Partially Decodable and Reversible Variable Length Code for Efficient Image Transmission

Susumu Nishida†, Guo Muling†, Madoka Hasegawa†† and Shigeo Kato††

† Graduate School of Engineering, Utsunomiya University

†† Faculty of Engineering, Utsunomiya University

Kato Lab., Yoto 7-1-2, Utsunomiya-shi, Tochigi 321-8585, Japan

Tel/Fax: +81-28-689-6256

E-mail : susumu@mclaren.is.utsunomiya-u.ac.jp

Abstract: Variable length codes are often used in entropy coding, but are very vulnerable in noisy environments. Reversible variable length codes, however, make possible to decode instantaneously in both forward and backward directions, so that more usable data can be retrieved when bit errors occur via transmission. Furthermore, partial decodability is desirable to introduce in the reversible variable length code because ROI (Region Of Interest) decoding function is sometimes required in recent image information systems such as the medical imaging, the digital museum and so on. In this paper, we propose a partially decodable and reversible variable length code by modifying Golomb-Rice code.

1. Introduction

Variable length codes are well known as an efficient code for data compression, but are very vulnerable in noisy environments because of bit errors and synchronization losses. Reversible variable length codes (RVLCs) are, however, instantaneous decodable code in both forward and backward directions. Therefore, more usable data can be retrieved by reverse decoding when transmission errors occur. Furthermore, bit errors can be detected by comparing the data stream decoded in both directions. Compared to forward error correction (FEC), no or less redundancy is imposed on code words of RVLCs [1][2].

An effective RVLC is proposed by Wen and Villasenor[3] by modifying the unary part of Golomb-Rice code[4]. Though Wen's Reversible Golomb-Rice (W-RGR) code has the same code word length distribution as Golomb-Rice code, but bit errors in binary part can not be detected.

Meanwhile, requirements for image coding are summarized in JPEG 2000 profiles. One of the requirements is ROI (Region Of Interest) decoding. In order to decode the ROI, it is needed to decode only code words corresponding to the region where one wishes to decode and display, without decoding from the beginning of the code sequence. Hence, it is desirable to develop an efficient partially decodable code.

In this paper, we propose a partially decodable and reversible variable length code by adding parity check bits alternatively as the unary part of Golomb-Rice code. We call this code Parity-check based Reversible Golomb-Rice code (P-RGR code). P-RGR code also has the same code word length distribution as Golomb-Rice code, and can detect one-bit error in any position of the encoded bit stream.

2. Wen's RGR code and its characteristics

Golomb code has been known as an efficient code for exponentially distributed source. When the order of Golomb code is given by a power of 2, the code is called Golomb-Rice code (GR code). Wen's RGR code is constructed by replacing the unary part of GR code words by the following rule.

CASE 1: In the code word of 1 bit length in the unary part, the terminating bit of the code word is set to a bit '0'.

CASE 2: In the code word of the length over 2 bits in the unary part, the starting bit and the terminating bit of the unary part are set to bit '1'. All other bits of the unary part are set to a bit '0'.

Binary parts of W-RGR code words are exactly same as the corresponding ones of GR code words. Examples of W-RGR codes of the order m=4 and 8 are shown in Table 1. W-RGR code can be easily decoded bi-directionally because the unary parts are forward-and-backward symmetry, and non-symmetrical binary parts are of the fixed length. However, the minimum Hamming distance of W-RGR code words is 1, so that the transmission errors caused in the binary part cannot be detected.

Table 1. Examples of W-RGR code and P-RGR code of order m = 2 and 4.

	W-RGR code		P-RGR code	
n	m=2	m=4	m=2	m=4
0	00	000	01	000
1	10	010	10	011
2	011	100	000	101
3	111	110	111	110
4	0101	0011	0011	0010
5	1101	0111	1100	0100
6	01001	1011	00100	1001
7	11001	1111	11011	1111
8	010001	00101	001011	00111
9	110001	01101	110100	01010
10	0100001	10101	0010100	10001
11	1100001	11101	1101011	11100

3. Proposal of P-RGR code using parity check functions

3.1 Construction method of P-RGR code

The P-RGR code of order m for symbol n is constructed in the following two steps.

Step1: Construct a binary part

The binary part of P-RGR code word is same as that of the corresponding GR code word. It is said that the binary part is represented by binary number n modulo m of $k = \lfloor \log_2 m \rfloor$ bits.

Step2: Construct a unary part

The unary part of the length $\lfloor n/m+1 \rfloor$ bits is constructed by progressively adding the parity check bits of their previous $k(=\lfloor \log_2 m \rfloor)$ bits. To construct parity check bits for the beginning k bits of the unary part, concatenate the binary part before the unary part. The polarity of the parity is selected as follows.

(1) In the case that the length of the unary part is only 1 bit, the unary bit is set by even (odd) parity of the previous $k(=\lfloor \log_2 m \rfloor)$ bits, i.e. even (odd) parity of the binary part.

(2) In the case except above, the beginning and the ending bits of the unary part are set by odd (even) parity of their previous $k(=\lfloor \log_2 m \rfloor)$ bits and the remaining bits of the unary part are set by even (odd) parity of their previous $k(=\lfloor \log_2 m \rfloor)$ bits.

For example, in case of n=9 and m=4, the binary part is '01', P-RGR code word is constructed such as the following Fig.1.

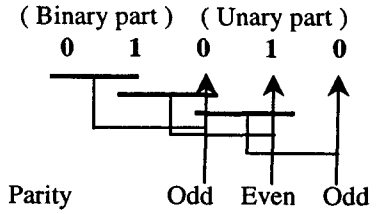

Figure 1. An example of P-RGR code word (n=9,m=4)

3.2 Bi-directionally decoding

The P-RGR code word set for the forward decoding and the backward decoding is exactly same except the code word assignments for symbols. This fact indicates that the backward decoding process can be constructed by symmetrical or similar manner as the forward decoding.

In the backward decoding, the end of the code word can be detected in the same manner as the forward decoding, because the parity rule described in the step-2 of chapter 3.1 is also established. Fig. 2 and Fig. 3 show block diagrams of the forward and backward decoding, respectively.

When the code parameter k is given, a code word can be decoded by the following steps.

Step1: Read k bits from the bit stream in forward or backward direction for each case. These k bits are interpreted as the binary part in the forward decoding, but they are only skipped in the backward decoding. Set a counter C for counting the length of unary part of P-RGR code.

Step2: Compare the bit in the code stream to the parity check bit of the previous k bits;

(1) If the beginning bit of the unary part is the even parity of the previous k bits, we know that the end of the code word is encountered and go to step3.

(2) Otherwise, compare the next bit in the code stream to the parity check bit of the previous k bits. This procedure is successively executed and increases the counter C by one until the next odd parity is encountered.

Step3: Combine the binary and unary part to reconstruct the symbol. For forward decoding, add the decimal value converted from the k bits of the binary part to the multiple of C and 2^k. For backward decoding, read k bits from the end of the code word by *bit-reversal*, and this forms the binary part. Add the decimal value converted from the k bits of the binary part to the multiple of C and 2^k.

From above discussions, we can see that the coding and decoding in both directions can be implemented mainly by a parity check circuit, a shift register, a counter and a binary to decimal converter.

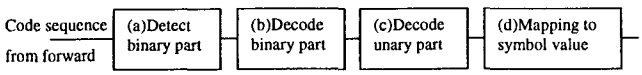

Figure 2. The block diagram of the forward decoding

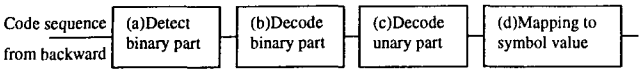

Figure 3. The block diagram of the backward decoding

3.3 The method of partially decoding

One of the methods to make the code stream partially decodable is to use the specified bit patterns as the marker code. In case of P-RGR code with the order m, the code word corresponding to symbol $2m$-1 is constructed by '1' run of k+2 bits. Therefore, unless the code word of all 1 pattern such like "111...1" is used for any symbol, any 1 run of the code stream is limited in the finite length. The '1' run with the maximum length occurs when the code word with the suffix of bit '1' of the k+1 bits length is concatenated to the code word with the prefix of bit '1' of k+1 bits length. Thus, the maximum length of 1 run is $2k$+2 bits. Because '1' run over $2k$+3 bits never exist in any code stream, '1' run over $2k$+3 bits can be used as the marker code for detecting the ROI. Figure 4 shows the marker code composed by '1' run of $2k$+3 bits.

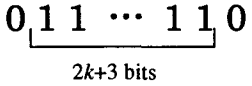

Figure 4. Marker code

When bit error occurs in the marker code, the beginning position of the marker code cannot be specified. To cope with the bit error in the marker code, the new marker code must include '1' run over $4k$+6 bits as shown in Fig.5. By using of this marker code, even if one bit error occurs in data stream, the partially decodable property can be kept available.

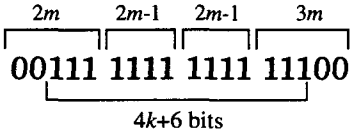

Figure 5. Marker code for considering a bit error

3.4 Comparisons of error detection abilities between W-RGR code and P-RGR code

Because Hamming distance between two code words increases by adding a parity check bit in the each codeword, the Hamming distance between any two code words of the same length in P-RGR code is at least 2. Therefore, if one and only one bit transmission error occurs in data block, the erroneous code is decoded to a code word of different length, and makes the data decoded in the forward direction differ from those decoded in the backward direction. Thus, the existence of error can be detected.

On the contrary, though the error caused in the unary part of the code word can be detected in W-RGR code, the error in the binary part is never detected because the code word can be decoded into the different code word of the same length.

Fig. 6 and 7 show examples of the decoding process of code stream in case that one bit error occurs in the binary part of the code stream.

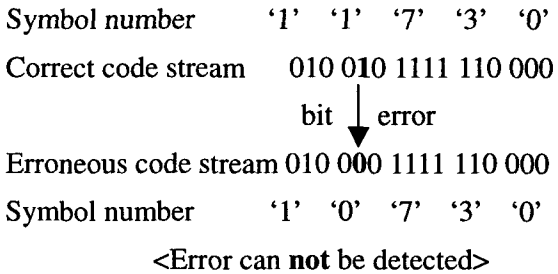

Figure 6. The decoding process with W-RGR code

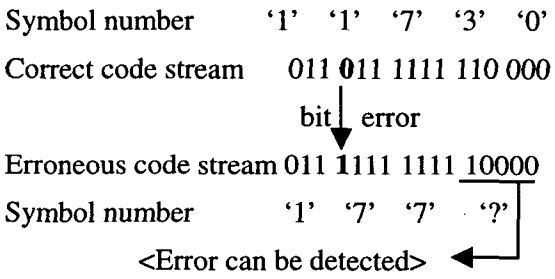

Figure 7. The decoding process with P-RGR code

The occupation ratio of the binary part in a code word increases as the length of the code word decreases. In other words, when the transmission error occurs in the short length code word, the error occurrence probability in the binary part is rather high. Because short length code words are assigned to frequent occurrence symbols, the occurrence probabilities of the short length code word are high in the code sequence. As the result, the detection probability of the error may lowers in the W-RGR code.

4. Simulation results and considerations

After dividing image "auto" (512 × 512 pixels, 8 bit/pel) with blocks of 512×512 pixels and 128×128

pixels, one of the lossless wavelet transforms is applied and encoded by proposed P-RGR code. The marker code is inserted between each block and each band, and one bit error is artificially introduced in the lowest frequency band.

Decoded images of the lowest frequency band corresponding to images decomposed into blocks of 512×512 pixels and 128×128 pixels are shown in Fig.8 and Fig.9, respectively.

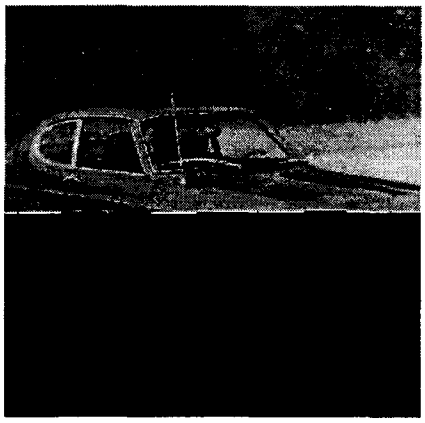

Figure 8. Decoded image 1 (block size 512×512)

Figure 9. Decoded image 2 (block size 128×128)

Fig. 8 shows that decoded data after error occurred are never recovered into correct data, whereas Fig. 9 shows that the re-synchronization is done at the continued block, and correct data can be decoded.

On the other hand, Fig. 10 shows the result of data recovery simulation.

In case that one bit error occurs in data stream, the synchronization of code words is lost. As a result, symbol "$2m-1$", which is not used in coding process, occurs. If symbol "$2m-1$" can be detected, more usable data can be retrieved by stopping the forward decoding and decoding from the backward direction.

Figure 10. Data recovery image for Fig.1 (block size 512×512)

5. Conclusions

We propose a partially decodable and reversible variable length code based on the parity check rule. The proposed P-RGR code can detect one-bit error in any position of the code stream and has the same code structure in both directions. In our future works, we will report the method for introducing the bi-directional decoding to the adaptive coding algorithm such as changing the code parameter k according to the characteristics of the image.

References

[1] Y. Takishima, M. Wada and H. Murakami, "Reversible Variable Length Codes," IEEE Trans. Comm., vol.43, No.2/3/4, pp.158-162, 1995.

[2] B. Girod, "Bi-directionally Decodable Streams of Prefix Code Words," IEEE Communications Letters, vol. 3 no. 8, pp. 245-247, August 1999.

[3] J. Wen and J. D. Villasenor, "Reversible Variable Length Codes for Efficient and Robust Image and Video Coding," Proceedings of the 1998 IEEE Data Compression Conference, pp.471-480. Snowbird, Utah, 1998.

[4] R. F. Rice, "Some practical universal noiseless coding techniques," Tech. Rep. JPL-79-22, March 1979.