ERROR RESILIENT MULTIPLEXED CODING WITH RVLC

Xiaosong Wang and Tokunbo Ogunfunmi

Department of Electrical Engineering, Santa Clara University
Santa Clara, CA 95053, USA
email: xwang1@.scu.edu, togunfunmi@.scu.edu
web: http://www.scu.edu/engineering

ABSTRACT

A new class of error resilient source code – Multiplexed Code was recently introduced. It takes advantage of the fact that real multimedia signals, such as images, audios and videos, contain heterogeneous information that can be grouped at different priority levels. High priority information is assigned to fixed length codewords; the inherent redundancy is exploited to represent low priority information.

In this paper, we introduce an enhanced version – MultiRVLC. It uses Multiplexed Codes with Reversible Variable Length Coding (RVLC) to achieve further error resilient improvement. For example, PSNR is increased by 3.4dB for still images at 5e-3 bit error rate.

1. INTRODUCTION

Multiplexed Code was proposed by Herve Jegou and Christine Guillemot in 2003 [1]. It attracts more and more attention because it overcomes the major drawback of variable length code (VLC) – de-synchronization. The raw real signals are decomposed into different priority levels, which can be related to signal frequency sub-bands, bit planes, quantization layers, etc. The relatively small but very important part, the high priority source $\mathbf{S_H}$ is encoded into fixed length codewords (FLCs). The inherent redundancy is then exploited to represent the low priority source $\mathbf{S_L}$. Since it is protected by FLCs, $\mathbf{S_H}$ is immune to de-synchronization, and also allows random access to the data stream.

It has been shown that Multiplexed Code achieves great error resilience at almost no cost of compression efficiency [1]. This paper demonstrates an enhanced scheme that exhibits further error resilience improvement, which applies reversible variable length code (RVLC) to low priority information instead of VLC. The outline of this paper is listed as follows. Section 2 describes the coding algorithm briefly. Section 3 exhibits the simulation results on coding efficiency, error resilient performance etc. Section 4 draws a conclusion on this paper. Section 5 suggests some future work.

2. ALGORITHMS

Here, we describe the algorithms: Mutiplexed Code and MultiRVLC.

2.1 Multiplexed Code

Multiplexed Code is derived from the idea of unequal error protection (UEP). That is, more protection or resource is allocated to the more important information in order to achieve better quality at the receiver end.

Two sources are considered in Multiplexed Code [2]: a high priority source $\mathbf{S_H}$ and a low priority source $\mathbf{S_L}$. They take values from finite alphabets $\mathbf{\Psi}$ and $\mathbf{\Psi'}$ respectively. The high priority sequence $\mathbf{S_H}$ is protected from the de-synchronization effect by applying a fixed *c*-bit length codeword to each symbol. Hence, total of $N = 2^c$ codewords need to be partitioned into Ω subsets, called *equivalence classes*, which are corresponding to Ω symbols of the alphabet $\mathbf{\Psi}$. Each subset $\mathbf{C_i}$ contains a set of N_i codewords $\{C_{i,1}, C_{,i2}, ..., C_{i,3}, ..., C_{i,N_i}\}$, such that $N = \sum_1^{\Omega} N_i$. Now each symbol $S_k = a_i$ of the high priority sequence $\mathbf{S_H}$ can be converted into a *c*-bit codeword $C_{i,q}$ belonging to subset $\mathbf{C_i}$. In other words, a_i is mapped into a pair of variables $\{\mathbf{C_i}, q\}$, in which q is the index of the codeword in subset $\mathbf{C_i}$, and q takes its value from 1 to N_i. This mapping relationship can also be defined as $C_{i,q} \Leftrightarrow (a_i, q)$ [2]. Obviously, mapping the sequence $\mathbf{S_H}$ into a sequence of *c*-bit codewords creates an additional sequence of indices **q**, which can be used to represent the low priority source $\mathbf{S_L}$. An example of Multiplexed Codes is shown in Table 1.

Table 1. An example of Multiplexed Codes (*c*=3)

Equi. Class **Ci**	Code-word $C_{i,q}$	Sym-bol a_i	Card(**Ci**) N_i	Probp_i	Index q_i
$\mathbf{C_1}$	000 001 010	a_1	3	0.38	0 1 2
$\mathbf{C_2}$	011 100	a_2	2	0.21	0 1
$\mathbf{C_3}$	101	a_3	1	0.16	0
$\mathbf{C_4}$	110 111	a_4	2	0.25	0 1

A high priority source $\mathbf{S_H}$ takes values from the 4-symbol alphabet $\mathbf{\Psi}=\{a_1,a_2,a_3,a_4\}$. Its *equivalent classes* **Ci**, i=1,2,3,4 are generated based on the stationary probability of $\mathbf{\Psi}$ as shown in Table 1. For the given high priority sequence $\mathbf{S_H}$={ a_2, a_1, a_1, a_4, a_2, a_3}, Λ=2·3·3·2·2·1=72. Hence, there are $\lfloor \log_2(\Lambda) \rfloor$=6 bits can be taken from the low priority bit-stream **b**="110101", resulting in the intermediate $\gamma = \sum_1^6 b_r \cdot 2^{r-1}$ =53. Consequently, the Euclidean decomposition results in $\boldsymbol{q}$=(1,2,2,0,1,0) since $(n_1, n_2, \cdots, n_{K_H-1}, n_{K_H})$ =(2,3,3,2,2,1),and

$$\gamma = q_1 + n_1(q_2 + n_2(\cdots(q_{K_H-1} + n_{K_H-1} \cdot q_{K_H})\cdots)).$$

Therefore, the multiplexed codes are 100 010 010 110 100 101.

2.2 MultiRVLC

In original Multiplexed Code, $\mathbf{S_L}$ is pre-encoded into bit-stream **b** using VLC, e.g. Huffman Code. The difference made for MultiRVLC, the enhanced version, is to pre-encode $\mathbf{S_L}$ into RVLCs. This action may not help much for the low priority information when the bit error rate (BER) is relatively high. Surprisingly, however, the reason is the dramatic fact of error resilience enhancement resides at the high priority part as proven by the simulation results shown in the following section.

3. SIMULATION RESULTS

Image 'lena', a real source, is used to evaluate both Multiplexed Code and MultiRVLC. Two-stage wavelet decomposition in Figure 1 is applied to the image to generate the high priority source – coefficients of low frequency subbands LL2, LH2, and low priority source – coefficients of high frequency subbands HL2, HH2, LH1, HL1, and HH1. For the 256*256 format 'lena' image, the quantization levels are set to 64 and 8 for high priority source and low priority source respectively. For high priority source, the fixed codeword length is set to 10.

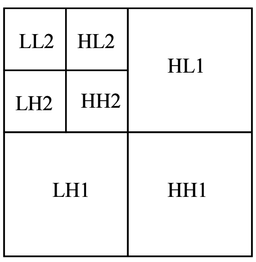

Figure 1. Two-stage Wavelet decomposition

Table 2 shows two sets of codewords being applied to $\mathbf{S_L}$. One set is for original Multiplexed Code, in which $\mathbf{S_L}$ is pre-encoded into bitstream using Huffman Code; the other set is for MultiRVLC, in which $\mathbf{S_L}$ is pre-encoded using asymmetric RVLC.

Table 2. Codeword sets for $\mathbf{S_L}$ (256*256 image)

Quantized Coeff.	Huffman Code	Asymmetrical RVLC
0	1	0
1	0110101	1000001
2	011011	100001
3	0111	1001
4	010	101
5	00	11
6	01100	10001
7	0110100	10000001

Fixed length code (FLC), Huffman code, and RVLC are also applied to the same image source as references to coding efficiency and PSNR performance. Huffman codes are implemented with resynchronization markers inserted. Asymmetric RVLCs are created according to the algorithms in [3, 4, 5].

3.1 Coding Efficiency

The coding efficiency is measured by '*bpp*', the average number of bits used per pixel. The results are shown in Table 3. For different size images, the quantization levels are adjusted accordingly in order to reduce the required bits per pixel to a reasonable amount. As seen from Table 2, Multiplexed Code is more efficient than Huffman code and RVLC. This is because Huffman Code and RVLC still retain some redundancy by using only integer size of codeword to represent each symbol, while this redundancy can be exploited by Multiplexed Code to some extent.

Besides, the coding efficiency is increasing with the image sizes. That can be stated by the increased inter-pixel correlation or the redundancy of the image when image size is larger.

Table 3. Coding efficiency at different image formats

Image Size/ PSNR	128*128/ 32.03dB	256*256/ 34.08dB	512*512/ 34.87dB
FLC	3.9502	3.7386	4.6014
Huffman	2.2048	1.8748	1.6807
RVLC	2.2426	1.9067	1.7593
Multiplex	2.0615	1.8024	1.6701
MultiRVLC	2.0616	1.8027	1.6702

As for MultiRVLC, it is noticeable that after applying RVLC to the low priority source, the coding efficiency is almost the same as that of the original Multiplexed Code. That can be clarified by the fact that the quantizaiton levels used for low priority source are fewer - 8 for 128*128, 256*256 image, and 16 for 512*512 image. Therefore, according to the construction algorithm of asymmetrical RVLC described in [3], the resulted asymmetric RVLCs are nearly as efficient as the original Huffman codes. As shown in Table 1, for 8 quantization levels, only one codeword of RVLCs is longer than the corresponding codeword of Huffman Codes, with only one-bit. Furthermore, this codeword has the smallest probability. That is to say, it will not cause much degradation on the overall coding efficiency, as confirmed by the results shown above.

3.2 Error resilient performance

Error resilient performance is a focus of attention in these coding schemes. The simulations are performed assuming a binary symmetrical channel with varying BERs. PSNR performance is used as a criterion to measure the image quality at the decoder side. The 256*256 'lena' image is the subject for all five coding methods: FLC (3.7368 *bpp*), Huffman Coding (1.8748 *bpp*), RVLC (1.9067 *bpp*), Multiplexed Coding (1.8024 *bpp*), and MultiRVLC (1.8027 *bpp*).

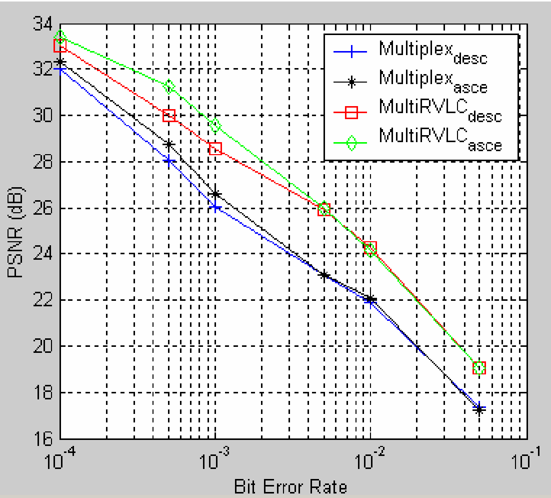

Figure 2. PSNR performance improved by arranging the low priority source in the ascending order of importance

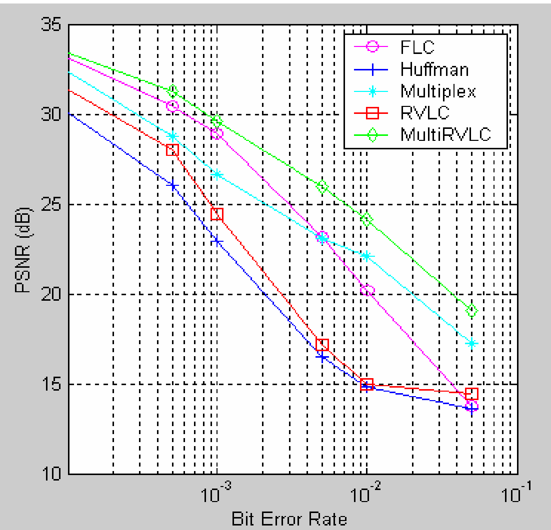

Figure 3. PSNR performance vs. BER

For Multiplexed Codes and MultiRVLCs, it is worth mentioning a notable action taken at the encoder side for the low priority source. That is, the subbands are arranged in the ascending order of priority. As a result, the subband HH1, which has the lowest priority, is put at the beginning of the bitstream **b**; the subband HL2, which has the highest priority, is put at the end of the bitstream **b**. As there is always a portion of bitstream **b** left intact, the most valuable part of the low priority bitstream, subband HL2, is most possible to be attached at the end of multiplexed flow without being converted. Such arrangement prevents the most valuable subbands of the low priority source from additional damage that can occur in the multiplexed flow. Figure 2 shows the performance improvement at relatively low BERs compared with the arrangement in descending priority order for both Multiplexed Code and MultiRVLC.

The results of error resilient performance for the five coding methods are shown in Figure 3. As expected, at very high BER, e.g., 5e-2, classical Huffman codes are totally corrupted. The image quality is generally not acceptable with PSNR less than 15 dB. Even RVLC does not show any improvement at this high BER. FLC shows better PSNR at low BER (<5e-3) than Multiplexed Code. That is because FLC evenly protects heterogeneous sources. At low BER, the image quality benefits from the fact that more bits are used for each pixel - 3.7368 *bpp* for FLC, compared with only 1.8024 *bpp* for Multiplexed Code. However, at very high BER, Multiplexed Code shows great error resilience improvement than FLC. That is the evidence of the advantage resulted from unequal error protection technique. High priority source, which is more important for the image quality, is protected with 10 bits now. As for the low priority source, it does not require too many bits since the pre-encoded variable length codes are corrupted badly anyway.

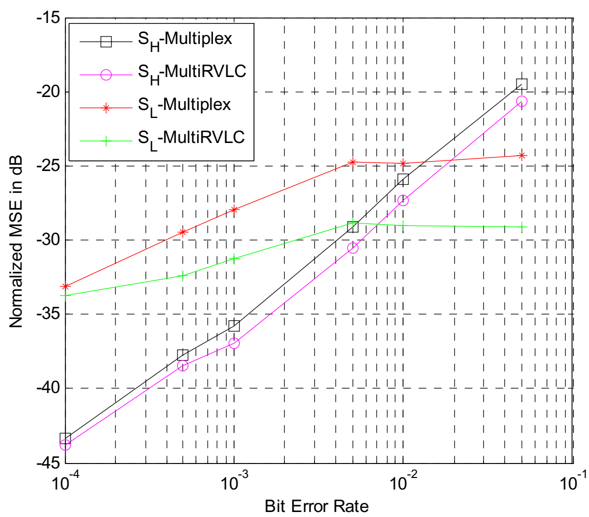

Figure 4. Comparison of Normalized MSEs on S_H and S_L

A significant phenomenon in Figure 3 is that MultiRVLC gives remarkable error resilient improvement over Multiplexed Code even when BER is very high (BER > 1e-2). This contradicts the observations that RVLC does not show much improvement over classical Huffman Code at BER greater than 1e-2. Separate mean square error (MSE) analyses on low priority source and high priority source have been done to pinpoint this contradiction. As shown in Figure 4, by applying RVLC to low priority source, not only the MSE for

S_L is reduced, but also a better MSE is achieved for the high priority source S_H, compared with original Multiplexed Code. That can be explained by revisiting the structure of RVLC codeword in Table 2. The coding schemes, Multiplexed Coding and MultiRVLC Coding convert the VLC (or RVLC) bitstream into a sequence of indices. The structure of RVLC codeword helps make the indices stay in the correct *equivalent classes* when error occurs. Therefore, by multiplexing RVLC onto the fixed length codewords, the error resilience property of the fixed codewords is improved compared with multiplexing the Huffman Codes onto the fixed length codewords. Another noticeable observation in Figure 4 is that the MSE of S_H keeps increasing with BER, while the MSE of S_L stays almost constant when BER > 5e-3. This can be clarified by the VLC/RVLC decoding procedure adopted at the decoder side, i.e., once the loss of synchronization is flagged, the rest of S_L coefficients are forced to 0 until the next resynchronization marker is located. Therefore, when the BER is high enough, the MSE of the decoded low priority information will not fluctuate much because a large portion of it is set to 0. At the same time, this part of information will not provide much improvement to image quality.

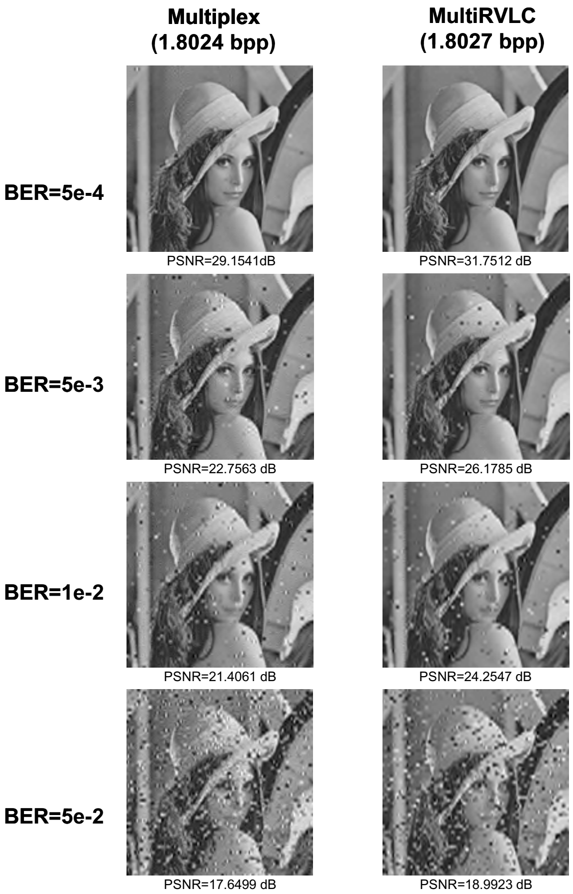

Figure 5. Visual quality comparison

However, it is not the case for S_H, which is encoded using fixed-length codewords. Specific values will be decoded no matter whether there are errors or not.

Visual quality is another criterion for image processing. After being corrupted by the noisy channel with varying BERs, the images are reconstructed as shown in Figure 5. As expected, at very high BER, Multiplexed Code and MultiRVLC exhibit significant error resilience capability through their visual knowledge. As the high priority information is protected by FLC, the error pattern is typically "salt and pepper". The improvement claimed by MultiRVLC is also consistently established by its better visual quality at any level of BER as shown in Figure 5.

4. CONCLUSIONS AND FUTURE WORK

This paper presented MultiRVLC. It is based on Multiplexed Code which shows great error resilient capability at almost no cost of coding efficiency. This coding technique recognizes the fact that real multimedia source generally shows heterogeneous characteristics. Therefore the high priority part can be protected from the common de-synchronization problem of VLC by using fixed length codewords. The consequent redundancy is then exploited to represent the low priority part. MultiRVLC - an enhanced version is proposed in this paper. It achieves remarkable error resilience improvement over the original Multiplexed Code. Compared with traditional Huffman Code and even RVLC, the simulation results demonstrated error resilient capability exhibited by Multiplexed Code and the improvement accomplished by MultiRVLC.

Since multimedia sources such as videos, audios, and images generally exhibit heterogeneous features. It is possible to apply Multiplexed Code and MultiRVLC to other multimedia formats than images. For example, video source is a feasible candidate. Actually, the technique of data partition has been adopted by MPEG-4 standard [6], in which the macroblock data such as DC and AC coefficients, Motion Vectors, Prediction Errors, texture information, etc are separated into high priority and low priority groups. Therefore, it could possibly improve the video quality over error-prone channels, e.g. wireless channel, by applying Multiplexed Code and MultiRVLC to these components.

5. REFERENCES

[1] H. Jegou, and C. Guillemot, "Error-Resilient Binary Multiplexed Codes", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003 (ICASSP '03), Vol. 4, pp 261-264, April 2003

[2] H. Jegou, and C. Guillemot, "A new class of codes for robust compression of heterogeneous data: Multiplexed Codes", www.infra.fr, August 2003,

[3] Y. Takishima, M. Wada and H. Murakami, "Reversible Variable Length Codes", *IEEE Transactions on Communications.*, vol. 43, pp. 158-162, 1995.

[4] Y. Lee, W. chang, H. Ho and C. Lee, "Construction of Error Resilient Synchronization Codeword for Variable-Length Code in Image Transmission", *IEEE conference of Image Processing*, Vancouver, 2000

[5] C. Tsai and J. Wu, "On Constructing the Huffman-Code-Based Reversible Variable-Length Codes", *IEEE Transactions on Communications*, vol. 49, No. 9, pp. 1506-1509, Sep 2001

[6] F. Pereira and T. Ebrahimi, "The MPEG-4 Book", pp 346-350, Prentice Hall, 2003