Interframe Coding with Variable Block-size Motion Compensation

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 compares several algorithms for motion-compensated interframe coding when, either a fixed block-size, or a variable block-size block-matching technique is employed for motion estimation. In the first part of the paper, we describe a variable block-size motion estimation scheme that improves the motion estimation performance by subdividing blocks of images into sub-blocks and estimating their movements. In the second part, the variable block-size technique is applied to encode the motion-compensated interframe differential pels in both the transform domain and the pel-domain coding. Our simulation results demonstrate that both transform domain and pel-domain coding algorithms benefit from the variable block-size (VBS) approach, and a pel-domain algorithm performs almost as well as a DST algorithm when the VBS approach is used.

INTRODUCTION

Previous studies [1]-[7] have shown that the block-matching motion-compensation technique significantly reduces the bits required for inter-frame video encoding. This technique first partitions an image frame (or field) into fixed-size blocks and estimates the displacements (motion vectors) for the moving blocks. Then, only the differences (so-called *motion-compensated frame differences (MCFD)*) between the current frame and the translated previous frame in the moving blocks are coded and transmitted. In general, the motion even in videoconferencing scenes is not purely translational and may result in large MCFD's when large block-sizes are used. Therefore, the use of smaller blocks improves the performance of motion compensation. This, however, also increases the computational complexity and the overhead of transmitting the increased number of motion vectors to the receiver. As a compromise between high motion-vector overhead and good motion-compensation performance, smaller block-sizes can be used only when the motion compensation on larger block-size does not perform very well. This is the basic idea of our proposed variable block-size motion-compensation algorithm. In coding the motion-compensated differential signals, the smaller block-size motion compensation is likely to produce less bits as compared to codes generated by using a larger block-size motion estimation. The bit saving due to a more accurate motion compensation may, however, be offset by the overhead required for sending the extra motion vectors. Thus we investigate the effectiveness of various coding algorithms on a real video sequence for two purposes: (1) to see the advantages offered by a variable block-size motion estimator in the real coding environment, and (2) to compare the transform-domain coding schemes with the pel-domain coding schemes.

Before further discussing details of the variable block-size motion compensation algorithm, we first briefly describe the overall coding system used in this paper. Typically, there are three basic elements in an interframe encoding system: (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. The function of a motion detector is to classify a block to be *moving* or *non-moving* before any motion estimation can be done for that block. Our motion detector here is the same as the one used in [6] and [7]. 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. A block-matching algorithm which tracks the motion between neighboring frames has a low computational complexity and a good performance as discussed in [6],[7]. The algorithm employed here uses the displacement vector of the identically located block in the previous frame as the initial match location for the current-frame block and corrects the initial estimate by a small exhaustive search around it. In order to decide if the best match has been found, the mean absolute (frame) difference (MAD) matching criterion [1] is adopted in this paper for its simplicity and good performance [6]. The MAD value also tells us how well a match has been made so that a sub-block matching may or may not be needed.

The frame differences after motion compensation (MCFD) have to be encoded and transmitted to the receiver if their values are significant. We follow the principle discussed in [7] to initially classify the image blocks into the following three types.

Type 1: *nonmoving blocks* -- The frame-difference values are mostly small and hence this type of block is identified by the motion detector as non-moving.

Type 2: *compensable moving blocks* -- These blocks are originally identified as moving by the motion detector, but after motion compensation the values in these blocks are found to be below the motion activity threshold. For such blocks only the motion vectors need to be transmitted to the receiver.

Type 3: *uncompensable moving blocks* -- These blocks are identified as moving blocks and cannot be well-compensated by the block-matching motion compensation. Both the motion vectors and the coded MCFD values are transmitted for reconstruction of these blocks. The Type 3 blocks are examined further by the variable block-size motion compensator as described in the next section.

VARIABLE BLOCK-SIZE MOTION ESTIMATION

The selection of block size in motion compensation involves a trade-off in the quality of motion compensation and the overhead of transmitting the motion vectors. A small block-size offers the advantage of achieving accurate motion estimates because the motion is likely to be more uniform across the small block, and therefore complicated movements (other than translational) can be better compensated. On the other hand, the disadvantages associated with small block-sizes are: (1) the displacement estimate is more sensitive to noise, and (2) the motion-vector overhead is larger. For large block-sizes, the above advantages and disadvantages are reversed. The variable block-size motion compensation algorithm provides a reasonable compromise as it allows some flexibility in improving motion estimates for parts of the hard-to-compensate blocks while it also tries to keep the overhead small.

The basic variable block-size (VBS) motion-compensation scheme is as follows. We first compensate all the moving blocks (identified by the motion detector) with a big block-size, and then the motion-compensated frame difference (MCFD) blocks are classified (by the motion detector again) into the compensable (Type 2) or the uncompensable (Type 3) blocks. The uncompensable (big) blocks are partitioned into sub-blocks, and the sub-block motion compensation algorithm is applied to them. After the sub-block motion compensation, the sub-block MCFD's are used to reconstruct the big blocks back together again. The (big-block) motion detector is applied to the VBS motion-compensated blocks to judge the performance of the sub-block motion compensation and categorizes them into the following subgroups:

Type 3a: *sub-block compensable moving blocks* -- With sub-block motion compensation these blocks are now classified as compensable blocks.

Type 3b: *sub-block uncompensable moving blocks* -- Even with sub-block motion compensation these blocks are still above the (block) motion activity threshold, i.e. not well compensated.

A diagram which illustrates the above procedure is shown in Fig. 1. In our experiments, the initial (big) block size is chosen to be 8×8, and the sub-block size to be 4×4.

In order to investigate the effectiveness of this variable block-size motion compensation algorithm, we apply it to a real video sequence. The original Miss 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 pels. This is typical of most movements in a video-conferencing environment.

In this part of computer simulation, we assume the coded pictures are perfectly reconstructed (i.e. no coding errors) so that only the effect of variable block-size motion compensation is demonstrated. As shown by Fig. 2(a), the fixed-size motion compensation on the 8×8 moving blocks can significantly reduce the number of blocks that actually require coding. The sub-block motion compensation (on 4×4 blocks) further reduces the number of such blocks by about 15%. Next, we observe from Fig. 2(b) that the mean square values of the motion-compensated frame-differences decreases by about 30% when the 4×4 sub-block motion compensation is used after the 8×8 block motion compensation. Thus the VBS motion compensation technique, which subdivides the 8×8 uncompensable blocks into 4×4 sub-blocks for motion compensation, could be effective in reducing bits in encoding the MCFD pels.

Fig. 3(a) shows the block diagram of an interframe encoding system with the fixed-size motion compensation. Fig. 3(b) shows an interframe encoding system with the VBS motion-compensation. The data compression and decompression in both coding systems are handled by either a transform-domain or a pel-domain coding technique. These techniques with or without the VBS approach are now discussed.

TRANSFORM DOMAIN CODING WITH AND WITHOUT VARIABLE BLOCK-SIZE MOTION COMPENSATION

Transform coding is a popular technique in both intraframe and interframe picture compression [8] Although the Discrete Cosine Transform (DCT) has been a favored coding technique for compressing the highly correlated signals in intraframe coding, we have found that the Discrete Sine Transform (DST) results in a slightly better reconstructed picture quality and a smaller bit rate as compared to DCT coding in compressing the low-correlated MCFD signals [7]. Therefore, DST is chosen for our simulations.

We use the basic coder structures of Fig. 3(a) and Fig. 3(b), and insert the transform compression algorithms into the data compression block. In the data compression block, transform of 8×8 or 4×4 uncompensable blocks, is taken and then a uniform midtread quantizer is applied to the dominant coefficients which are selected by the following rules: (1) coefficients are significant if greater than a threshold (= twice the quantization step), (2) the three highest frequency significant coefficients are discarded for 8×8 blocks whereas only one highest frequency significant coefficient is discarded for 4×4 blocks, and (3) the four lowest frequency coefficients are retained even if they are insignificant for 8×8 blocks whereas at least the 3 lowest frequency coefficients are retained for 4×4 blocks. In addition, to make final entropy bits closer to reality every coded transform coefficient assumes at least 1.1 bit.

We consider three schemes.

Scheme 1: Quantized-DST Coefficients in 8×8 blocks with fixed block-size 8×8 motion-compensation -- we quantize the selected coefficients of the Type 3 blocks and then entropy-code them.

Scheme 2: Quantized-DST Coefficients in 8×8 blocks with VBS motion-compensation -- we quantize the selected coefficients of the sub-block uncompensable 8×8 block (Type 3b) and then entropy-code them.

Scheme 3: Quantized-DST Coefficients in 4×4 sub-blocks with VBS motion-compensation -- we quantize selected coefficients of the 4×4 active uncompensable sub-block and then entropy-code them. The sub-blocks on which coding needs to be done are identified by a sub-block activity detector, which is similar to the motion detector in Section 1 with size 4×4, $T_0=3$ while $N_0=4$. Fig. 3(c) illustrates this active/inactive segmentation process.

We measure the entropy bit-rate, the mean square coding (reconstruction) error, and the number of uncompensable blocks of this DST coder for the three schemes. The results are shown in Fig 4. The number of uncompensable blocks (Fig. 4(a)), i.e., the Type 3 blocks in the fixed block-size case and the Type 3b in the VBS case, is about 25% less with VBS motion-compensation. From Fig 4(b) we find that the number of active sub-blocks is roughly two thirds of the total number of sub-blocks for the VBS motion-compensation case. Fig. 4(c) shows that the entropy bits per field for the Scheme 2 (VBS motion-compensation) is, about 20% lower on the average (roughly 2,250 bits per frame) than that of the Scheme 1 (fixed block motion-compensation). Furthermore, Scheme 3 (VBS motion-compensation and coding) is, about 12% lower in terms of entropy-bits than Scheme 2. The mean square coding errors for the uncompensable blocks (Type 3) are roughly the same for all schemes (Fig. 4(d)). The visual quality of all coded sequences are also found to be very similar, with slight preference to sequences obtained using Schemes 2 and 3.

The VBS motion-compensated coder of Scheme 2 can reduce the transform codes by 2,250 bits per frame on the average, but, it requires extra motion-vectors, which may cost about 1,000 to 2,000 bits per frame if the differential motion vectors (between the 8×8 motion-vector and the 4×4 motion-vectors in that block) are transmitted. The VBS motion-compensated coder of Scheme 3 reduces the bit rate on the average by another 1,100 bits per frame as compared to coder of Scheme 2, but, requires another overhead (200 to 500 bits per frame) for transmitting the sub-block (activity) information.

Overall, the VBS Schemes 2 and 3 provide, both statistically and visually, somewhat better performance as compared to the ordinary fixed block-size Scheme 1; however these improvements are small.

PEL DOMAIN CODING WITH AND WITHOUT VARIABLE BLOCK-SIZE MOTION COMPENSATION

As an alternative to transform domain coding approach, we consider the pel domain coding of MCFD signals. Our previous study [6],[7] shows that the significant transform coefficients in a MCFD block are usually widely spread and do not have a regular distribution pattern. In addition, it is often necessary to retain one third or more of the transform coefficients in a block to obtain a reasonable quality reconstruction. The distribution of MCFD pels in a block (in the pel domain) do not appear to be much worse. Typically, one half or less of the MCFD pels are significant and spread out in a block. This motivates us to test pel-domain compression algorithms and compare their performance to the transform domain algorithms.

We still use the basic coder structures of Fig. 3(a) and Fig. 3(b), with pel-domain compression algorithms inserted into the data compression block. Three simple pel-domain coding schemes are considered.

Scheme 1: Quantized-MCFD in 8×8 blocks with fixed block-size 8×8 motion-compensation -- we simply coarse-quantize (using a midtread) quantizer all the pels in the Type 3 blocks and then calculate the entropy of the quantized pels. Similar to that in the DST coefficient quantization case, every quantized pel assumes at least 1.1 bit.

Scheme 2: Quantized-MCFD in 8×8 blocks with VBS motion-compensation -- every pel of the sub-block uncompensable 8×8 block (Type 3b) is quantized and entropy-coded.

Scheme 3: Quantized-MCFD in 4×4 sub-blocks with VBS motion-compensation -- every pel of the 4×4 active uncompensable sub-block is quantized and entropy-coded. The sub-blocks on which coding needs to be done are identified by the same sub-block activity detector as discussed in Scheme 3 for transform coding.

Again, we compare the above three schemes based on the following criteria:
(1) number of coded MCFD pels,
(2) entropy of the coded MCFD pels, and
(3) mean square coding errors (of Type 3 blocks).

The results are shown in Figs. 5(a), 5(b) and 5(c), respectively. In terms of both the entropy bits (Fig. 5(b)) and the number of coded pels (Fig. 5(a)), Scheme 3 outperforms the other two. The mean square coding errors (Fig. 5(c)) of Scheme 3 are slightly higher than that of Scheme 2 due to the few large-magnitude MCFD pels classified as inactive sub-blocks by our active/inactive detector. The quality of reconstructed pictures is not noticeably different using any of the three pel-domain approaches.

CONCLUSION

Several block-matching motion-compensated coding schemes are compared with and without the VBS motion compensation. First, we find that both pel-domain and transform domain coding algorithms benefit from the VBS approach. Next, some pel-domain and transform-domain schemes are found to be comparable on the statistical basis for e.g. pel-domain approach (Scheme 3) and DST approach (Scheme 2) both using the same VBS motion compensation. The quantizer used in the pel-domain schemes (quantization step size = 11) is chosen slightly coarser than the one used in the DST coding scheme (quantization step size = 9) and results in nearly equal mean square coding errors in both domains. As far as the coded picture quality is concerned, the DST Schemes 2 and 3 with VBS motion-compensation are preferred to either the fixed block-size DST Scheme 1 or any of the pel-domain Schemes 1, 2 or 3. Although the quantization noise is more visible in the pel-domain coded sequences as compared to the transform-domain coded sequences, pel-domain schemes can perhaps be improved with relative ease as compared to the transform-domain approach.

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 measurement 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] A. Puri, H.-M. Hang, and D. L. Schilling, "An efficient block-matching algorithm for motion-compensated coding," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, April 1987, pp.25.4.1-25.4.4.

[7] A. Puri, H.-M. Hang, and D. L. Schilling, "Motion-compensated transform coding based on block motion-tracking algorithm," *Proc. IEEE Int. Conf. Commun.*, June 1987, pp.5.3.1-5.3.5.

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

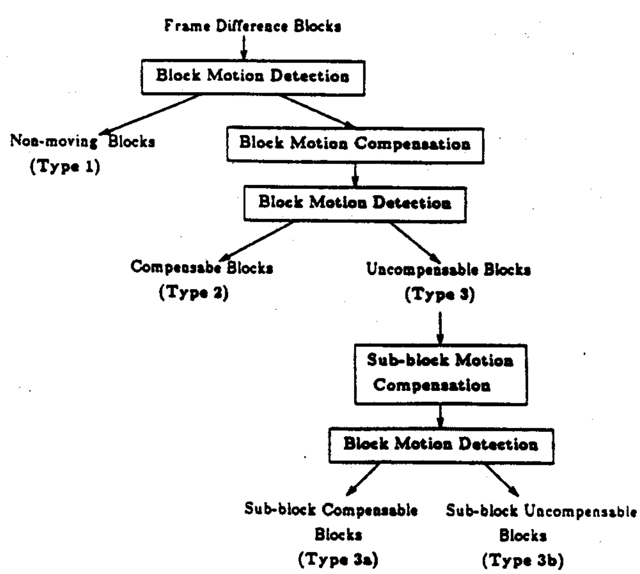

Figure 1 Frame Difference Blocks Classification

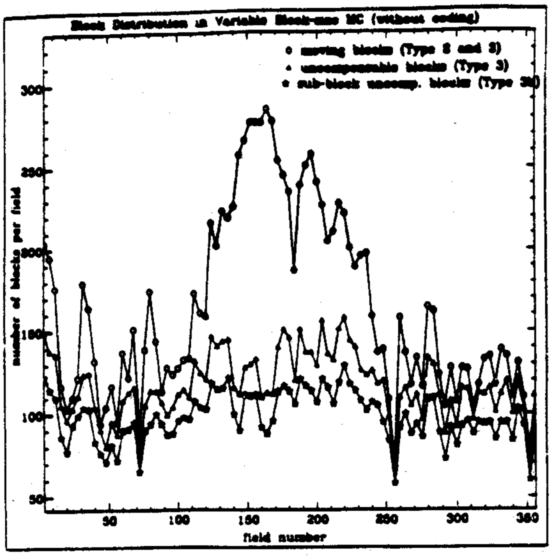

Figure 2(a) Frame Difference Block Distribution in Variable Block-size Motion Compensation (without coding)

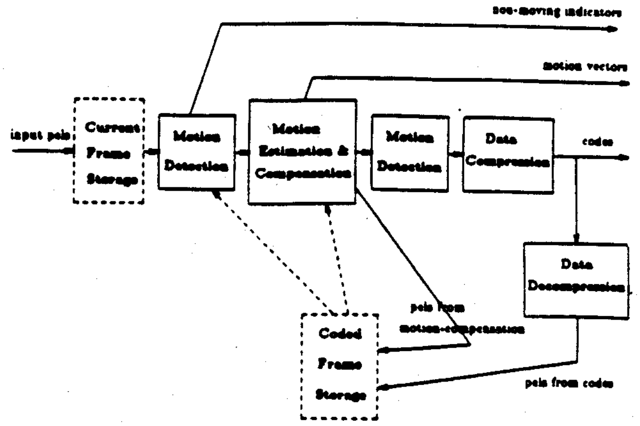

Figure 3(a) Block Diagram of the Interframe Coding System with Fixed Block-size Motion Compensation

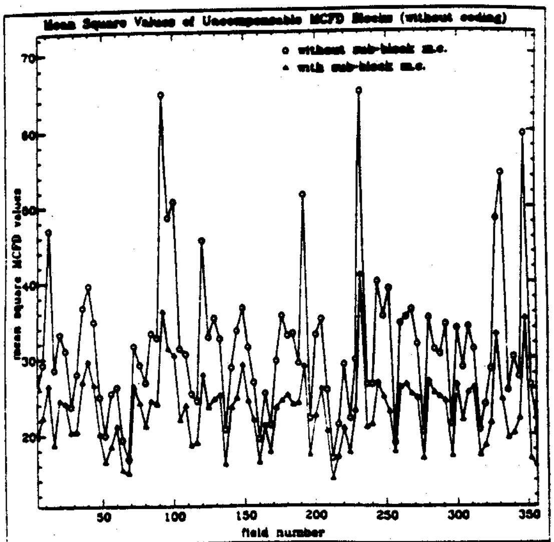

Figure 2(b) Mean Square Values of Uncompensable MCFD Blocks (without coding)

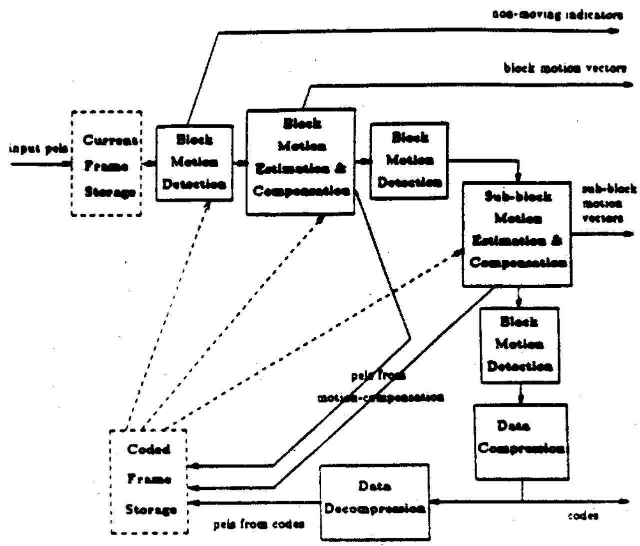

Figure 3(b) Block Diagram of the Interframe Coding System with Variable Block-size Motion Compensation

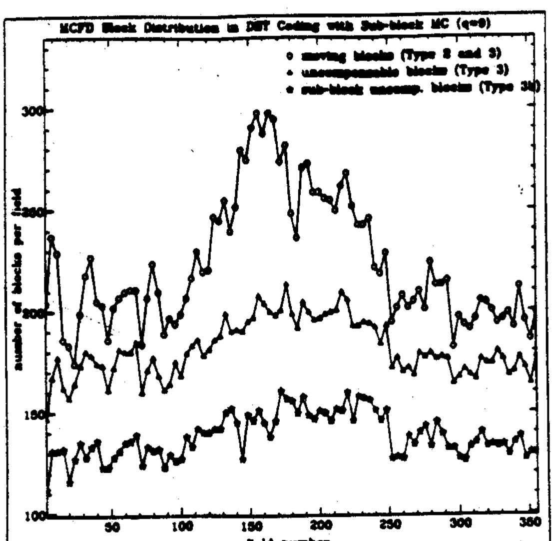

Figure 4(a) Frame Difference Block Distribution in DST Coding with Variable Block-size Motion Compensation (quant. step = 9)

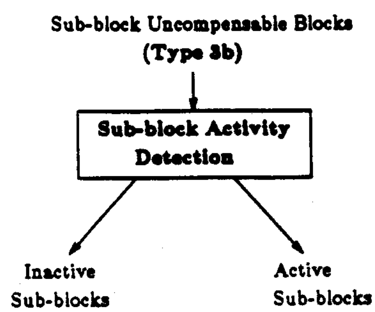

Figure 3(c) Sub-block Classification for coding

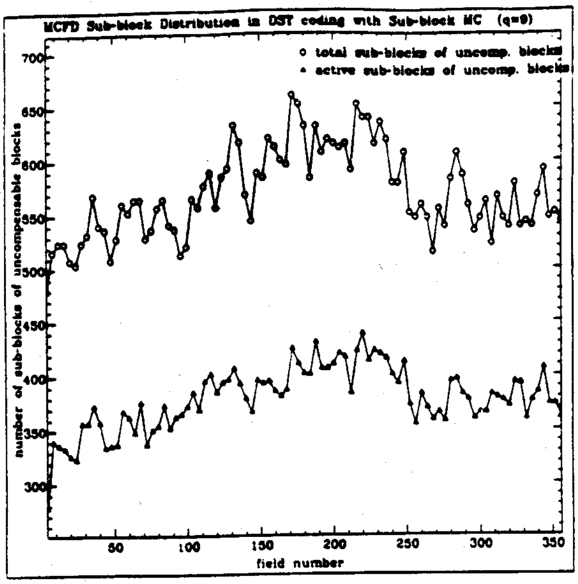

Figure 4(b) Frame Difference Sub-Block Distribution with Variable Block-size Motion Compensation

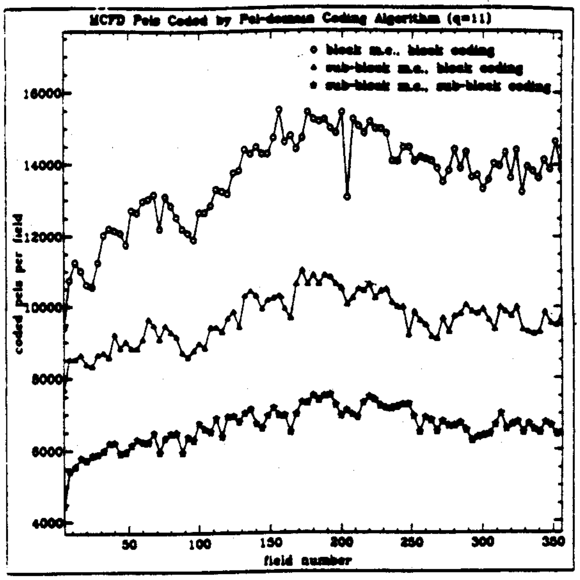

Figure 5(a) MCFD Pels Coded by Pel-domain Coding Algorithms (quant. step = 11)

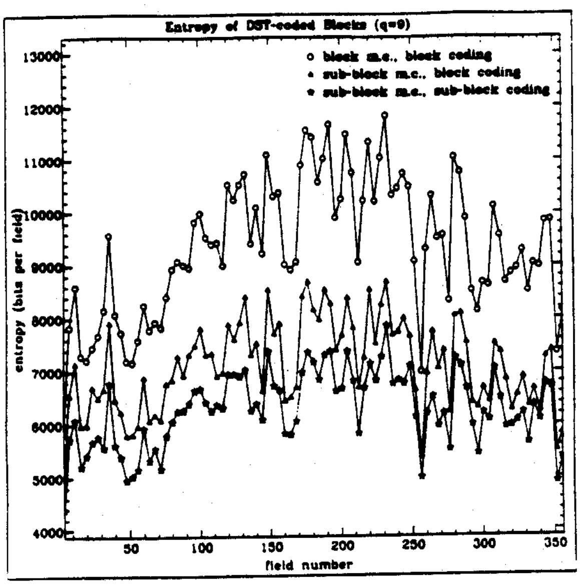

Figure 4(c) Entropy of DST-coded Blocks (quant. step = 9)

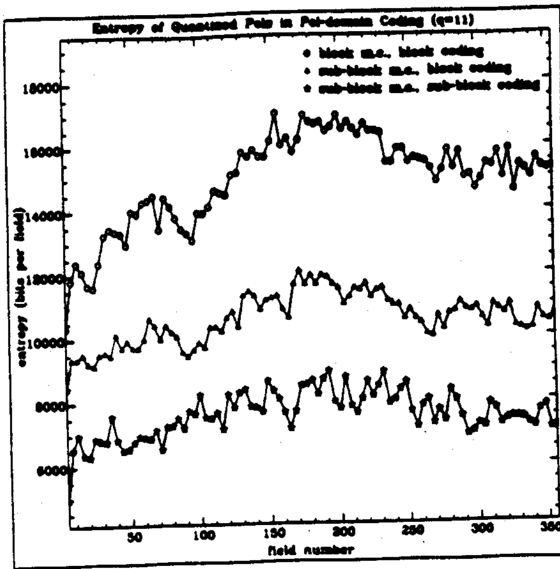

Figure 5(b) Entropy of Quantised Pels in Pel-domain Coding (quant. step = 11)

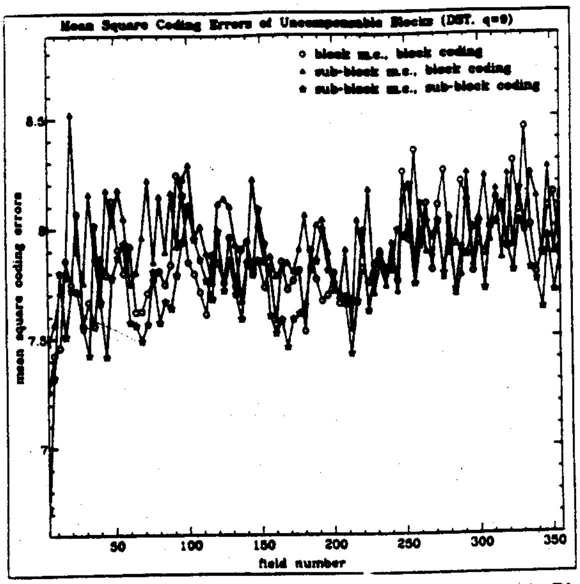

Figure 4(d) Mean Square Coding Errors of Uncompensable Blocks in DST Coding (quant. step = 9)

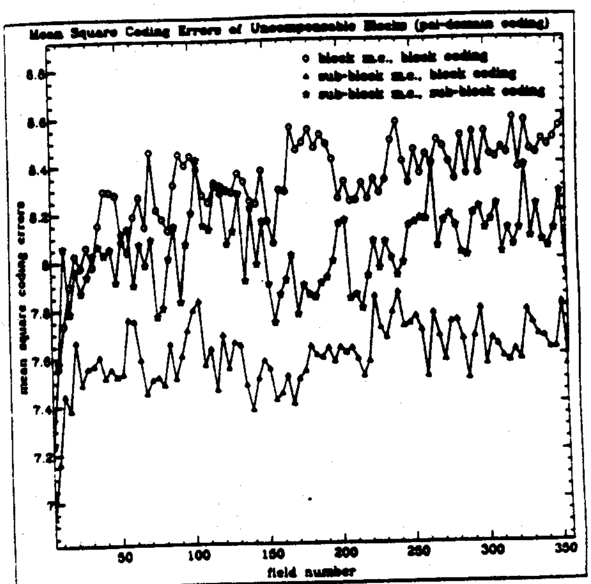

Figure 5(c) Mean Square Coding Errors of Uncompensable Blocks in Pel-domain Coding (quant. step = 11)