비디오 압축을 위한 영상간 차분 DCT 계수의 문맥값 기반 부호화 방법

이 진 학, 김 재 균*
현대전자산업㈜
*한국과학기술원 전기및전자공학과

Context-based coding of inter-frame DCT coefficients for video compression

Jin-hak Lee and Jae-kyoon Kim*
Hyundai Electronics Industries Co., Ltd.
*Dept. of Electrical Engineering, KAIST

Abstract

This paper proposes context-based coding methods for variable length coding of inter-frame DCT coefficients. The proposed methods classify run-level symbols depending on the preceding coefficients. No extra overhead needs to be transmitted, since the information of the previously transmitted coefficients is used for classification. Two entropy coding methods, arithmetic coding and Huffman coding, are used for the proposed context-based coding. For Huffman coding, there is no complexity increase from the current standards by using the existing inter/intra VLC tables. Experimental results show that the proposed methods give ~ 19% bits gain and ~ 0.8 dB PSNR improvement for adaptive inter/intra VLC table selection, and ~ 37% bits gain and ~ 2.7dB PSNR improvement for arithmetic coding over the current standards, MPEG-4 and H.263. Also, the proposed methods obtain larger gain for small quantizaton parameters and the sequences with fast and complex motion. Therefore, for high quality video coding, the proposed methods have more advantage.

1 Introduction

Most of current video compression standards (MPEG-1,2,4, H.261, H.263) use a motion compensated prediction and a two dimensional discrete cosine transform (2D-DCT) on the differential signal. The DCT decorrelates the signal and compacts the energy of an image block into low-pass transform coefficients. These coefficients are quantized for lossy compression with high compression ratio, and the quantized coeffcients are entropy coded.

In the entropy coding of current standards, all of the coefficients are ordered into a one-dimensional array by means of a pre-defined zigzag scan pattern as shown in Fig. 1. And then run-level pairs are encoded, where runs denote the number of zero coefficients until the next non-zero coefficient of a certain level.

For the variable length coding (VLC) of the run-level pairs, most of the standards employ a single VLC table in the inter-frame coding. This approach does not exploit how the statistics of the run-level pairs depend on the image motion, quantization step sizes, positions in the block, and others.

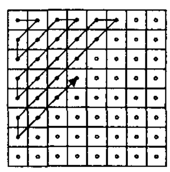


Fig. 1: ZigZag scan of 2D-DCT coefficients

Considering coding efficiency, classification without overhead always reduces the entropy. Not to transmit additional overhead for the classification information, pre-determined rules that both encoder and decoder agree on are used. Classification methods based on the order of the run-level symbols [1], starting position of the zero-run [3,4,5], or quantization step size [4,5] have been developed. Such classifications would not require any additional coding overhead to indicate which class a run-level symbol belongs to.

Symbol statistics may vary according to many factors as well as positions of coefficients and quantization step size. For example, if an image sequence contains fast or complex motion, DCT blocks produce many symbols with short zero-run and large level. So, the statistics for the symbols vary on a block-by-block basis. Considering this property, VLC table selection method sending extra overhead information the table of which produces less bits was proposed in [6]. However, coding efficiency cannot be good in the low bit-rate because of the overhead bit.

In this paper, we propose a context-based coding method to classify run-level symbols depending on the preceding coefficients. No extra overhead needs to be transmitted since the information of the previously transmitted coefficient is used for classification. Intuitively, in the case of the blocks that have many symbols with short zero-run and large level, there are many coefficients with large level in the neighborhood of symbols being coded. Hence, we can efficiently classify the symbols according to the contexts.

2 Context-based coding of DCT coefficients

In a DCT-based coding system, motion compensated difference is divided into 8X8 blocks and 2-D DCT is performed on each block. Because the DCT has a good energy packing property, a quantized DCT block usually has a lot of zeros in the lower-right corner and a few non-zero values in the upper-left corner of a block. Two examples of a quantized DCT block are shown in Fig. 2. The coefficients in the blocks are scanned in a zigzag fashion and re-ordered into a one-dimensional array as shown in Fig. 3.

In H.263 [1] and MPEG-4 [2] which have been recently standardized, (run-level, end) symbols are encoded rather than run-level-pairs, where 'end' is a flag indicating whether the currently encoded non-zero coefficient is the last one in the block. In two examples of Fig. 3, (run-level, end) symbols are presented on the non-zero coefficients. The level means the absolute value of the non-zero coefficient and one-bit sign information is transmitted for each symbol.

The statistics for the symbols vary, in general, on a block-by-block basis and even on the positions in the block: The statistics of chrominance are different from that of luminance because of its smaller bandwidth; A small value of quantization parameter and complex motion produce coefficients with larger level in the block; Also coefficients with large level reside in the low frequency region in the block because of the energy packing property of DCT.

Another property is that there are coefficients with large levels in the neighborhood of the symbols that have short zero-runs and large levels as stated before. This property efficiently classifies the symbols irrespective of various factors including quantization parameter, position, motion, and others. In Fig. 3, it is easily observed that there are coefficients with large levels in front of symbols with short zero-runs and large levels.

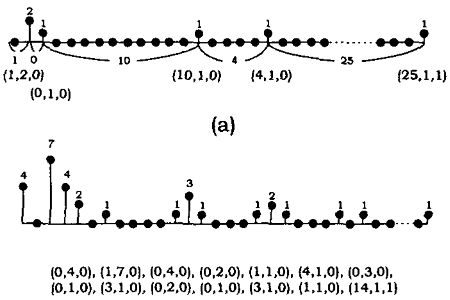
0	-2	0	0	-1	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	-1	0	0	0	-1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a)

-4	0	0	-1	0	0	0	0
-7	-2	0	1	0	0	0	0
4	0	-3	-1	1	0	0	0
0	1	2	0	0	-1	0	0
0	-1	-1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b)

Fig. 2: Examples of quantized 2-D DCT block



(b)

Fig. 3: 1-D representation of zigzag scanned DCT coefficients and (run, level, end) symbols to be coded

We exploit this property and propose a context-based coding method to classify the symbols depending on the preceding coefficients. Two entropy coding methods, arithmetic coding and Huffman coding, are used for the proposed method.

2.1 Context-based coding

We define a symbol s that denotes (run-level, end) as follows.

$$s = f(z, b, e) \tag{1}$$

where f is the mapping function, and z, b and e are zero-run, level and end, respectively. For example, in the case of the first symbol of Fig. 3 (a), z=1, b=2 and e=0, respectively. We define a symbol s_i whose position is located at the i-th position in the zigzag scan order. We call this one "i-th symbol" hereafter. The position of symbol means the starting position of the zero-run or the position of the non-zero coefficient in the case where zero-run is 0.

When the level of the i-th coefficient in the zigzag scan order is defined as a_i, a context for the i-th symbol is defined as follows.

$$c_i = a_0 a_1 \ldots\ldots a_{i-1} \quad (2)$$

where a_0 is always 0.

The statistics for the i-th symbols vary according to the context c_i. In general, the context represents all the property stated before. In the case where the block has large quantization parameter, rapid motion or luminance, there would be many coefficients with large value in the context. When the i-th symbol s_i is conditional on the context c_i, the conditional entropy is denoted as follows.

$$H(s_i \mid c_i) = -\sum_{c_i=0}^{M_i} p(c_i) \sum_{s_i=0}^{N} p(s_i \mid c_i) \log_2 (p(s_i \mid c_i)) \quad (3)$$

where M_i and N are the maximum values of the context c_i and symbol, respectively.

To minimize the code length $-\log_2 (p(s_i \mid c_i))$ of symbol s_i, it is necessary to maximize the conditional probability $p(s_i \mid c_i)$. The conditional entropy of the symbol s is denoted as follows.

$$H(s \mid c) = \sum_{i=1}^{64} \alpha_i H(s_i \mid c_i) \quad (4)$$

where α_i is the probability that a certain symbol is in the i-th position.

The goal is to minimize the entropy $H(s \mid c)$. From the viewpoint of coding efficiency, classification without overhead always reduces the entropy of the symbol [9].

$$H(s \mid c) \le H(s) \quad (5)$$

where $H(s) = -\sum_{s=0}^{N} p(s) \log_2 p(s)$.

Suppose that the coefficients have the level of A bits. Then, there are 2^{iA} different contexts for each symbol with 64 different position. It is impractical to use the contexts because of the complexity and it is necessary to reduce the number of contexts. The tradeoff is between coding efficiency gain and complexity increase.

In order to reduce the number of contexts, coefficients for the contexts may be given by the two-dimensional closeness to the position of the symbol in the 2-D block. For example, eight coefficients close to the position of the symbol in the 2-D block are used as shown in Fig. 4. When the 1-D position i equals (x,y) in the 2D-block, a context for the i-th symbol is defined as follows.

$$c = c_i = a_{i-1} a_{x-1,y} a_{x-1,y-1} a_{x,y-1} a_{x-2,y} a_{x-1,y-1} a_{x-1,y-2} a_{x,y-2} \quad (6)$$

where $a_{x,y}$ means the coefficient located at the position (x,y) in the block.

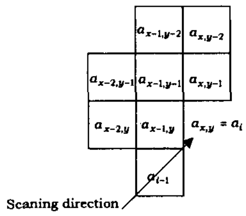


Fig. 4: An example of context

Using the reduced context c in Eqs. (6), we do not have to discriminate the position of the symbol since the context also includes the meaning of the position. The context of the top-left position in the block has more coefficients with large value than that of the bottom-right position. Hence, the context c in Eqs. (6) well represents the symbol statistics changing by position in the block as well as block-by-block. The conditional entropy of the symbol in Eqs. (4) can be denoted as follows.

$$H(s \mid c) = -\sum_{c=0}^{M} p(c) \sum_{s=0}^{N} p(s \mid c) \log_2 (p(s \mid c)) \quad (7)$$

where M is the maximum value of the context.

Since there are not some coefficients for the context in the border or top-left corner of the block, the coefficients have to be carefully manipulated. The number of contexts in Eqs. (6) is still too many to implement. Large number of contexts incurs high complexity on the memory space to store VLC table or probability model. In the case of adaptive arithmetic coding, too many contexts also make it difficult to estimate conditional probabilities by frequency counts within a single image. This is known as "context dilution" problem [7]. In the case of low bit-rate, this problem might be more serious since there are not enough symbols in a single image to estimate the statistics of the source symbols to be coded. In order to reduce the number of contexts, we have to quantize the contexts to be smaller in number.

In this study, a simple context is used to reduce the number of contexts. The level of immediately preceding non-zero coefficient is used for the context of the proposed method i.e. $c_i = a_{i-1}$. And the level is quantized to four values. When the level is greater than 3, context value is chosen to be 4.

2.2 Context-based arithmetic coding

Two arithmetic coding, adaptive and fixed, methods are used for the proposed context-based arithmetic coding. In general, even though adaptive arithmetic coding gives the best performance, it has disadvantages such as error resilience problem and complexity increase for probability model update.

For the fast estimation of the probability model in the adaptive arithmetic coding, we propose n-ary arithmetic coding with multiple probability models. For the symbol s, non-negative values of x_k are encoded using k-th probability model of size n for each context. x_k is defined as follows.

$$x_k = \min\{s - k(n-1), (k+1)(n-1)\}, \quad k = 0,1,\ldots\ldots k_{\max} \quad (8)$$

where min is the minimum operator.

Fig. 5 shows the probability model whose $P_k(x_k)$ means the frequency of x_k. We use the same method as [10] for probability model update. All $P_k(x_k)$ of the model are initialized by 1 at the start of the frame and whenever x_k appears, $P_k(x_k)$ is increased by 1. Supposing that x_k appears in the k-th model, the probability that x_k equals m is as follows.

$$\Pr\{x_k = m\} = \frac{P_k(m)}{\sum_{x_k=0}^{n-1} P_k(x_k)} \quad (9)$$

For example, suppose that n and s are 8 and 17, respectively. Then, three values of x_k, $x_0 = 7$, $x_1 = 7$ and $x_2 = 3$, are encoded by arithmetic coding according to their probability models. $x_0 = 7$ and $x_1 = 7$ mean that the symbol s is larger than 6 and larger than 13, respectively.

The size n should be chosen taking into account the probability distribution of the symbols. By mapping the higher probable symbol to the lower value, this method gives fast probability estimation of the highly probable symbols in the adaptive arithmetic coding. On the other hand, in the case of the fixed arithmetic coding, this method gives the same results as multi symbol arithmetic coding with single probability model used in H.263 오류! 참조 원본을 찾을 수 없습니다.].

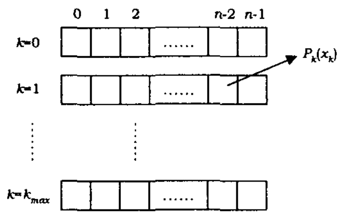


Fig. 5: Probability model

2.3 Context-based coding with adaptive VLC table selection

The different symbol probability models for the fixed arithmetic coding can be treated by using the same number of VLC table. Adaptive VLC table selection method would give little lower gain than the fixed arithmetic coding. In this section, we propose a context-based VLC table selection method, which does not increase complexity from current standards.

Most of current standards use two VLC tables, intra and inter ones, for intra and inter coded block, respectively. However, in the case of the block having fast and irregular motions, the motion compensated signals behave rather like the intra-coded ones as shown in Fig. 2 (b) and the inter coded blocks produce fewer bits with the intra VLC table. It is also observed that the same property is presented for the blocks having many coefficients with large level.

Taking into account the property, we propose a context-based inter/intra VLC table selection method for inter coded macro block. According to the context stated in the previous section, one of the two VLC tables is chosen for each symbol, where intra VLC table is selected if context value is greater than 2.

3 Experimental results

In order to evaluate the performance of the proposed context-based coding, experiments were carried out under the H.263 and MPEG-4 standards. Any optional modes were not used in H.263 codec and only basic tools were used in MPEG-4 codec.

Four image sequences, Flower garden and Foot ball in SIF format, and News and Akiyo in CIF format, were used as shown in Fig. 6. Flower garden sequence has fast translation motion, and Foot ball sequence has fast irregular motion. Akiyo is a head and shoulder image sequence having slow motion, and News sequence has both fast and slow motion. The first and the last two image sequences are 5s and 10s in time, respectively. Each sequence was coded with 10 frames/s. Only the first frame was coded as an intra frame and it was not used for performance comparison. The total number of bits generated by the standards, H.263 and MPEG-4, with and without the proposed methods were compared as proposed and conventional methods, respectively. The proposed method was not applied for intra macro-block inside an inter frame, but the bits from the intra macro-block were included in the total number of bits. In Fig. 8 and Fig. 10, Gain (%) means percentage of bits decrease in the proposed method over conventional method as below.

$$Gain(\%) = \frac{(N_{conv} - N_{prop}) \times 100}{N_{conv}}$$

where N_{prop} and N_{conv} are the number of bits from the proposed and the conventional method, respectively.

3.1 Comparison of classification methods

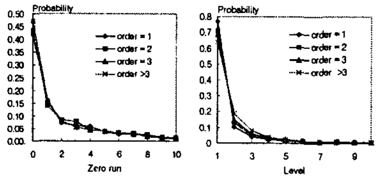
The probability distributions of zero-run and level were compared according to the classification methods for news sequence having both fast and slow motion as shown in Fig. 7. The zero-run and level are two elements of the symbol (run-level, end) and they take up the major part of the bit-rate for the symbols. The level means absolute value of a non-zero coefficient and zero-run denote the number of zero coefficients until the non-zero coefficient.

The proposed method and two conventional methods, by order of symbol [1] and position of symbol [3,4,5], were compared. The order of symbol means the coding order of a symbol in the block. The position of symbol means the starting position of the zero-run or the position of the non-zero coefficient in the case where zero-run is 0.

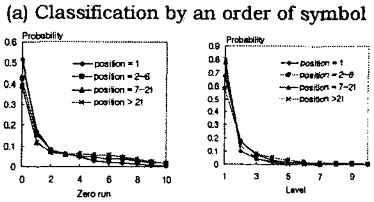
The Fig. 7 shows that the proposed context-based method gives the best classification. Classification method using the order of symbol is the worst.

In Fig. 7 (b), larger position of symbol leads to lower level and longer zero-run as we expected. However, the amplitudes of the DCT coefficients for inter block are more uniformly spread than intra block. Also, the distribution is slightly different from that of motion compensated prediction error investigated by [8] since it is conditional probability having non-zero coefficient before the symbol being coded. For example, the probability that the level is 1 at the position 0 is lower than that at the position 1~5.

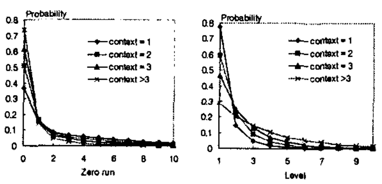
The experimental results show that the proposed method better classifies the symbols according to their statistics than the classification based on the position of the coefficients in the block. And the similar results are presented for the other sequences.



(a) Classification by an order of symbol



(b) Classification by starting position of the zero-run



(c) Classification by context

Fig. 7: Probability distribution of zero-run and level according to the classification methods

3.2 Arithmetic coding

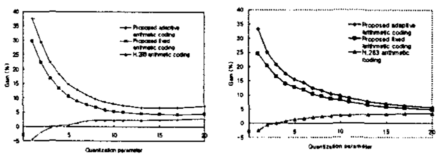
Probability models for fixed arithmetic coding were generated by training the coefficient statistics from the CIF resolution image sequences - table tennis and containership. The arithmetic coding is based on 오류! 참조 원본을 찾을 수 없습니다.]. Five context values 0~4 are used, where 0 is the case of the first symbol in the block, and the other ones are the quantized level of the previous non-zero coefficient.

The coding performance gains of the proposed methods and H.263 arithmetic coding over H.263 VLC coding were plotted in Fig. 8. Since the coding techniques are based on entropy coding, the PSNR is the same for each.

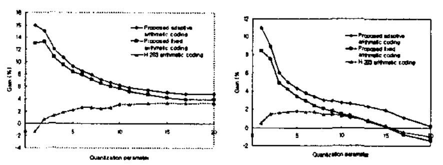
The plot shows that the performance of the two proposed methods are better than that of the H.263 arithmetic and VLC coding except for the case with quantization parameter over 15 in akiyo sequence. The gain is higher for smaller quantizaton parameter and the sequences with fast and complex motion.

The gain of the fixed arithmetic coding is lower than the adaptive one, but the tendency is very similar. It means that symbol statistics are efficiently classified according to context regardless of the other properties such as quantization parameter and motion.

Fig. 9 shows the PSNR as a function of the coding bit-rate for the four image sequences. The proposed method gives PSNR improvement up to 2.7 dB.



(a) Flower garden sequence (b) Football sequence



(b) News sequence (d) Akiyo sequence

Fig. 8: Gain of the proposed context based arithmetic coding method over the H.263

3.3 Adaptive VLC table selection method

The coding performance gains of the proposed method over the MPEG-4 codec are plotted in Fig. 10. The plot shows that the proposed method is always better than the conventional method in terms of coding efficiency. The gain is also higher for smaller quantizaton parameter and the sequences with fast and complex motion.

Fig. 11 shows the PSNR versus bit-rate for the four image sequences. The proposed method gives

PSNR improvement up to 0.8 dB.

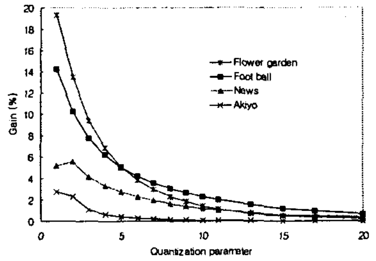


Fig. 10: Gain of the proposed context based VLC table selection method over the MPEG-4

4 Conclusion

In this paper, context-based coding methods are proposed for variable length coding of 2D-DCT coefficients.

The results of our experiment show that coding efficiency gain of the proposed method is larger for smaller quantizaton parameter and the sequences with fast and complex motion. Therefore, for high quality video coding, the proposed method is much more advantageous than the conventional method.

The proposed context-based coding with adaptive inter/intra VLC table selection gives bits gain up to 19% and PSNR improvement up to 0.8 dB. This method could be easily implemented without complexity increase from the current standards.

The proposed context-based arithmetic coding method gives bits gain up to 37% and PSNR improvement up to 2.7dB. The fixed arithmetic coding gives a little lower gain than the adaptive one, but has a similar tendency in coding efficiency irrespective of bit-rate. It shows that the proposed context-based coding method is close to optimum regardless of bit-rate under the assumption that the adaptive arithmetic coding is optimum.

The proposed method can be extended to the cases using more than one coefficient for context to obtain more coding gain in the fixed arithmetic coding and adaptive VLC table selection method. However, in the adaptive arithmetic coding, they do not give more coding gain for low bit-rate cases and may give worse ones because of the context dilution problem.

In order to efficiently transmit video data in the future high bit-rate network, the proposed method is promising for the newly made standards, H.263 and MPEG-4, which have been optimized for low bit-rate

References

[1] ITU-T H.263, "Draft recommendation H.263: video coding for low bitrate communication," Oct. 1995.

[2] ISO/IEC 14496-2 Final Draft of International Standard, "Information technology-generic coding of audio-visual objects", Dec. 1998.

[3] E. Reed and J. Lim, "Efficient coding of DCT coefficients by joint position-dependent coding," *Proc. ICSSP'98*, Vol. 5, pp. 2817-2820, May 1998.

[4] F. Hartung and B. Girod, "Improved encoding of DCT coefficients for low bit-rate video coding using multiple VLC tables" *Proc. ICIP'99*, Vol. 2, pp. 51-55, 1999.

[5] J.C. Jeong and J.M Jo, "Adaptive Huffman coding of 2-D DCT coefficients for image sequence compression," *Signal Processing: Image Communication*, 7:1-11, 1995.

[6] B.W. Jeon, J.H. Park, and J.C. Jeong, "Huffman coding of DCT coefficients using dynamic codeword assignment and adaptive code book selection," *Signal Processing: Image Communication*, 12:253-262, 1998.

[7] M. Weinberger, J. Rissanen, and R. Arps, "Application of universal context modeling to lossless compression of gray-scale image", *IEEE Trans. Image Processing*, vol. 5, no. 4, pp. 575-586, 1996.

[8] F. Bellifemine, A. Capellino, A. Chimienti, R. Picco, and R. Ponti, "Statistical analysis of the 2D-DCT coefficients of the differential signal for images," *Signal Processing: Image Communication*, 4:477-488, 1992.

[9] Richard E. Blahut, *Principles and practice of information theory*, Reading, Massachusetts: Addison-Wesley Publishing Company, 1987.

[10] I. H. Written, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, no. 6, pp. 520-540, 1987.