Entropy Coding for a Hybrid Scheme with Motion Compensation in Subprimary Rate Video Transmission

TOSHIO KOGA AND MUTSUMI OHTA

Abstract—**Entropy coding has been investigated for motion-compensated interframe (MC) prediction followed by two-dimensional discrete cosine transform (DCT) for prediction error. In particular, variable word length coding methods for motion vector and transform coefficients have been discussed assuming low bit rate such as 384 kbits/s for transmission of videoconference sequences. For motion vector information, it is advantageous to employ a one-dimensional code set common to both horizontal and vertical components of motion vectors. The code set can be obtained using a combined distribution of the two components.**

In order to encode transform coefficients, different methods are applied to significant and insignificant blocks. Run-length coding is adequate for representing clusters of insignificant blocks. In encoding transform coefficients in significant blocks, a zone coding method which encodes transform coefficients within a minimum area enclosing all nonzero coefficients is suitable. Simulation of video sequences shows that a combination of the coding methods described here can achieve high coding efficiency for videoconference sequences.

I. Introduction

In general, it is well understood that interframe prediction is very effective for TV signal coding [1]–[9]. In order to further improve the coding efficiency, motion compensation (MC) has been successfully applied to interframe prediction [10]–[19]. Block matching detection method, where motion vector detection is carried out on a block-by-block basis, shows very high efficiency when applied to video teleconference sequences [14].

For violent motion with a speed beyond motion compensation range, motion compensation is less effective than for normal motion. In this case, application of intraframe coding can improve coding efficiency, since intraframe correlation is higher than interframe correlation. To achieve high efficiency even for violent motion, an adaptive combination of motion-compensated interframe prediction and intraframe (previous sample) prediction has been developed and implemented in a practical codec [18], [19].

Although hardware complexity increases, intraframe coding efficiency can be improved when transform coding is properly applied instead of simple intraframe prediction. Transform coding, for example, two-dimensional discrete cosine transform (2D DCT) can be used for video coding [20]–[22]. It has been shown that a combination of motion-compensated interframe prediction and 2-D DCT, i.e., MC + DCT is highly efficient [20], [23]–[28].

In addition to the efficiency improvement in redundancy reduction coding, efficiency improvement in entropy coding is important as well, particularly at low bit rates. For low bit-rate video conference signal transmission, for example at 384 kbits/s or less, it becomes more important to represent zero signal information efficiently, since a vast majority of motion-compensated prediction errors is zero due to moderate motion and relatively coarse quantization. For the entropy coding of the motion-compensated prediction error, an entropy coding with run-length representation of zero signal has been reported [19]. When transform coding is applied to the MC prediction error, however, the entropy coding method needs modification taking into account the statistics of the transform coefficients [26], [29], [30].

This paper describes an analysis of entropy coding methods for motion vectors and transform coefficients resulting from a hybrid scheme of motion-compensated interframe prediction and two-dimensional transform coding. In Section II, a functional block diagram of the hybrid scheme followed by an entropy coder is depicted. Section III describes a suitable entropy coding method for motion vectors. Section IV describes several entropy coding methods for transform coefficients resulting from 2D-DCT of motion-compensated prediction error. In Section V, efficiency is compared among these methods focusing on the encoding of transform coefficients in significant blocks in terms of entropy values as well as practical Huffman code sets designed from statistics of the transform coefficients.

II. Entropy Coding for a Hybrid (MC + DCT) Encoder

A functional block diagram of a hybrid encoder is shown in Fig. 1. Prediction error is obtained by taking the difference between an input signal and a displaced signal from frame memory. A motion vector is detected in the motion detector (MD) by using the block matching algorithm [4]. The prediction error is transformed with 2D-DCT and quantized. Motion vectors and quantized coefficients are code-converted into Huffman codes [32], [33] by variable word-length (VWL) coders in the entropy coding part.

Manuscript received February 6, 1987; revised April 6, 1987. This paper was presented in part at the IEEE Global Telecommunications Conference, Houston, TX, Dec. 1–4, 1986.

The authors are with the C & C Information Technology Research Laboratories, NEC Corporation, Kawasaki, Japan 213.

IEEE Log Number 8715471.

0733-8716/87/0800-1166$01.00 © 1987 IEEE

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on November 27,2024 at 19:04:51 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2005, p. 1
Amazon.com v. Nokia
IPR2024-00921

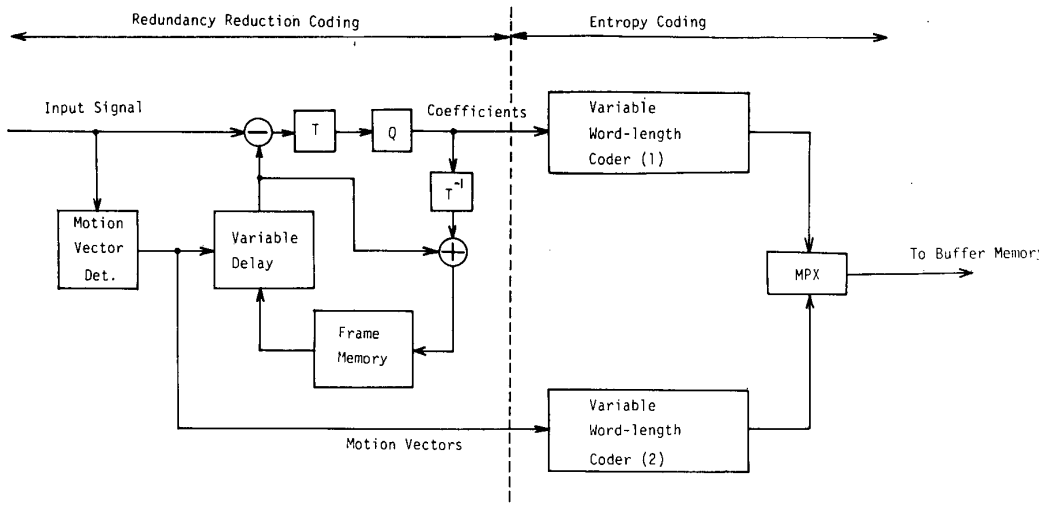


Fig. 1. Basic functional block diagram of hybrid coder with motion compensation.

Quantized coefficients are inverse-transformed with inverse 2D-DCT and used for reproducing the locally decoded signal. The locally decoded signal is stored in the frame memory (FM).

For low bit-rate transmission application, redundancy reduction should be taken into account in code conversion of the information to be transmitted in variable word-length coders as well. In particular, reduction of the information amount used in representing zero information, i.e., zero motion vectors as well as zero transform coefficients is very important. No doubt that less than one bit/pel coding cannot be realized as long as every zero coefficient is expressed with a 1-bit code, since this is a minimum length code. Most coefficients encoded in the entropy coder are of the zero level. Therefore, the entropy coder should encode the zero coefficients with few bits in order to achieve low rate transmission far below the one bit/pel rate. Along with the encoding of transform coefficients, motion vector information should be encoded efficiently.

Coding efficiency is compared in the following sections among different coding methods for motion vectors as well as for transform coefficients, assuming video teleconference signal transmission at a subprimary rate such as 384 kbits/s. Four motion video sequences typically encountered in a video teleconference situation were used for simulation. These sequences include moderate to active sequences, with one person (two sequences), three conference participants, and six persons in a split-screen mode.

III. Entropy Coding of Motion Vector Information

Since a motion vector is encoded only once every moving block, the motion vector information occupies a considerably smaller part of the whole information to be transmitted than transform coefficients. However, redundancy of even one bit/block on the average results in an information amount increase by tens of kbits/second. This leads us to adopt an efficient way of encoding the motion vector information.

Originally, motion vector can be considered as two-dimensional. For a motion compensation range of $\pm H$ samples/frame and $\pm V$ lines/frame, respectively, the number of motion vectors is given by $(2H + 1) \times (2V + 1)$, e.g., 225 for $H = V = 7$. Therefore, a number of codewords is needed in variable word-length coding of motion vectors in two dimensions, e.g., 225 for $H = V = 7$.

In order to simplify the motion vector coding, two approximation methods can be mentioned both using one-dimensional expressions [29]. Motion vectors can be represented with two components Vx (pels/frame) and Vy (lines/frame), in horizontal and vertical directions, respectively. In a two-dimensional expression (2D), a codeword is assigned to a combination of Vx and Vy. In a one-dimensional expression (1D-A), two codewords are assigned to Vx and Vy, one for Vx and the other for Vy, respectively. These two codewords belong to two different code sets each consisting of 15 codewords when the motion compensation range is assumed to be $H = V = 7$. One code set is designed from the statistics of Vx and the other from that of Vy. As a further simplification (1D-B), the two components Vx and Vy are put together to produce a combined distribution irrespective of direction. A single code set consisting of 15 codewords is derived from the distribution. The code set is commonly used for encoding the two components, resulting in two codewords for one motion vector.

Information amounts per nonzero motion vectors are shown in Fig. 2 for the three expressions, i.e., 2D, 1D-A, and 1D-B in terms of entropy. The block size used in the experiment was 8 pels × 8 lines both for motion compensation and DCT. The two 1D methods are shown to produce almost the same information rate. When compared to the 2D method, the increase in information amount from the 2D method by employing either of the two 1D methods was about 2–5 percent corresponding to the ratio of nonzero motion vectors as shown in Fig. 3.

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on November 27,2024 at 19:04:51 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2005, p. 2
Amazon.com v. Nokia
IPR2024-00921

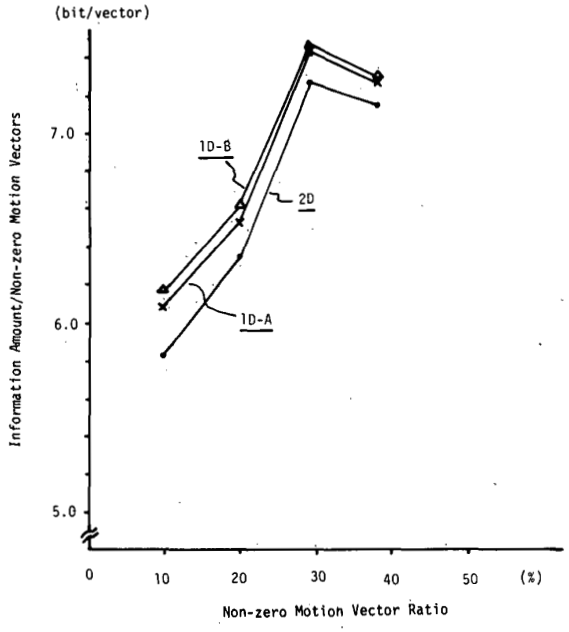


Fig. 2. Information amounts of vectors for different expressions.

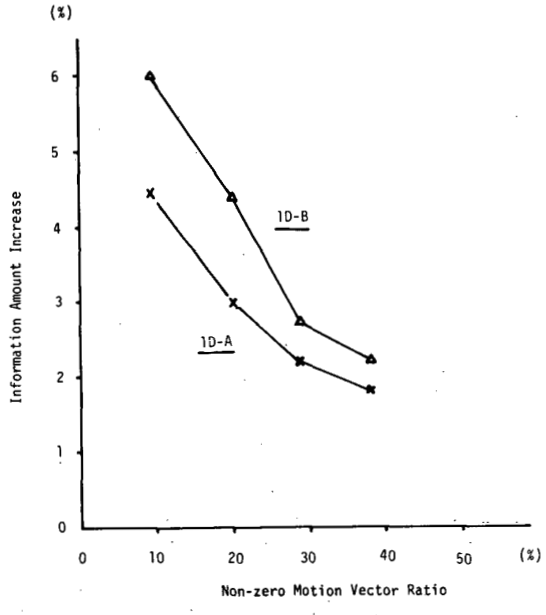


Fig. 3. Information amount increase by employing 1D coding method over the 2D method.

The figure clearly shows that the increase becomes smaller as the ratio becomes larger. The 1D-*B* produced about 1 percent higher information rate than the 1D-*A*.

Fig. 4 shows the rate increase from entropy values for motion vector information when practical Huffman code sets were designed and used for the three expressions individually. The increase is about 0.2–0.3 bit/nonzero motion vector. The 1D-*B* produces about 1 percent smaller information rate than the 1D-*A* contrary to the result

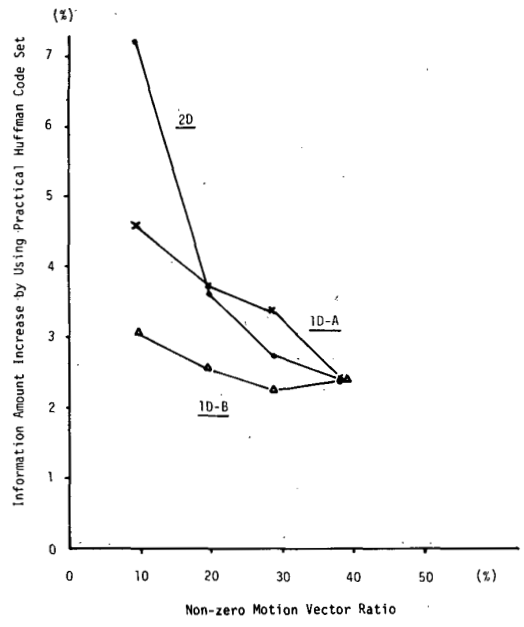


Fig. 4. Information amount increase by using practical code sets shown in Table I.

TABLE I
CODE LENGTH EXAMPLES OF MOTION VECTORS. (a) TWO-DIMENSIONAL CODES (2D). (b) ONE-DIMENSIONAL CODES (1D − *A*). (c) ONE-DIMENSIONAL CODES (1D − *B*).

(a)

Vertical	0	±1	±2	±3	±4	±5	±6	±7
±7	8	9	9	9	9	10	10	10
±6	8	9	9	9	9	10	10	10
±5	7	8	9	9	9	10	10	10
±4	7	8	9	9	9	9	10	10
±3	7	7	8	8	8	9	10	10
±2	6	7	7	8	8	9	10	10
±1	5	5	6	7	8	8	9	9
0		5	6	6	6	8	8	8
	0	±1	±2	±3	±4	±5	±6	±7
	Horizontal							

(bit)

(b)

	0	±1	±2	±3	±4	±5	±6	±7
Horizontal	2	3	4	4	4	5	6	6
Vertical	2	3	4	4	5	5	5	5

(bit)

(c)

0	±1	±2	±3	±4	±5	±6	±7
2	3	4	4	5	5	6	6

(bit)

shown in Fig. 3. The code set examples are listed in Table I for the three expressions. The maximum code length is limited to ten bits in the two-dimensional expression. Small fluctuations in motion vector distribution were smoothed when designing the code sets.

According to Figs. 3 and 4, the increase is considered to be the same and about 3 percent for both one-dimensional expressions. If these methods are compared only from the viewpoint of coding efficiency, the 2D method seems to be better, since it produces 2–3 percent lower information rate than the 1D methods. However, the motion vector is encoded only once every block. The information rate of motion vector should be evaluated in the

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on November 27,2024 at 19:04:51 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2005, p. 3
Amazon.com v. Nokia
IPR2024-00921

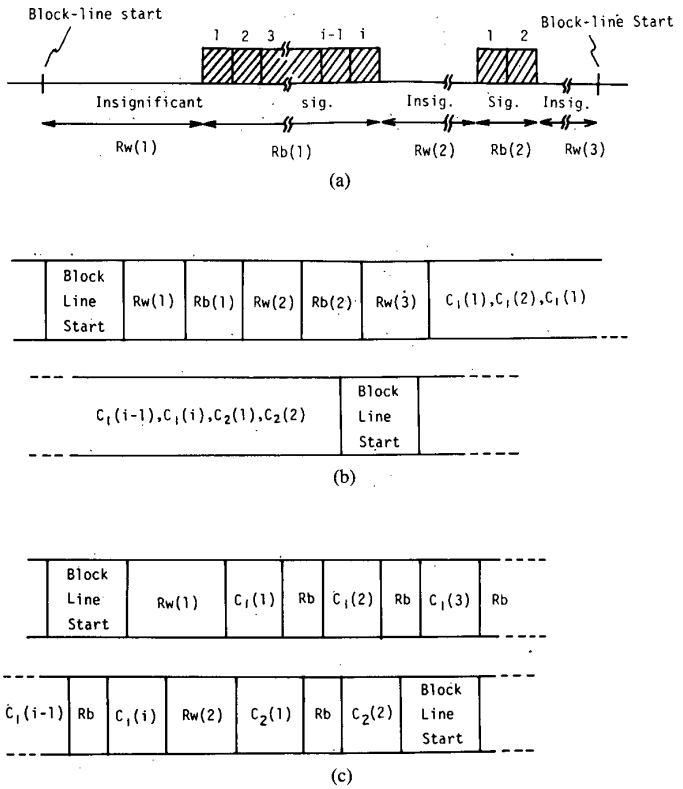


Fig. 5. Block addressing. (a) Location of significant blocks. (b) Code arrangement example (*A*). (c) Code arrangement example (*B*).

same manner as transform coefficients. That is, the information rate increase should be divided by the number of samples within a block. This corresponds to the increase by 0.03–0.08 percent. This increase is very small and can be neglected. The same conclusion can be obtained after the comparison of the two 1D methods. There is, in effect, no difference in coding efficiency among the three methods.

From the viewpoint of hardware complexity, the one-dimensional expressions will be better since the 2D expression requires 225 codewords while the 1D-*B* expression needs only 15 codewords for $H = V = 7$. The 1D-*A* method needs 30. The ratio of the number of codes is 225 to 30 or 225 to 15. For $H = V = 15$, the ratio is 961 to 60 or 961 to 31. Too many codewords are needed in the 2D method. It is concluded that the one-dimensional expression (1D-*B*) with a single code set is considered to be adequate for encoding the motion vectors.

IV. Entropy Coding of Transform Coefficients

In low bit-rate transmission, there are many insignificant blocks, wherein there are no coefficients to be transmitted. In general, these insignificant blocks appear in clusters. Therefore, it is of paramount importance to encode those blocks with few bits. A typical example will be readily mentioned, e.g., a white or insignificant block skipping method, where insignificant blocks are represented by run-length codes.

However, run-length coding is not necessarily suitable for coding of transform coefficients, since significant (nonzero) coefficients tend to appear in the vicinity of dc and/or low frequency components resulting in short runs of zero coefficients, e.g., 11.6 on the average. There will be a suitable encoding method for zero coefficients in these blocks.

A. Block Addressing

Two different methods can be mentioned to express the location of significant blocks as exemplified in Fig. 5(a). One is a method which employs run lengths of significant (*Rb*) and insignificant (*Rw*) blocks at first, then encodes coefficients information in significant blocks as shown in Fig. 5(b). For example, this process is carried out on a block line basis. A block line is a line of blocks, consisting of N scanning lines if the vertical dimension of the blocks is N lines. In this block addressing method, a block-line delay is needed for transform coefficients as a delay compensation buffer.

According to simulation results obtained from the typical video conference sequences, an average run length of significant blocks ranges from 2.5 to 4.5 blocks, and that of insignificant blocks ranges from 5.5 to 7.0 blocks. A part of the run-length distribution for significant and in-

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on November 27,2024 at 19:04:51 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2005, p. 4
Amazon.com v. Nokia
IPR2024-00921

TABLE II
A PART OF RUN-LENGTH DISTRIBUTION OF SIGNIFICANT AND INSIGNIFICANT BLOCKS

Run-length	1	2	3	4	5	6	7	8	9	10
Sig. Blocks	49.95	17.23	8.43	5.27	3.70	2.99	2.05	1.81	1.36	1.23
Insig. Blocks	32.58	16.81	9.90	6.81	5.19	3.77	2.86	2.56	2.04	1.74

(%)

significant blocks is shown in Table II. Mean run lengths of significant and insignificant blocks were 3.2 and 5.9, respectively. In designing a Huffman code set for practical use, code lengths are usually a few bits or longer. Approximately, the information rate for significant blocks will be around one bit/block.

Therefore, run-length coding is not considered to be appropriate for representing the run lengths of significant blocks. An alternative is exemplified in Fig. 5(c), where significant blocks are encoded one by one with *Rb* (1 bit) for distinction of significant and insignificant blocks. The code *Rb* represents that the current block is significant. Coded coefficients are arranged right after *Rb* except at the first significant block after insignificant block(s). This facilitates hardware implementation since the variable word-length coding is carried out on a block-by-block basis. This method needs only a one-block delay.

B. Coefficients Coding

In the transformed picture subblocks, the occurrence probability of zero-level signal is very high in high frequency coefficients. Using this property, efficient variable word-length coding can be achieved.

Fig. 6 shows the mean length of effective scanning from the beginning of each block to the last nonzero coefficient when zigzag scan is employed. "Zigzag scan" is a method to scan from lower (upper left) to higher coefficients (lower right) across a diagonal axis as shown in Fig. 7(a). For a significant block ratio ranging from about 10–60 percent, mean length was about 10–30 pels. This indicates that after the last nonzero coefficients, there are zero-coefficient runs whose lengths are about 30–50 pels when the block size is assumed to be 8 pels by 8 lines. If these zero coefficients can be represented by a several-bit code, very efficient VWL can be realized.

In the following discussions, four variable word-length coding methods are compared using the zigzag scan only for significant blocks in which one or more nonzero coefficients are included.

Basic (Reference) Method: Method A [Fig. 7(a)]: Zigzag scanning is applied to all the coefficients within a block, in conjunction with run-length coding for zero coefficients as well as Huffman coding for nonzero ones. It is well known that zero-coefficient run (zero run) becomes longer in the high frequency part and coding performance is more effective when diagonal zigzag scan is used. The information amount for nonzero coefficients is the same even if other scanning methods are employed.

Let us assume that the occurrence probability of nonzero coefficients be *Pb* and the entropy value be *Eb* which is calculated from the distribution of nonzero coefficients. The entropy/pel of nonzero coefficients is given by $Pb * Eb$. In addition, it is assumed here that the locations of zero and nonzero coefficients are run-length coded and followed by nonzero coefficient values.

Entropy per pel by method *A* is given by

$$\frac{E}{R} + Pb * Eb \tag{3.1}$$

where, E/R is a mean run-length entropy value/pel for zero and nonzero runs put together. Entropy E is calculated from the combined distribution of zero and nonzero runs, R being a mean length.

Zone Coding [Method B, Fig. 7(b)]: "Zone" is a minimum rectangle including dc coefficient and all nonzero coefficients [31]. There are 64 kinds of zones with a minimum size of $1 * 1$ and a maximum size of $8 * 8$. Zigzag scanning is assumed within this zone. Run lengths of zero coefficients can be expected to be much shorter and the number of runs can be much less than method *A*, resulting in a smaller information rate for runs. The zone index showing which zone is selected must be transmitted.

Entropy per pel by method *B* is calculated by

$$\frac{Er}{Rz} * \frac{Bz}{Bs} + \frac{Ez}{Bs} + Pb * Eb \tag{3.2}$$

where entropy *Er* is calculated from a combined distribution of zero and nonzero runs within the zones. *Rz* is a mean run length within the zones. *Ez* represents entropy calculated from the zone occurrence distribution. *Bs* represents a block size, *Bz* represents an average zone size. The first term gives run-length entropy/pel, the second term gives the entropy/pel for the zone index information to be transmitted.

Alternative Methods: Two alternative methods different from zone coding method are considered here. In these methods, the last zero run immediately after the last nonzero coefficient has appeared in each block is not transmitted, while information indicating the last nonzero coefficient is transmitted. When the last zero run is long, high efficiency is expected.

i) Scan Length Transmission [Method C, Fig. 7(c)]; It may be better to encode the scan length, e.g., the number (N) of nonzero coefficients in each block to indicate that the encoding has finished for the block [31].

Entropy per pel by this method is given by

$$\frac{En}{Rn} * \frac{Bn}{Bs} + \frac{Esc}{Bs} + Pb * Eb \tag{3.3}$$

where a distribution of zero and nonzero runs excluding the last zero run in each block is used for calculation. Entropy *En* and a mean run length *Rn* are calculated using the distribution. *Bn* represents the scan length from the beginning of a block to the last nonzero coefficients or the number of coefficients which is actually scanned. *Esc* rep-

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on November 27,2024 at 19:04:51 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2005, p. 5
Amazon.com v. Nokia
IPR2024-00921

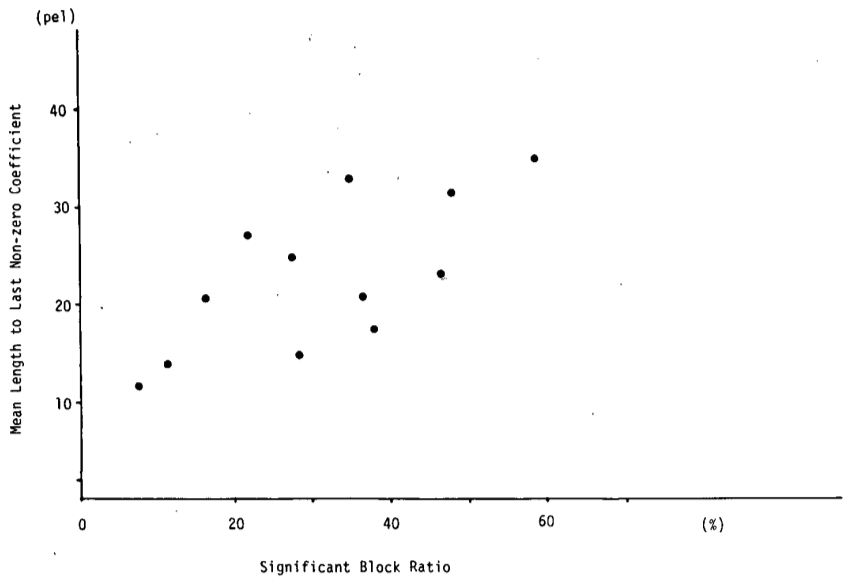


Fig. 6. Mean length from beginning to last nonzero coefficient.

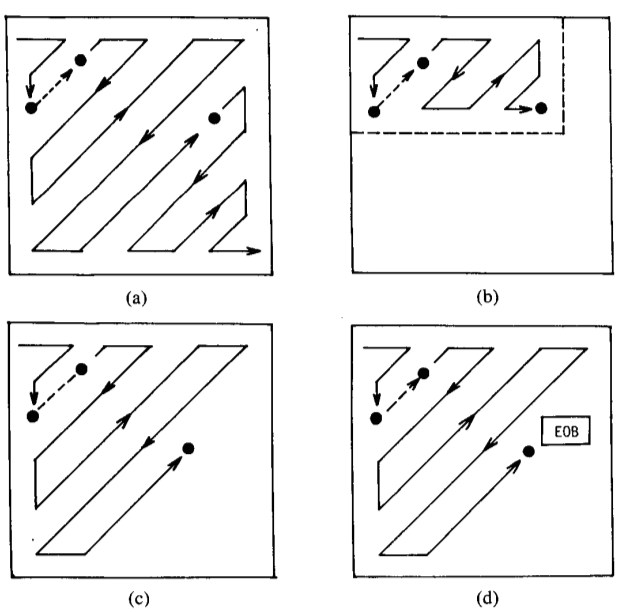


Fig. 7. Four variable word length methods. (Dots represent significant coefficients.) (a) Basic (zigzag). (b) Zone. (c) Number of significant coefficients (N). (d) End of block (EOB).

resents the entropy calculated from the distribution of a nonzero coefficients number in each block. Bn is a mean value of this distribution. The first term gives run length entropy/pel, the second term gives the entropy/pel of information indicating the end of scanning, i.e., the number of nonzero coefficients.

ii) End of Block Code Transmission [Method D, Fig. 7(d)]: Right after the last nonzero coefficients, EOB (end of block) code is added to indicate that scanning is finished in the block [23].

Entropy per pel by this method (method D) is given by

$$\frac{Ee}{Rn} * \frac{Bn}{Bs} + Pb * Eb \tag{3.4}$$

where a combined distribution of zero runs, nonzero runs, and the last nonzero runs discarding the last zero runs in each block are used for calculation. Ee is an entropy value calculated from the combined distribution. A mean run length Rn and Bn are the same as those used in (3.3).

The last term $Pb * Eb$ of (3.1)–(3.4) has the same value for all the methods since entropy of nonzero coefficients does not depend upon scanning methods. Therefore, the entropy values excluding the last term show the difference in efficiency when the four different variable word length coding methods are employed.

V. Experiment on Entropy Coding of Transform Coefficients in Significant Blocks

In order to compare the coding efficiency of the four methods described in Section IV, an experiment was carried out for actual motion video sequences. For simulation purpose, the four motion video sequences were used as in the motion vector encoding part. Block size is 8 pel by 8 lines.

Source data are encoded in the hybrid coder shown in Fig. 1. Quantizer characteristics for coefficients are shown in Fig. 8. Step size is represented by S. Three different step sizes (S = 4, 8, 12) were used in the experiment. There are +69 levels and a zero in each quantizer characteristic. When S is chosen to be 8, the average information rate was 300 kbits/s to transmit 15 frames of the moderate video sequences in a second. For a value of S from 10 to 12, the rate was about 300 kbits/s in transmitting 10 frames of the active and or split-screen sequences.

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on November 27,2024 at 19:04:51 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2005, p. 6
Amazon.com v. Nokia
IPR2024-00921

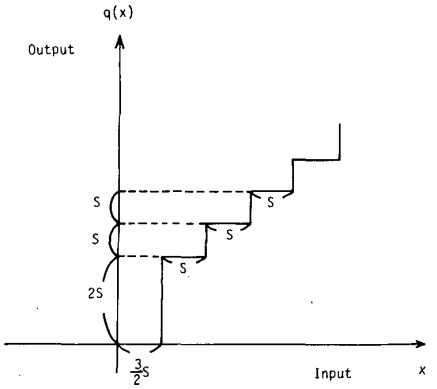


Fig. 8. Quantizer characteristics. (Step size is denoted by S. 139 levels: ± 69 + zero).

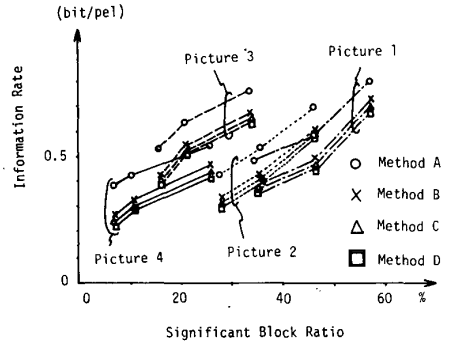


Fig. 9. Coding performance comparison. (See Fig. 7.) (Three different quantizers are used, i.e., S = 4, 8, and 12. overhead information is included.)

A. Comparison in Terms of Entropy

In Fig. 9 simulation results about the four VWL methods are shown. The vertical axis represents entropy value (bit/pel) and the horizontal axis significant block rate (percent). The three different quantizers were applied to each of the video sequences. In calculating the entropy values defined by (3.1)-(3.4), insignificant blocks are excluded.

For method A, the information rate was higher than the other three methods. Entropy of method B (zone coding) was slightly higher than the two other methods C and D (about 1-6 percent) for all the pictures and quantization step sizes. This is because the last zero run is always coded. According to Table III, the number of coded zeros was 56.8 for method A, while the number for other methods was about 20 or less.

Efficiency of methods B and C is nearly equal. If "over head information" which indicates zone or the number of significant coefficients is excluded, the information rate by method B is smaller than method C or D, since zero coefficients coded in method B were fewer than in method C or D as shown in Table III. The mean number of coefficients (zero and nonzero) coded in method B was smaller by about 5-30 percent than in method C or D, while every method generates the same number of nonzero coefficients. However, over head information rate in method B was slightly higher than that in method C or D.

B. Comparison in Terms of Practical Codes

Comparison was also made using practical Huffman codes designed for each method. These codes were designed using the distribution in the calculation of (3.1)-(3.4).

In methods B, C, and D, mean lengths of zero coefficient runs were about three to five as shown in Table III. The practical Huffman code length is usually equal to or greater than three bits when a one-bit codeword is reserved. This shows that it is not advantageous to employ run length coding for zero coefficients. It is practically suitable to assign one bit, i.e., the reserved 1-bit code for a zero coefficient.

TABLE III
CODING PERFORMANCE COMPARISON FOR FOUR METHODS. (SEE FIG. 7.) (NONZERO LEVEL INFORMATION IS NOT TAKEN INTO ACCOUNT.)

	Mean Zero Run-length	Number of Coded Zeros (per block)	Overhead (per block)
Method A	11.6	56.8	0
Method B	3.6	15.5	6.0 (Zone)
Method C	5.1	19.7	4.2 (N)
Method D	5.1	19.7	5.3 (EOB#) (bit)

(EOB# : 4 bits for EOB code and 1.3 bits due to the mean code length increase.)

Overhead information amounts in methods B, C, and D are also shown in Table III. Here, the overhead information of method B represents zone index, which is coded with a fixed length code since the distribution of zone indexes was almost uniform. Since there are 64 kinds of zones, 6 bits were assigned for the overhead information. In method C, the number of nonzero signals is the overhead information. In method D, the EOB code is the overhead information and was represented with 4 bits. The overhead information amount in method D was 5.3 bits/block. This increase from EOB (4 bits) is due to the modification of Huffman code resulting from the inclusion of EOB in the coefficient distribution.

In Fig. 10, the total information amount (excluding nonzero level code) per block is shown for each method. Method A produced the highest information rate, and then D, C, and B.

This comparison result is a little different from that in entropy value comparison. This is because one bit is assigned for each zero coefficient within effective scanning areas in methods B, C, and D. In practice, one bit is needed to represent a zero unless run-length coding is employed. In method B, least zero signals were coded. Although there was little information rate difference among the three methods, method B produced least information rate. In addition, method B transmits zone information which can be used for adaptive quantization of transform coefficients as well as for selecting a suitable scanning method in coding the coefficients within the zone.

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on November 27,2024 at 19:04:51 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2005, p. 7
Amazon.com v. Nokia
IPR2024-00921

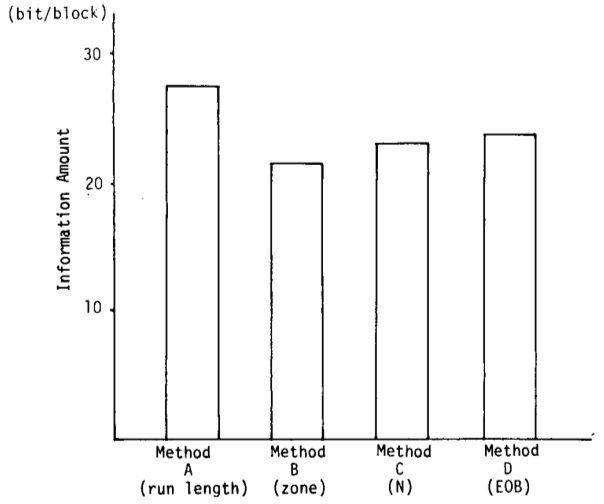


Fig. 10. Total amounts of codes for four methods. (Nonzero level information is excluded.)

Therefore, method *B* (zone coding) seems to be most promising. Essentially, this indicates that a method enclosing nonzero coefficients with the least number of zeros is effective.

For further efficiency improvement, there will be a possible combination of zone method and number or EOB method. In addition, employment of multiple scanning orders in horizontal, vertical, zigzag, and other scanning directions will be effective if they are appropriately applied to nonzero transform coefficients. Optimization of the Huffman code set also helps information amount reduction.

VI. Conclusion

Without an efficient entropy coding algorithm, low bit-rate transmission at a subprimary rate or below would be impossible. In this paper, a practical entropy coding algorithm for low bit-rate motion video transmission has been discussed for a hybrid scheme of motion-compensated interframe prediction and transform coding.

For the motion vector encoding method, it is advantageous to employ a common one-dimensionl VWL code set for both horizontal and vertical components. As for coding the coefficients, it is advantageous to combine a block addressing (insignificant block skipping) method to represent locations of significant blocks, and a smallest zone method in which nonzero coefficients in significant blocks are encoded with minimum zero coefficients enclosed by the zone. Coefficient values are Huffman coded with a minimum length of one bit.

Acknowledgment

The authors wish to thank T. Ishiguro, K. Niwa, Y. Iijima, and K. Iinuma for fruitful discussions and their encouragement throughout this work.

References

[1] A. N. Netravali and J. O. Limb, "Picture coding: A review." *Proc. IEEE*, vol. 68, pp. 366–406, Mar. 1980.
[2] H. Kaneko and T. Ishiguro, "Digital television transmission using bandwidth compression techniques," *IEEE Communications*, vol. 18, pp. 14–22, July 1980.
[3] J. C. Candy, M. A. Franke, B. G. Haskell, and F. W. Mounts, "Transmitting television as clusters of frame-to-frame differences," *Bell Syst. Tech. J.*, vol. 50, pp. 1889–1917, July–Aug. 1971.
[4] T. Ishiguro, K. Iinuma, Y. Iijima, T. Koga, and H. Kaneko, "NETEC system: Interframe encoder for NTSC color television signals," in *Proc. 3rd Int. Conf. Digital Satellite Commun.*, Kyoto, Japan, Nov. 1975, pp. 309–314.
[5] K. Iinuma, Y. Iijima, T. Ishiguro, H. Kaneko, and S. Shigaki, "Interframe coding of 4-MHz color television signals," *IEEE Trans. Commun.*, vol. COM-23, pp. 1461–1466, Dec. 1975.
[6] H. Yasuda, H. Kuroda, H. Kawanishi, F. Kanaya, and H. Hashimoto, "Transmitting 4-MHz TV signals by combinational difference coding," *IEEE Trans. Commun.*, vol. COM-25, pp. 508–516, May 1977.
[7] H. Kaneko, Y. Iijima, T. Ishiguro, and K. Iinuma, "NETEC-6/3 video transmission equipment for teleconference," in *Proc. INTELCOM 79 Exposition*, Dallas, TX, Feb. 1979, pp. 579–582.
[8] T. S. Duffy and R. C. Nicol, "A codec for international visual teleconferencing," in *Proc. GLOBECOM'82*, Miami, FL, Nov. 1982, p. E2.6.1.
[9] N. Mukawa, H. Kuroda, and T. Matsuoka, "An interframe coding system for video teleconferencing signal transmission at a 1.5 Mb/s rate," *IEEE Trans. Commun.*, vol. COM-32, pp. 280–287, Mar. 1984.
[10] T. Ishiguro and K. Iinuma, "Television bandwidth compression transmission by motion-compensated interframe coding," *IEEE Communications*, vol. 20, pp. 24–30, Nov. 1982.
[11] M. G. Musmann, P. Pirsch, and H.-J. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. 73, pp. 523–548, Apr. 1985.
[12] A. N. Netravali and J. D. Robbins, "Motion compensated television coding: Part-I," *Bell Syst. Tech. J.*, vol. 58, pp. 629–668, Mar. 1979.
[13] 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.
[14] 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. 1981.
[15] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Commun.*, vol. COM-33, pp. 888–896, Aug. 1985.
[16] T. Koga, A. Hirano, K. Iinuma, Y. Iijima, and T. Ishiguro, "A 1.5 MB/s interframe codec with motion compensation," in *Proc. Int. Conf. Commun.*, Boston, MA, June 1983, p. D8.7.1.
[17] H. Murakami, S. Matsumoto, Y. Hatori, H. Yamamoto *et al.*, "A 15 Mbit/s universal codec for TV signals using a median adaptive prediction coding method," in *Proc. 6th Int. Conf. Digital Satellite Commun.*, Phoenix, AZ, Sept. 1983, pp. VII-A-17–VII-A-24.
[18] K. Matsuda, T. Tsuda, T. Itoh, and S. Maki, "A new motion compensation coding scheme for video conferences," in *Proc. Int. Conf. Commun.*, Amsterdam, The Netherlands, May 1984, pp. 234–237.
[19] T. Koga, K. Iinuma, K. Niwa, S. Tsugane, M. Nishiwaki, and Y. Iijima, "Sub-T1 rate motion video codec for teleconference," in *Proc. Int. Conf. Commun.*, Chicago, IL, June 1985, p. 2.4.1.
[20] T. Koga, A. Furukawa, J. Ohki, and S. Tsugane, "A 384 kbit/s motion video codec with scene change detection," in *Proc. Int. Conf. Commun.*, Toronto, Ont., Canada, June 1986, pp. 12.4.1–12.4.5.
[21] A. K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, pp. 349–384, Mar. 1981.
[22] K. R. Rao, Ed., *Discrete Transforms and Their Applications.* New York: Van Nostrand Reinhold, 1985.
[23] W. H. Chen and W. K. Pratt, "Scene adaptive coder," *IEEE Trans. Commun.*, vol. COM-32, pp. 225–232, Mar. 1984.
[24] 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.
[25] Y. Kato, N. Mukawa, S. Okubo, H. Hashimoto, and H. Yasuda, "Discrete cosine transform coding for video conferencing using vector and scalar quantization," in *Proc. GLOBECOM'86*, Houston, TX, Dec. 1986, pp. 8.4.1–8.4.5.
[26] M. Ohta and T. Koga, "Adaptive VWL coding of transform coeffi-

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on November 27,2024 at 19:04:51 UTC from IEEE Xplore. Restrictions apply.

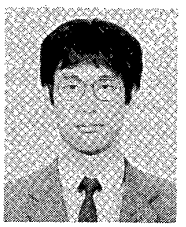
Nokia Exhibit 2005, p. 8
Amazon.com v. Nokia
IPR2024-00921

cients for subprimary rate video transmission," in *Proc. GLOBECOM'86*, Houston, TX, 1986, pp. 8.5.1-8.5.5.
[27] M. Kaneko, Y. Hatori, A. Koike, and H. Yamamoto, "Improvement transform coding algorithm for motion-compensated frame-to-frame difference signals," in *Proc. GLOBECOM'86*, Houston, TX, 1986, pp. 8.6.1.-8.6.5.
[28] R. Srinivasan and K. R. Rao, "Motion-compensated coder for video-conferencing," *IEEE Trans. Commun.*, vol. COM-35, pp. 297-304, Mar. 1987.
[29] Specialists Group on Coding for Visual Telephony (CCITT SG-XV), Meet. Rep. 7, Nurnberg, Nov. 1986.
[30] H. Brusewitz and P. Weiss, "A video conference system at 384 Kbit/s," in *Proc. PCS'86*, Tokyo, Japan, Apr. 1986, p. 212.
[31] Specialists Group on Coding for Visual Telephony (CCITT SG-XV), Meet. Rep. 5, Tokyo, Japan, May 1986.
[32] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE*, vol. 40, p. 1098, Sept. 1952.
[33] H. Murakami, S. Matsumoto, and H. Yamamoto, "Algorithm for construction of variable length code with limited maximum word length," *IEEE Trans. Commun.*, vol. COM-32, pp. 1157-1159, Oct. 1984.

Toshio Koga received the B.S. and M. S. degrees in electronics from Kyushu University, Fukuoka, Japan, in 1969 and 1971, respectively.

Since joining the Central Research Laboratories of NEC Corporation in 1971, he has been engaged in research on bandwidth compression of television signal and visual communication systems. He is currently Research Manager in the Home Electronics Research Lab, C & C Information Technology Research Labs.

Mr. Koga is a member of the Institute of Electronics, Information and Communication Engineers of Japan and of the Institute of Television Engineers of Japan.



Mutsumi Ohta was born in Osaka, Japan, on Nov. 22, 1958. He received the B.S. and M.S. degrees in engineering from Kyoto University, Kyoto, Japan, in 1981 and 1983, respectively.

Since he joined NEC Corporation in 1983, he has been engaged in the development of video signal coding algorithms. Currently, his research interest is development of motion video encoding for low bit-rate transmission. He is a research staff member of Home Electronics Research Lab., C & C Information Technology Research Labs., NEC Corporation.

Mr. Ohta is a member of the Institute of Electronics, Information and Communication Engineers of Japan.

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on November 27,2024 at 19:04:51 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2005, p. 9
Amazon.com v. Nokia
IPR2024-00921