Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 741–748
ISBN 978-83-60810-27-9
ISSN 1896-7094

Basic Consideration of MPEG-2 Coded File Entropy and Lossless Re-encoding

Kazuo Ohzeki Yuǎn yù Wei
Shibaura Institute of Technology Toyosu, Koutouku, Tokyo, 135-8548 Japan
Email: {ohzeki,m710101}@sic.shibaura-it.ac.jp

Eizaburo Iwata
Universal Robot Inc.
Toyosu, Aoumi, Kotoku Tokyo Japan
Email : eiza@urobot.co.jp

Ulrich Speidel
University of Auckland
Tamaki – 331, Morrin Road, Glen Inne, Auckland New Zealand
Email: ulrich@cs.auckland.ac.nz

Abstract—**Re-encoding of once compressed files is one of the difficult challenges in measuring the efficiency of coding methods. Variable length coding with a variable source delimiting scheme is a promising method for improving re-encoding efficiency. Analyses of coded files with fixed length delimiting and with variable length delimiting are reviewed. Motion vector codes of MPEG-2 encoded files are modified as a variable-to-variable coding point of view. Length, bit-rates, and varieties of videos are examined. The largest file is 16 seconds of D1 full size at 720 ×480 among five video files. By entropy evaluation, an improvement of almost 20% in coding efficiency over the conventional MPEG-2 is obtained.**

I. Introduction

In this paper, for coded files, such as MPEG-2 encoded ones, redundancy will be evaluated to provide re-encoding information. There are four frameworks of encoding according to whether input event length and output bit length is fixed or variable. One example is fixed length input and fixed-length output that is an F-F type. In the same manner, there are three other types of F-V, V-F and V-V. Among these types, V-V is the most general and has the greatest possibilities in realizing the most efficient encoding method [1-2]. However, the V-V type has the difficult problem of how to delimit the input events into most efficient length. Universal coding and arithmetic coding are flexible methods. But they are basically still F-V types. Based on several examples of V-V trials, we will show an example of V-V re-encoding. The method improves a compression ratio of 20% compared to the MPEG-2 standard method.

Examples of V-V coding methods are reported in several papers. Jacob presented that the V-F codes are better than the F-V codes for finite alphabets of K-th order ergodic Markov sources [1]. Yamamoto et al mentioned that a VF code is called proper if the set of parse strings of the code satisfies the prefix condition [3]. He also described that non-proper VF codes should be considered in order to realize efficient VF coding. This implies that variable parsing without considering prefix condition has a large possibility to enhance coding efficiency. Abrahams presented a survey paper on the theoretical literature on fixed-to-variable-length lossless source code trees, called code trees, and on variable-length-to-fixed lossless source code trees, called parse trees [4]. He focused on Huffman coding and Tunstall V-F coding for parsing and making tree methods with a large bibliography for further investigation of algorithmic and performance perspectives. Matsui et al examined the compression ratio of the Tunstall-Huffman code. The Tunstall-Huffman code is a variable-to-variable code. They obtained better results by the Tunstall-Huffman V-V coding than with stand-alone coding methods such as the Tunstall V-F coding or the Huffman F-V coding [5].

All these V-V coding are theoretical not for video files. MPEG-2 coding of video files is not proved to be an optimum method. In fact one of the authors has shown reduandancy of coded MPEG-2 files using FV codes. In this paper, we will try to enhance to use VF codes for MPEG-2 coded files. Though the aplication is restricted to motion vector parts, coding efficiency of the VV codes is much larger than that of the FV case.

The proposed method is an improvement of the MPEG-2 coding. So it is not compatible with the standard methods.

As for V-V coding, there have been few papers. The reason may be that it is difficult to parse sequences at optimal delimiting points. To cope with this problem, an example of review trials will be presented in this paper. A fixed length analysis of sequences was first implemented. The length of the fixed parsing was incrementally examined. Many heuristic trials were carried out and problems with this analysis will be listed.

To conduct variable parsing of sequences, the T-code generating method [6] was taken up. T-code was introduced by Titchener in 1984 [7]. Though the generating process is only a parsing process, it may be an influential tool to optimal variable length coding. It generates codes by copying pre-generated shorter codes. T-code generating experiments were carried out for more than 50 different video sequences. Efficient parsing is itself efficient coding. How to combine efficiency conditions with a parsing algorithm will be a further problem.

According to coding of motion vectors, Yu et al presented two-dimensional motion vector coding for low bit-rate video phones. However, their Huffman coding was generated by the JPEG procedure and the number of motion vector bits were about only one-frame for full D1 digital video size, 720×480 [8]. Shimizu et al proposed a method using representation of norm and angle for motion vectors. The method was complicated and still two separate codes were used [9]. Matsuda et al proposed a lossless re-encoding scheme for MPEG-1 video. They used an arithmetic coder for both DCT and motion vector data for the MPEG-1 coder [10]. In the following sections, an exemplified review of the fixed parsing method, and a variable parsing method with T-code are presented for long-term investigation. Then a practical variable length coding will be presented with several variations of construction parameters.

II. Re-Encoding of MPEG-2 coded files

I. Fixed Length Analysis

Before considering V-V coding, we will review F-V coding to evaluate actual entropy. There is redundancy in the MPEG coded bit-stream. There are two problems in evaluating redundancy for coded bit-streams as statistical data with a long interval. These are quantity and quality. How much coded data should be prepared? How many kinds of pictures should be prepared? For this problem, we introduce a new inverted distribution model. Based on the model, we analyze entropy space and show that the distribution is symmetrically uniform on a distorted circle at isentropic space cut out by the same entropy plane. Though the model is uniformly distributed, actual random samples of coded bit-stream statistics are not uniformly distributed as in the case of the model. They are located in a small region by our experimental results. We conclude that the MPEG coded bit-stream is not random, but is very much correlated. Based on these results, we evaluate entropy of coded bit-streams. For the interval of 20bit, the entropy value is 0.9.

If the encoder is an ideal one, the output of the encoder should be a perfect random number, which cannot be re-compressed at all into a file of smaller size. For evaluating ideal encoder performance, random number analysis is one of the most effective methods. Among several methods of random number analysis, entropy evaluation is used.

How much bit data is needed to get reliable results? A relation between the length of the bit stream and the bit pattern interval was obtained as formula (1),

$$n \simeq m \cdot 2^{L} \cdot \log_e 2 \ , \qquad (1)$$

where L is an interval length of the bit pattern, 2^{-m} is the level of significance, and n is the required length of the bit stream to be analyzed. Using the formula (1), we can obtain a relation between the length of a bit stream and the interval of a bit pattern to be analyzed with a probabilistic parameter "m". For a coded bit stream with a certain length, using the formula (1), we can get the length of interval "L", which assures accuracy of the result with a risk probability of $\frac{1}{2^m}$.

To consider how many kinds of pictures and coding parameters we should prepare for valid results, we introduced entropy space and cut it with an equi-entropy plane. The entropy per bit is Ent(R) =1.0. The probability distributions give a specific entropy value between 0<Entropy<1.

Fig. 1 shows isentropic curves that are plotted with probability points with the same entropy values. This figure is the case of a computational low dimension. We can utilize this idea for higher dimensional cases because we cannot exactly analyze all cases but some sample cases among all. If we take two points at random on one of these isentropic curves, these two points are not always close at hand, but rather located on symmetrically opposite each other in many cases. The two points that locate in completely symmetrical positions with each other have inverted probability distributions. If we make a sum set of the distribution of these two points with respect to their occurrence frequency, the distribution of the sum set is flat and its entropy should be 1.0. To observe these behaviors, we choose all patterns on the isentropic curves. Then, taking two samples from this pattern set at random, we put them together to form a new point. The new point locates midway between two points in Fig.1, whose entropy increases because the new point moves in an inner direction in entropy space. This method is formalized as an evaluating method 1

Method 1: S1. Evaluate entropy of files, F1,F2,…. S2. Select files, Fs1,Fs2,…,Fsn, with the same entropy value E, e.g. E=0.9 S3. Make combined files Cp=U(Fsi,Fsj), i≠j S4. Evaluate Entropy of Cp S5. Evaluate increase of Entropy from S1 to S4.

If there is no difference between the entropy values of S1 and S4, then the original files are strongly correlated located in a narrow region. If there is a difference, the maximum value indicates the distribution of the original files. If the maximum value is not 1.0, then there should be redundancy in the set of original files.

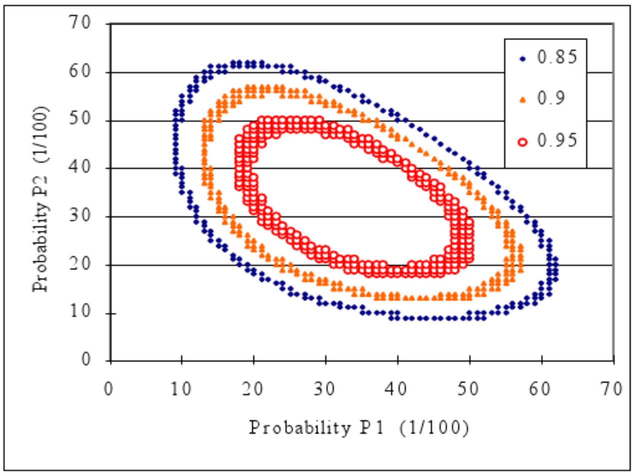

Fig. 1 Isentropic curves for all probabilistic data.

II. Variable Length Analysis

To construct an optimal V-V coding, analysis of input sequences and parsing of the sequence is important. Savari et al presented an analysis of variable-variable length codes [11]. It found that The Tunstall V-F codes can be considered to have an equi-probable code set. Then, combining the Tunstall V-F code with Huffman F-V code will provide higher efficiency. Matsui experimentally proved the concept using text files [5]. In this section, T-code analysis which was carried out in [12] will be introduced.

Fig. 2 shows an example that accomplishes better performance for the V-V encoding scheme than for the F-V encoding scheme. There is a 44-bit sequence in the number column of Fig. 2(d). Parsing this sequence by the fixed length of two bits, we get Fig. 2(a). For three events, allocating one and two bit codes, we get in total 35 coded bits. Next, parsing this sequence by the fixed length of three bits, we get Fig. 2(b). For four events, allocating two bit codes, we get in total 28 coded bits. On the other hand, in the variable coding case, Fig. 2(c) shows variable code example. There are 20 coded bits in total. Fig. 3 shows another four bit fixed code case. The total number of bits is either 21 or 22. Still the variable case has a smaller number of bits.

This example shows that there is at least a better performance variable code than that of all fixed codes within a designated code length.

An example of the T-code generation is described in Fig. 4. The result after parsing is not necessarily the prefix condition. However, the parsed codes represent the original

2 bit pattern	freq	code length	bits	code length	bits
0001	4	1	4	2	8
0010	1	3	3	2	2
1000	4	2	8	2	8
1001	2	3	6	2	4
			21		22

Fig. 3 4 bit code case.

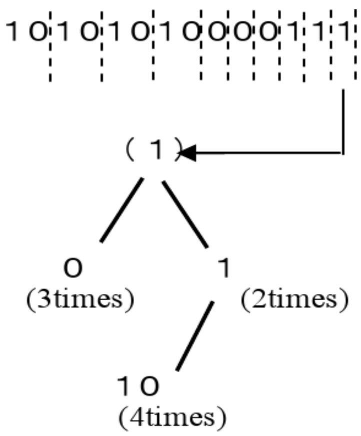

Fig. 4 T-code generation example

sequence and may provide an influential tool to design a variable parsing code set. According to the T-code generation rule [6], detaching the rightmost bit “1”, at first code the second bit “1” from the right. In this case, “1”appears twice and proceeds to two bits to the left. Next, for the “0”, code newly “0”. Further, the “0” appears three times and counts three for the “0”. Then, another “0” appears, which is the fourth time. But in this case, as there is another “1”, “10” becomes a newly defined code as an extended new code from the previous generated codes. Further, this “10” code appears four times. The detailed generation rule is described using recursive formulation.

2bit pattern	freq	code length	bits
00	9	1	9
01	6	2	14
10	7	2	12
11	0	-	0
			35+

Fig2 (a) 2 bit code case

32bit pattern	freq	code length	bits
000	2	2	4
001	4	2	8
010	4	2	8
011	0	-	
100	4	2	8
101	0	-	
110	0	-	
111	0	-	
			28+

Fig2 (b) 3 bit code case

V-bit pattern	freq	code length	bits
1000	8	1	8
100	3	2	6
10	1	3	3
1	1	3	3
			20

Fig2(c) Variable bit case

No.	2bit	3bit	V
1	1	1	1
2	0	0	0
3	0	0	0
4	0	0	0
5	1	1	1
6	0	0	0
7	0	0	0
8	0	0	0
9	1	1	1
10	0	0	0
11	0	0	0
12	1	1	1
13	0	0	0
14	0	0	0
15	0	0	0
16	1	1	1
17	0	0	0
18	0	0	0
19	0	0	0
20	1	1	1
21	0	0	0
22	0	0	0
23	1	1	1
24	0	0	0
25	1	1	1
26	0	0	0
27	0	0	0
28	0	0	0
29	1	1	1
30	0	0	0
31	0	0	0
32	0	0	0
33	1	1	1
34	0	0	0
35	0	0	0
36	1	1	1
37	0	0	0
38	0	0	0
39	0	0	0
40	1	1	1
41	0	0	0
42	0	0	0
43	0	0	0
44	1	1	1

Fig2 (d) Input Sequence and three code case

III. Experimental Results

A. Fixed Length Coding

The authors analyzed MPEG-2 coded files to re-encode them. The length of the fixed delimiting interval is more than 20 bits. The variety of 20 bit data is about one million. The file length in which these 20 bit patterns appear once is about 2.5 Mbytes. For valid statistical evaluation, at least 10 instances of each pattern should appear on average, which means that an input file needs to be about 25MB. Table I shows the necessary sizes of file and corresponding original video lengths needed for measuring bit length by assuming a ten times occurrence for the files.

Table I
Necessary sizes of file and corresponding original video lengths needed for measuring bit length by assuming ten time occurrence for the files.

bit	Number of patterns (2^{bit})	Size of files	Video lengths（6Mbps）
20	1048576	26MB	35 Sec
30	1073741824	40GB	15 hour
40	1099511627776	55TB	848 days

B. T-code Analysis

Here, to obtain optimal delimiting methods of unknown bit streams in general, for the first step, T-code analysis is carried out. This is only half of the total coding design. But as the first step, we analyze bit streams by T-code and investigate the resultant entropy behavior. Table II shows entropies of generated T-codes for 50 different videos. The values are about half. This implies that the T-code generates codes in a balanced manner. Table III shows increase behaviors when combining two files. Table IV shows further results of combining three files and five files. These values are all normalized to input single bit, and the entropy value of 0.5 means the compressed size is 1/2 of the original size. This entropy is not the so-called T-entropy in [6].

Fig. 5 shows the increasing tendency of T-code entropy when combining a number of files. At the number of five files, the saturation tendency can be seen. The calculation time for T-code analysis takes a long time, and it is limited to evaluation for longer files.

C. Re-encoding of MPEG-2 Coded Files

Based on the concepts above, the coding efficiency of MPEG-2 coded files is shown as a V-V coding paradigm. Fig 6 shows a V-V coding design and coding execution.

According to coding of motion vectors, Yu et al presented two-dimensional motion vector coding for low bit-rate video phone. However, their Huffman coding was generated by a JPEG procedure and the numbers of motion vector bits were equivalent to about only one-frame for full D1 digital video size, 720×480 [8]. Shimizu et al proposed a method using representation of norm and angle for motion vectors. The method was complicated and still two separate codes were used [9]. Matsuda et al proposed a lossless re-encoding scheme for MPEG-1 video. They used an arithmetic coder to both DCT and motion vector data for the MPEG-1 coder [10].

To realize the whole system, we will propose a re-encoding method that integrates two existing variable codes into a single code as the first step. Based on the information theory, blocking source inputs brings efficiency to the lower bound of entropy. For the Markov source, blocking gains more efficient results. In actual MPEG-2 encoding, 2D-VLC is the only example of blocking source events with run-length of zeros and amplitude in DCT coefficient coding. We constructed block coding for motion vector coding in this paper. Motion vectors of MPEG-2 consist of a set of symmetrical 16 variable-length-codes from 1-bit to 10-bit and one bit code for a zero vector. In the following parts of this section,

Table II.
Entropies of T-codes for videos.

video	entropy	video	entropy
01.mpg	0.534372	26.mpg	0.524720
02.mpg	0.539007	27.mpg	0.523816
03.mpg	0.529123	28.mpg	0.526155
04.mpg	0.532101	29.mpg	0.539802
05.mpg	0.539262	30.mpg	0.535905
06.mpg	0.538966	31.mpg	0.540882
07.mpg	0.537607	32.mpg	0.536436
08.mpg	0.532928	33.mpg	0.534316
09.mpg	0.529504	34.mpg	0.537676
10.mpg	0.535544	35.mpg	0.534722
11.mpg	0.527797	36.mpg	0.537318
12.mpg	0.524526	37.mpg	0.533381
13.mpg	0.525225	38.mpg	0.536857
14.mpg	0.528234	39.mpg	0.540498
15.mpg	0.533929	40.mpg	0.540714
16.mpg	0.531385	41.mpg	0.539724
17.mpg	0.535190	42.mpg	0.538251
18.mpg	0.527166	43.mpg	0.539320
19.mpg	0.522915	44.mpg	0.531534
20.mpg	0.538911	45.mpg	0.542534
21.mpg	0.530039	46.mpg	0.528326
22.mpg	0.538085	47.mpg	0.526850
23.mpg	0.520653	48.mpg	0.532450
24.mpg	0.537217	49.mpg	0.534462
25.mpg	0.525208	50.mpg	0.533274

Table III.
T-code entropy for combined files.

videos	entropy
01+02.mpg	0.537676
11+12.mpg	0.528685
21+22.mpg	0.537929
31+32.mpg	0.543184
41+42.mpg	0.545325

Table IV
T-code entropy for combined files. Three files and five files case.

videos	entropy
01+02+03.mpg	0.541919
11+12+13.mpg	0.527247
21+22+23.mpg	0.538656
31+32+33.mpg	0.543422
41+42+43.mpg	0.547081
01+02+03+04+05.mpg	0.538796

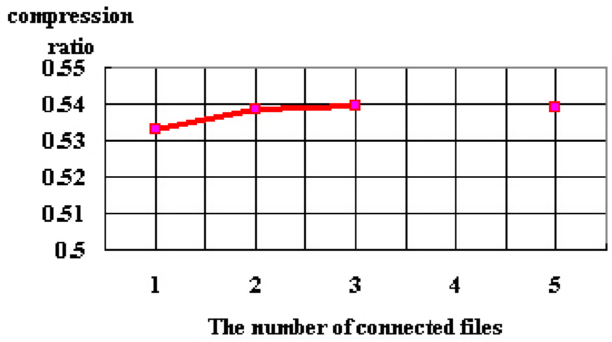

Fig.5 Increasing tendency of T-code's entropy vs. the number of connected files.

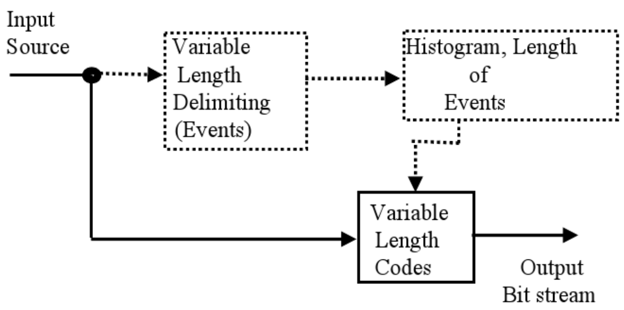

Fig.6 Design of V-V coding and re-encoding.

several experiments are carried out to improve the coding algorithm.

I. Symmetrical coding

In this subsection, 16 different non-zero events and a zero event are coded in a pair of two consecutive original VLCs. There is another sign bit to represent positive or negative for non-zero motion vectors. In this experiment, the sign bit is a fixed one bit, and is excluded for the calculation of efficiency. This scheme is called "Symmetrical coding" because positive and negative evens are coded by the same codes. The motion vectors appear as a pair of horizontal and vertical vectors as shown in Fig.7. For this format, we can block the horizontal vector and the vertical one in a single code.

Table V shows the frequency of motion vectors in encoding a video by MPEG-2 encoder, TM-5. The video size is full D1 (720×480) and the length is half a second.

Table VI shows entropy of coded bits with improving ratios of compression rates. The entropy of blocked motion vectors is reduced 27% at most from the original MPEG-2 coded bits. Table VII shows a part of the constructed codes.

Table V
Frequency of motion vectors of a video. (without sign bit consideration, video 'car' 0.5 sec)

mv	1	2	3	4	5	6
freq	50276	7784	1748	840	501	389
mv	7	8	9	10	11	12
freq	470	623	267	138	187	135
mv	13	14	15	16	17	
freq	253	126	230	257	50	

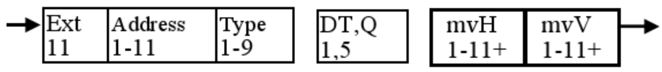

Fig.7 MPEG-2 macroblock layer structure. Ext=Macroblock_extension_code, DT=DCT_type, Q=macroblock quantization step, mvH=motion vector for horizontal direction, mvV=for vertical direction. The numerical values in the bottom sections of boxes are allocated bits for the codes.

Table VI
V-V re-encoding of motion vectors. MPEG-2 TM-5. (without sign bit consideration)

Measuring scheme	Bits	Improvement ratio
Coded bits of mv	1.62 (bit/mv)	1.0
Entropy of mv	1.30 (bit/mv)	0.80
Entropy of 2mv	2.38 (bit/2mv)	0.73
Coded bits of 2mv	2.47 (bit/2mv)	0.77
Coded bits of 2mv per mv	1.24 (bit/1mv)	0.77

Table VII
V-V codes for a pair of motion codes. (part) (without sign bit consideration)

Index	Code Value :	Bit pattern	Bit
1	1 :	1	1
2	15 :	01111	5
3	33 :	0100001	7
4	79 :	01001111	8
5	436 :	0110110100	10
6	291 :	100100011	10
7	851 :	1101010011	11
8	616 :	1001101000	11
9	2743 :	0101010110111	13
10	3539 :	0110111010011	13
11	2688 :	0101010000000	13
12	2742 :	0101010110110	13
13	5378 :	01010100000010	14
14	2905 :	0101101011001	13
15	10759:	010101000000111	15
16	10758:	010101000000110	15
17	0	00	2
18	29	011101	6
19	219 :	011011011	9

The Huffman codes are generated using free software by Marcus Geelnard available at http://bcl.comli.eu/. Table VIII shows a part of the constructed codes.

II. Influence of Video Length

In this subsection, we examine the necessary length of input video for these experiments. It is better to use video test sequences as far as possible to examine the performance pre-

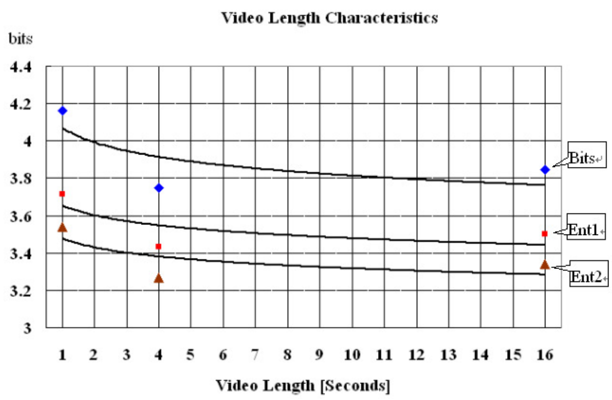

Fig.8 Video length influence of MPEG-2 re-encoding for the video No. 4, at 8Mbps.

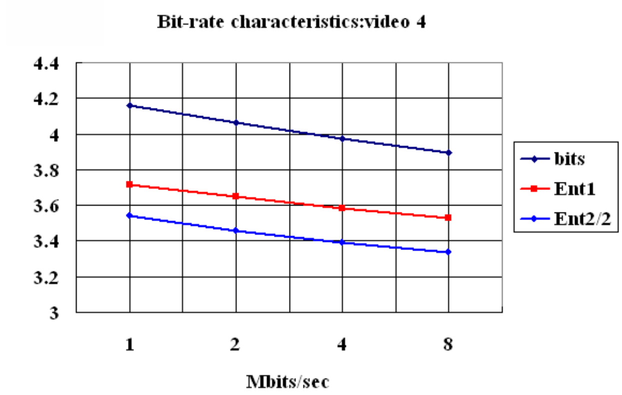

Fig. 9 Bit rate characteristics for the video No.4 with a duration of four seconds.

cisely. On the other hand, it is better to keep time and data volume to a minimum. New results of re-encoding for several cut-out partial sequences from a single video are listed in Table 8. Fig. 8 is a graph showing re-encoding. From the start of 0.5 seconds, information decreases gradually toward 16 seconds. The least squares regression lines on the graph can be gradually decreased and nearly converge. The videos used in this paper are listed in Table 9. The video used to examine of the influence of video length is No.4 "autobahn". The size of the motion picture is 720x480.

The bit-rate of encoding is 8Mbps. Sixteen different non-zero events and a zero event including the sign bit to represent positive or negative for non-zero motion vectors (MV) are coded in a pair of two consecutive original VLCs. In this experiment, the sign bit is included in newly generated VLCs and for the calculation of efficiency.

Viewing these behaviors, we choose the video length of 4 seconds for the following experiments.

III. Influence of bit-rate

In this subsection, the influence of bit-rate is examined. There are many choices of bit-rates in MPEG-2 encoding. It is important to check the effects of bit-rate to re-encoding code design. Fig. 9 shows bit-rate characteristics with log-scale for the horizontal axis. Bits mean the original MPEG-2 motion vectors. Ent1 means entropy of motion vectors of MPEG-2. Ent2/2 means entropy of motion vectors obtained by the proposed re-encoding method. In general, a decrease can be seen with bit-rates. But improving ratios from MPEG-2 to re-encoding may be the same. The entropy is larger for low bit-rates, which means a role of motion vectors is large and may require more bit-rate for describing videos. For high bit-rates, the smaller entropy means that there may be redundancy and all motion vectors are not necessarily required. These understandings coincide with the former comments that improvement of motion vector coding is effective for low bit-rates in references [8] and [9].

IV. Evaluation of a variety of videos

Table VIII and Fig 10 show overall comparison of re-encoding efficiency. The first column of Table VIII is the num-

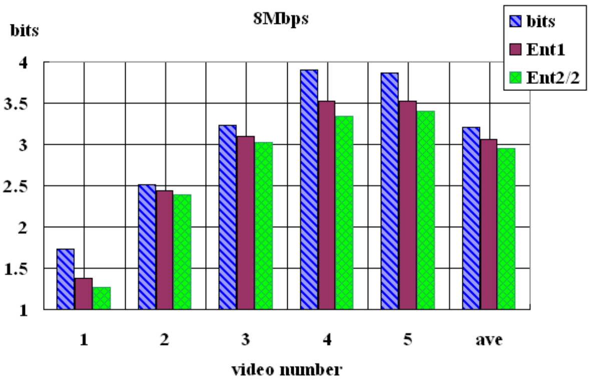

Fig. 10 Overall comparison of re-encoding efficiency. Video length is four seconds for No. 1-5. Bit-rate is 8Mbps.

bers of coded bits of motion vectors for the case of the original MPEG-2. Ent1 means entropy of motion vectors of MPEG-2. Ent2/2 means entropy of motion vectors in the case of the re-encoding method. In Fig.10, improvement ratios from MPEG-2 to re-encoding are large for videos No. 1, No.4 and No.5, but are small for videos No.2 and No.3. The characteristic of videos No.2 and No.3 is relatively smaller motion. On the other hand, videos No.1, No.4 and No.5 have large motion scenes. Ave in Fig. 10 means the average of five results. Table IX are videos used in these experiments.

V. Comparison of quantity of the number of MVs

Table X shows increased bits of motion vectors used in our experiments in this paper. A large number of video data are used to analyze the methods in detail and to improve reliability of the experiments. About 30 times more than the conventional experiments carried out by Yu et al. [8].

Fig.11 (a)-(d) are sample pictures of videos used in these experiments except (e) which is a rugby game on television. The former four videos are presented at author's homepage [13].

IV. Conclusion

A new re-encoding paradigm is reviewed and a two dimensional semi-optimization is examined. A large number of

Table VIII
Video length influence of MPEG-2 re-encoding. Video is No.4, 8Mbps.

Length of video [Sec]	bits	Ent1	Ent2/2
0.5	3.95	3.57	3.37
1.0	3.90	3.53	3.34
4.0	3.75	3.43	3.27
16.0	3.84	3.50	3.34

Table IX
Video sequences used in this paper.

No.	name	content / original size
		content
		original size
1	car	a taxi left to right
		SD:720x480
2	giraffe	jiggle by hand movement
		SD: 720x480
3	cherry	swinging cherry blossom in wind
		HD: 1920x1080
4	autobahn	highway driving
		HD:1920x1080
5	rugby(75)	rugby game in television
		SD: 720x480

Table X
Increased bits of motion vectors in this paper compared to the conventional paper [8].

Yu's experiments [8]		Our experiments	
video sequences	Bits of motion vectors	video sequences	Bits of motion vectors
Miss Am	6987	1.car	110464
Mother & Daughter	9135	2.giraffe	397424
Salesman	5377	3.cherry	412304
Car Phone	14961	4.autobahn	246924
Foreman	22105	5.rugby	615714
total	58565	total	1782830

video data are used to analyze the methods in detail and to improve reliability of the experiments. By two-dimensional Huffman re-encoding by V-V codes, 5-26% coding efficiency is obtained. Though the result is restricted to the motion vector parts, the efficieny improves much comparing to the conventional methods using F-V codes for MPEG-2 with 5% coding efficiency.The results may be stable as to length of video and bit-rates. However, for the variety of video contents, the result may still not be convergent. This implies that a larger variety of videos, including still scenes and large motion ones should be tried in the future.

Fig. 11(a) Video 1 car

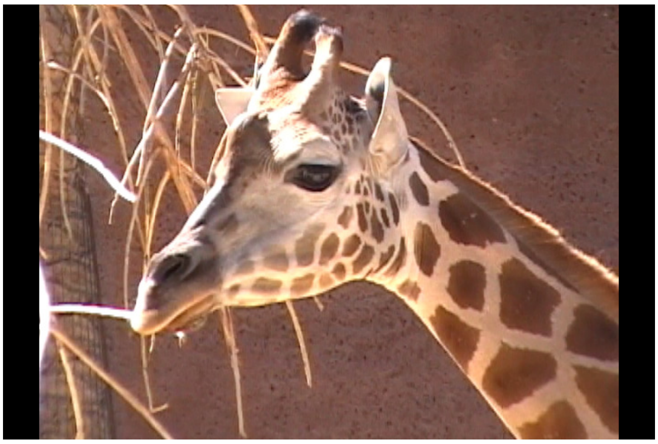

Fig. 11(b) Video 2 giraffe

Fig. 11(c) Video 3 cherry

Acknowledgment

Kazuo Ohzeki, the author, thanks Dr. Mark Titchener who provided tcalc software to calculate T-code in this paper. He also thanks Mr. T. Kato who carried out experiments of T-code analysis.

References

[1] Jacob Ziv. Variable-to-fixed length codes are better than fixed-to-variable length codes for Markov sources. IEEE Trans. IT, 36(4): pp861-863, July 1990.

Fig. 11(d) Video 4 autobahn

[2] Te Sun Han, Kingo Kobayashi , 2001. *Mathematics of Information and Coding*. American Mathematical Society, Boston, MA, USA. **Book**.

[3] H. Yamamoto and H.Yokoo, “Average-Sense Optimality and Competitive Optimality for Almost Instantaneous VF Codes”, IEEE Trans. IT, 47(6): pp.2174-2184, Sept. 2001.

[4] J. Abrahams, "Code and parse trees for lossless source encoding ", Proceedings of Compression and Complexity of Sequences pp.145 - 171 , Jun. 1997.

[5] Yudai Matsui and Takuya Kida, "Study on Efficiency of Tunstall-Huffman Code", Proc. of Data Engineering and Information Management 2009 Demi Forum i1-28 (in Japanese) http://db-event.jpn.org/deim2009/proceedings/files/i1-28.pdf

[6] Ulrich Gunther et al., “Representing Variable-Length Codes in Fixed-Length T-Depletion Format in Encodes and Decoders”, J. of Universal Computer Science Vol. 3 No. 11 pp.1207-1225, Springer. 1997.

[7] Mark Titchener, "Digital encoding by means of new T-codes to provide improved data synchronization and message integrity", Technical Note, IEE Proceedings, Volume: 131, Pt. E, Number: 4 , July 1984, Page(s): 51 –53.

[8] Guo Yao Yu, and Cheng-Tie Chen, "Two-dimensional motion vector coding for low bit rate videophone applications", Proc. ICIP 1995, vol. 2, pp. 2414.

[9] Astushi SHIMIZU, Astushi SAGATA, Kazuto KAMIKURA,Naoki KOBAYASHI, " Motion Vector Coding by Using Representation of Norm and Angle Components", IEICE Trans. J84-D-II(11), 2379-2386, 2001. (in Japanese).

[10] Ichiro Matsuda, Kei Wakabayashi, Yu Ikeda and Susumu Itoh "A Lossless Re-encoding Scheme for MPEG-1 Video",Proceedings of 17th European Signal Processing Conference (EUSIPCO-2009), pp.1834-1838.

[11] S.A, Savari, W. Szpankowski, "On the analysis of variable-to-variable length codes ", Proceedings. 2002 IEEE International Symposium on Information Theory,page176, 2002.

[12] Kazuo OHZEKI, Tsuyoshi KATO, and Engyoku GI, “Basic consideration for lossless re-encoding of MPEG coded files using V-V codes”, IEICE Technical Report, IE2010-6, pp.31-36, April, 2010. (in Japanese).

[13] http://www.sic.shibaura-it.ac.jp/~ohzeki/oz4c/mmap/videos/index.html