *compensable* and *uncompensable*. The ones that can pass the motion detector -- small-magnitude MCFD blocks -- are *compensable*. Therefore, all the image blocks are classified into one of the following three types as shown in Fig. 2.

Type 1: *nonmoving blocks* -- Their values are small and hence identified by the motion detector as nonmoving. They usually belong to the unchanged area like the background in images. In a real coder a special short code will be necessary to indicate the nonmoving blocks to the receiver.

Type 2: *compensable moving blocks* -- These blocks are originally identified as moving by the motion detector, but after motion compensation their values are found below the motion activity threshold. i.e., they belong to the changed areas where motion compensation works well. For such blocks only the motion vectors need to be transmitted to the receiver.

Type 3: *uncompensable moving blocks* -- These blocks are labeled as moving by the motion detector but cannot be well-compensated by a simple block translation because of the newly exposed region (uncovered background) or the rotational movement of regions. Both motion vectors and coded MCFD values are required to be transmitted to the receiver for reconstruction of these blocks.

Based on test data, assuming perfect reconstruction, approximately half of the overall MCFD blocks are found to be of Type 1; i.e., are not moving. Among the moving blocks, around half of them are well-compensable (Type 2) and rest of them are classified as uncompensable. This means that the transform coding algorithm for data compression need only be applied to about a quarter of the original blocks in each frame. In a real coding system the quantizer characteristics might change the number of blocks for each category as some coding errors might not be negligible and may result in inaccurate representation of the original picture. The difference between original and reconstructed picture may not be significant if a fine quantizer is employed but may be substantial if a coarse quantizer is employed.

Many coding algorithms may be used to encode the MCFD signals. The popular ones are predictive coding and transform coding [1]-[5]. The discrete cosine transform (DCT) has been recognized as an efficient coding technique for single picture (intraframe) coding; however, it is not clear that DCT would also be an efficient encoding technique for MCFD because the statistical properties of MCFD are very different from those of the original signals [8]. For instance, the pel-pel spatial correlation coefficient of MCFD is typically in the range 0.30-0.55, which is much lower than the pel-pel correlations typical (0.9) of original images. For highly-correlated signals, DCT has a good energy compaction and decorrelation property, if the signal source can be modeled as a first-order Markov model [9]. However, in the cases of low-correlation signal sources, other transforms may have a better energy compaction and decorrelation property [9]. Of course, this does not imply DCT may be a bad candidate in encoding MCFD because the first-order Markov model may not be adequate for the MCFD signals in the first place. Rather, the above observation motivates us to compare different transforms on the MCFD's.

Three orthogonal transforms for coding MCFD signals are studied: DCT, discrete sine transform (DST) and Karhunen-Loeve transform (KLT). The DCT and the DST are well defined ([9] and elsewhere) and will not be stated here. The true 2-D KLT derived based on the autocorrelation matrix of test data is complex and picture-dependent. In this study, in order to obtain a simple and practical KLT, we assume that the MCFD signal source is a separable first-order Markov model. Hence, the 2-D KLT in this case is an outer product of two 1-D KLT's and the 1-D KLT basis vectors have a closed-form representation [9] and is controlled by only one single parameter -- the pel-pel correlation coefficient. To get the KLT basis vectors, we first solve ω_p (between 0 and π) in the following equation:

$$\tan N\omega_p = \frac{-(1-\rho^2)\sin\omega_p}{(1+\rho^2)\cos\omega_p - 2\rho}, \qquad (3)$$

where N is the size of the transform and ρ the pel-pel correlation. The KLT basis vectors $\{t_{pq}\}$ are then obtained from

$$t_{pq} = \left(\frac{2}{N+\lambda_p}\right)^{1/2} \sin\left(\omega_p\left(q - \frac{(N-1)}{2}\right) + \frac{(p+1)\pi}{2}\right),$$

$$0 \le p,q \le N-1, \qquad (4)$$

in which the eigenvalue λ_p is calculated by

$$\lambda_p = \frac{1-\rho^2}{1+\rho^2 - 2\rho\cos\omega_p}. \qquad (5)$$

We use ρ=.5 as the parameter to generate our KLT since it roughly approximates the overall measured correlation coefficient of the MCFD in our previous study [8].

We first calculate the mean and variance calculations of transform coefficients (not included because of space restraints) for uncompensable moving blocks using these three transforms. their energy compaction ability in the transform domain is about the same. Typically about 30% or more transform coefficient are significant, and the DC component variance is of the same magnitude as the next few higher frequency coefficients.

We further look at few sample blocks of MCFD and their transform coefficients (Tables 1 and 2). The contents of MCFD blocks contain irregular patterns of positive and negative values, some significant and others insignificant. Looking at the coefficients in transform-domain for each of the transforms we again cannot find a regular structure of the dominant coefficients. The large-magnitude coefficients seem to be widely spread in all cases and retaining only a few low-frequency components may not be the best strategy. One possible solution is transmitting the locations of the dominant coefficients for each block; however, it requires large overhead. So far, it is not very clear which transform is more suitable for encoding the MCFD's.

SOME TRANSFORM CODING RESULTS

We have discussed earlier various parts of our block motion-compensated transform coding system, except for the quantizer. Since our statistics of transform coefficients do not seem to provide additional information for quantizer design and bit allocation, we therefore use a simple quantizer: a uniform midrise quantizer with small quantization steps. In theory, a fine uniform quantizer together with an entropy coding on the quantization indices would perform close to the (nonadaptive) optimum entropy-constrained quantizer designed for known probability distributions. A modified threshold sampling scheme is used to select the dominant coefficients: only the coefficients larger than the threshold (= twice of the quantization step) are quantized. In addition, we discard the three highest frequency components that are significant (greater than the threshold) and retain the 4 lowest frequency components even if they are small. Finally, we measure the entropy of the quantized coefficients. Overall, this quantizer should be universal and close to the theoretical optimum quantizer limit.

To allow a simple comparison of various transform coding algorithms, a fixed quantization step size of 6 is used for all the coefficients. The results (entropy bits and mean square error at each field) of DCT, DST and KLT coding schemes are displayed in Figs. 3 and 4. The DST and KLT coded sequences have a slightly larger mean square error as compared to DCT coded sequence but they require less entropy bits per picture. Also included are various performance measures for DST coding using fixed quantization step sizes of (3, 6 and 9) in Figs. 5(a), 5(b) and 5(c). The motion activity for various quantization step sizes are displayed in Fig. 5(a). With a coarser quantizer the number of uncompensable blocks increases whereas the number of compensable blocks decreases slightly as expected. Also, increasing the quantization step-size increases mean square reconstruction error (Fig. 5b) and reduces the coefficient bit rate generated (Fig. 5c) as further anticipated. The results for DCT and KLT(0.5) are very similar and thus are not included here. Statistically speaking, the overall performance of these three transforms is very close and there is no clear winner.

Finally, visual comparison is performed on the reconstructed picture sequences to compare the picture quality of DCT, KLT(0.5) and DST coding algorithms. The subjective differences in picture quality are found to be very small. On careful observation DST coded sequence seems to be perceptionally superior to the DCT or the KLT(0.5) coded sequence by a small margin. Thus DST coding scheme is slightly preferred in terms of entropy bits required to code the picture as well as the reconstructed picture quality even though it has slightly higher overall mean square error.

CONCLUSION

A block motion-tracking algorithm has been introduced for motion compensation and a transform coding scheme has been proposed in which only the uncompensable moving blocks are coded. Three orthogonal transforms are tested as possible candidates for this transform coding scheme for low-bit rate video applications. Their performance is found to be very similar in terms of mean square error and entropy bits. In a comparison test of visual quality on the reconstructed picture sequences, the DST coded sequence was found to be perceptionally preferable to the DCT or the KLT(0.5) coded sequences by a small margin.

REFERENCES

[1] H. G. Musmann, P. Pirsch and H.-J. Grallert, "Advances in picture coding," *Proc. IEEE*, Vol.73, pp.523-548, Apr. 1985.
[2] J. R. Jain and A. K. Jain, "Displacement measure-

ment and its application in interframe image coding," *IEEE Trans. Commun.*, Vol.COM-29, pp.1799-1808, Dec. 1981.

[3] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," in *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, Nov. 29-Dec. 3, 1981, pp.G5.3.1-G5.3.5.

[4] Y. Ninomiya and Y. Ohtsuka, "A motion-compensated interframe coding scheme for television pictures," *IEEE Trans. Commun.*, Vol.COM-30, pp.201-211, Jan. 1982.

[5] S. Ericsson, "Motion-compensated hybrid coding at 50 kb/s," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Mar. 1985, pp.10.8.1-10.8.4.

[6] C. D. Bowling and R. A. Jones, "Motion compensated image coding with a combined maximum a posteriori and regression algorithm," *IEEE Trans. Commun.*, Vol.COM-33, pp.844-857, Aug. 1985.

[7] B. G. Haskell, "Frame replenishment coding of television," in *Image Transmission Techniques*, W. K. Pratt, Ed. New York: Academic Press, 1979.

[8] A. Puri, H.-M. Hang, and D. L. Schilling, "An efficient block-matching algorithm for motion-compensated coding," to be published in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, April 1987.

[9] R. J. Clarke, *Transform Coding of Images*, London, *England:* Academic Press, 1985.

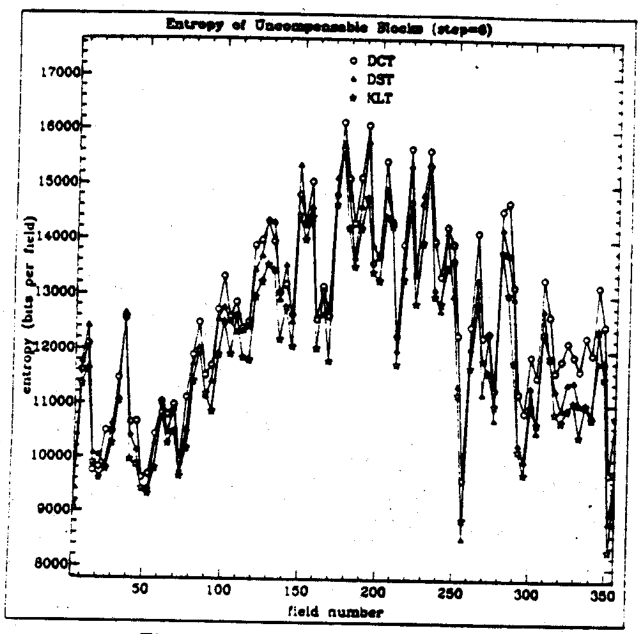

Figure 4 Bit Rates of Various Transform Coding Schemes

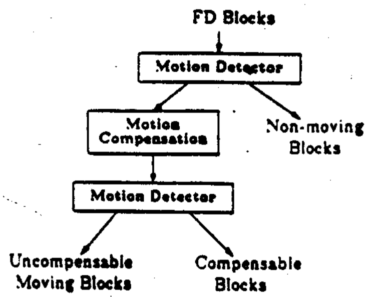

Figure 2 Frame Difference Block Types

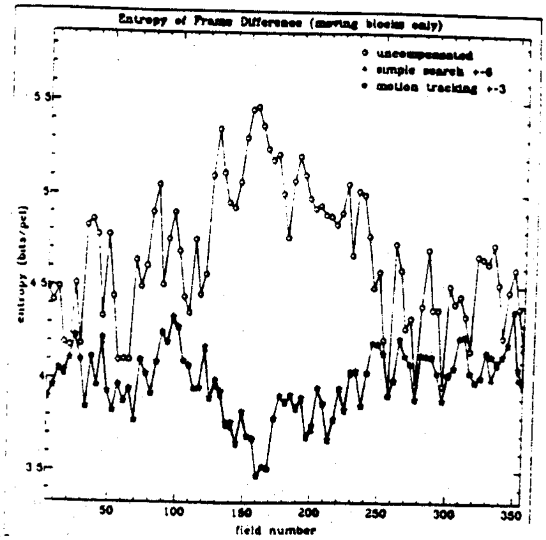

Figure 1 Performance Comparison Based on Entropy

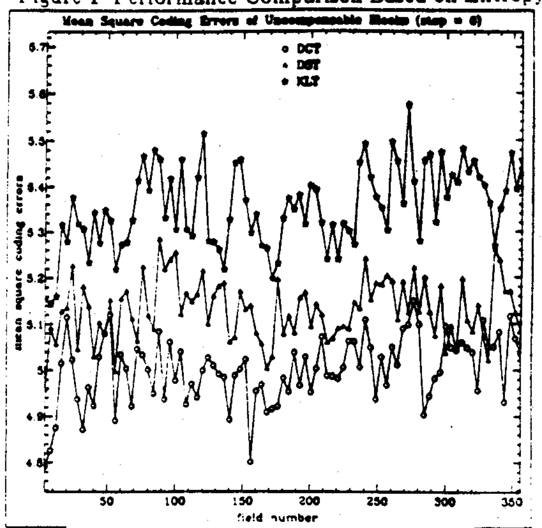

Figure 3 Mean Square Errors of Various Transform Coding Schemes

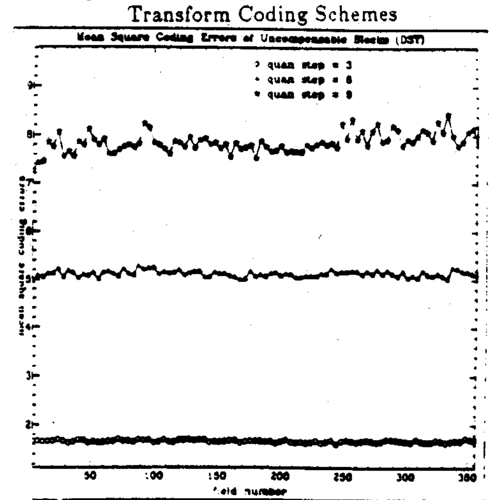

Figure 5(b) Transform Coefficient Step-size Effect on the Mean Square Coding Error (DST)

Table 1 A Typical Small-magnitude Uncompensable MCFD Block

Luminance Value								
Pel	1	2	3	4	5	6	7	8
1	-3	1	1	1	-3	0	2	4
2	-3	-4	-2	-3	-4	-2	0	0
3	1	-2	1	1	-6	-2	-2	1
4	-1	0	2	-1	-5	-2	-1	0
5	1	1	6	-7	-3	0	0	2
6	-2	3	5	-7	-2	0	1	1
7	0	1	4	-2	-1	-4	0	4
8	-1	3	1	1	0	0	1	0

DCT Coefficients								
Freq	1	2	3	4	5	6	7	8
1	-3.13	-0.18	7.08	-8.55	-6.13	-0.32	3.70	2.02
2	-4.25	-3.04	-0.14	-0.08	2.04	3.81	0.52	-3.87
3	2.14	-3.18	-2.83	-0.08	2.65	0.15	-4.44	-2.43
4	2.40	-1.95	2.26	-0.89	-4.36	-3.94	1.18	3.97
5	3.13	0.30	-1.01	0.26	-1.38	-0.39	-0.80	-1.50
6	1.85	0.91	0.06	-3.54	2.25	0.87	-0.88	-1.51
7	2.23	0.98	-0.69	0.31	-1.77	-0.21	-0.17	-2.08
8	2.25	1.04	1.73	-1.25	1.90	0.54	2.13	0.56

DST Coefficients								
Freq	1	2	3	4	5	6	7	8
1	-5.81	3.39	7.43	-7.26	-5.66	-2.08	4.47	2.40
2	-4.59	-2.60	-3.18	-1.16	2.05	4.88	0.18	-4.50
3	0.04	-2.04	-0.33	-3.35	1.50	-0.67	-2.54	-1.31
4	-0.03	-2.66	2.97	-0.36	-3.48	-3.30	0.99	2.46
5	2.79	-0.25	0.72	-1.01	-0.69	-0.49	-1.33	-1.48
6	1.19	1.11	0.33	-3.31	1.77	0.00	-0.42	-1.41
7	2.84	0.78	0.79	0.20	-1.65	0.08	-0.66	-2.35
8	1.46	1.29	1.65	-1.34	2.24	0.33	2.88	0.47

KLT Coefficients								
Freq	1	2	3	4	5	6	7	8
1	-4.98	1.98	7.31	-8.11	-5.99	-1.10	4.40	2.41
2	-4.61	-2.83	-1.89	-0.58	2.24	4.40	0.33	-4.38
3	1.13	-2.77	-1.01	-2.35	1.69	-0.15	-3.35	-1.75
4	1.04	-2.30	2.89	-0.71	-3.83	-3.57	1.17	3.17
5	3.17	-0.07	0.04	-0.32	-1.15	-0.36	-1.11	-1.53
6	1.56	1.14	0.30	-3.44	2.09	0.51	-0.65	-1.45
7	2.58	0.87	0.10	0.33	-1.77	-0.10	-0.38	-2.19
8	1.75	1.18	1.85	-1.24	2.13	0.47	2.36	0.53

Table 2 A Typical Large-magnitude Uncompensable MCFD Block

Luminance Value								
Pel	1	2	3	4	5	6	7	8
1	11	11	6	4	1	-5	-7	-7
2	-12	12	9	3	-6	-12	10	24
3	-61	5	-3	-6	1	-7	-1	15
4	-20	4	-1	3	9	-12	2	13
5	29	20	17	18	-3	-17	1	8
6	-8	16	23	13	-18	-9	3	2
7	-21	1	13	-14	-17	0	5	1
8	-8	5	-2	-13	-8	3	5	-1

DCT Coefficients								
Freq	1	2	3	4	5	6	7	8
1	3.38	-3.34	5.69	-49.04	-20.38	-12.57	-14.67	-1.21
2	4.20	-5.55	-6.73	-6.57	15.87	-11.26	-9.64	-0.02
3	-10.32	-0.51	5.21	19.01	-12.46	1.08	9.26	0.39
4	-23.84	45.29	9.83	2.34	6.58	23.58	9.20	-1.77
5	13.13	25.37	5.91	25.81	21.88	6.19	-0.42	3.97
6	-20.91	-14.56	-15.56	4.41	-8.81	-9.05	2.92	2.89
7	-10.20	-1.98	-14.74	-1.87	-6.88	-6.53	-3.46	-7.21
8	-3.36	4.31	-0.40	3.15	-1.19	-2.74	2.41	0.26

DST Coefficients								
Freq	1	2	3	4	5	6	7	8
1	5.72	14.44	10.84	-48.24	-11.07	-22.95	-18.07	-5.30
2	-0.06	-14.30	-11.41	-8.50	12.39	-21.18	-11.22	-2.26
3	-11.40	-6.44	8.17	-5.51	-22.83	-5.83	1.07	-1.57
4	22.43	39.65	13.81	7.57	18.03	26.01	9.26	1.14
5	7.19	14.46	8.06	26.49	19.15	12.46	4.30	7.43
6	-9.17	-6.95	-15.83	2.42	-10.62	-5.83	0.06	1.87
7	-3.94	1.64	-12.28	2.68	-9.60	-5.09	-6.03	-6.95
8	-2.98	3.97	-2.28	5.26	-2.15	-1.32	2.16	0.56

KLT Coefficients								
Freq	1	2	3	4	5	6	7	8
1	4.99	6.19	8.34	-51.35	-15.96	-17.41	-16.93	-2.82
2	1.86	-10.94	-8.97	-7.87	14.42	-16.28	-10.71	-0.64
3	-11.03	-5.32	5.92	4.25	-19.16	-1.98	5.02	-0.21
4	22.94	42.79	13.32	6.50	12.62	24.74	8.54	-0.60
5	9.81	19.62	7.50	27.87	20.71	9.29	1.64	5.39
6	-14.47	-11.26	-17.39	2.70	-10.40	-7.92	1.59	2.38
7	-6.21	-0.09	-14.54	-0.18	-8.68	-6.26	-4.61	-7.20
8	-3.21	4.08	-1.42	4.25	-1.57	-2.18	2.33	0.36

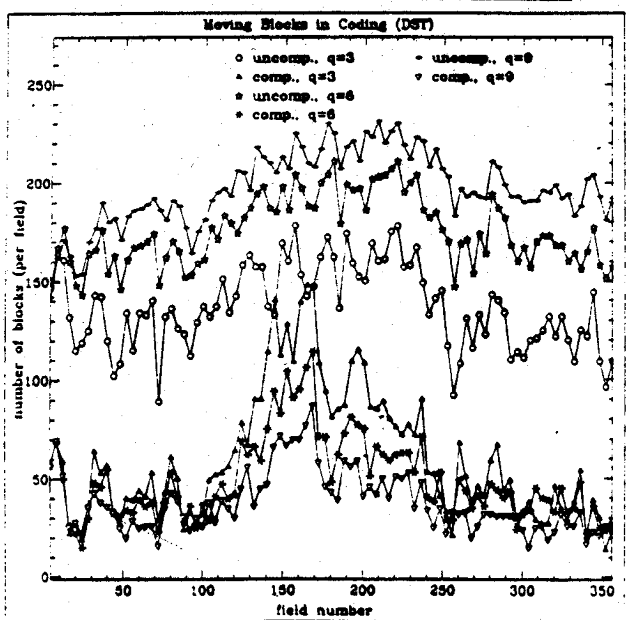

Figure 5(a) Motion Activity for Uncompensable and Compensable Blocks (DST)

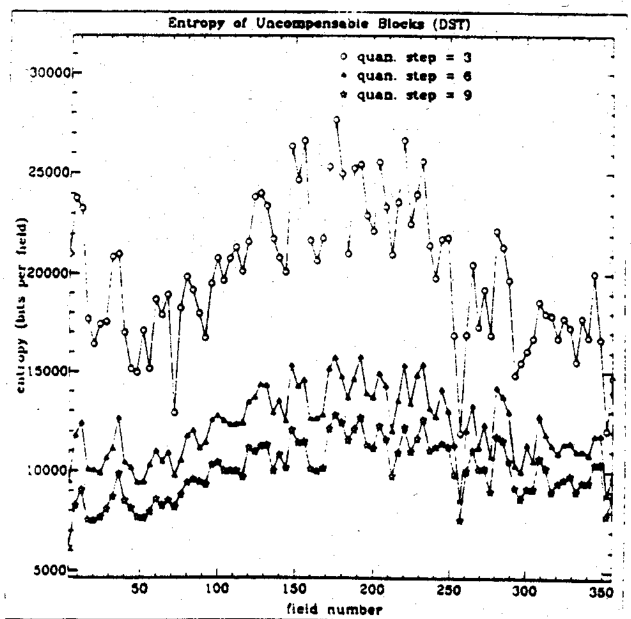

Figure 5(c) Transform Coefficient Step-size Effect on the Bit Rates (DST)