Error Resilient Hybrid Variable Length Codec* with Tough Error Synchronization for Wireless Image Transmission

Yew-San Lee, Cheng-Mou Yu, and Chen-Yi Lee

Department of Electronics Engineering, National Chiao Tung University

1001, Ta Hsueh Road, Hsinchu, 300, Taiwan, R.O.C.

Tel : +886-3-5726111 Ext.54238; Email : mountain@royals.ee.nctu.edu.tw

ABSTRACT

Variable-length codes, also called Huffman code [1], is the most popular data compression technique used for solving channel bandwidth bottleneck in many image compression standards such as JPEG, MPEG-2, and H263. But, it is vulnerable to loss of synchronization if they are transmitted consecutively through a noisy channel. It will result in large drops in image quality. In [2], we have proposed a novel HVLC coding scheme, which exhibits high synchronization, error correction capability and low redundancy. Simulation results show that it can achieve high signal-to-noise ratio (PSNR = 28dB) compared to existed VLC schemes at bit error rate (BER) of 10^{-3} environment. Here, we present HVLC codec system. It can support up to 256-entry 12-bit symbol table. The codec system provides advanced options and fully programmable for pixel-based and DCT-based image coding algorithm. Based on 0.35um CMOS SPQM technology, the codec system occupies a silicon area of 3400um x 3400um and consists of 88,000 gate count. It can operate at 3V power supply and yields high encoding/decoding rate of 133M-Symbol/sec for real time application requirement.

1. INTRODUCTION

Recently, the advent of wireless personal communication services has created a number of challenging research problems in the areas of communication, signal processing and networking. In order to minimize storage requirements and face with bandlimited wireless channel, it is necessary to employ compression techniques that can reduce the data rate without sacrificing image quality. Variable length code (VLC) which has minimal redundancy for a given data source is the most popular technique used in many image compression standards. VLC are more efficient at reducing the bit rate than fixed-length codes (FLC). Another major challenge in dealing with the wireless channel has to do with its inherent unreliability. This is contrast with wired networks in which the physical loss is very small, e.g. of the order of 10^{-9} . Hence, error-resilient VLC coding scheme becomes an important design issue.

A main problem in VLC bitstream transmission is to maintain the synchronization between the encoder and the decoder. Although VLC ensures optimal coding, i.e., minimal redundancy, it is highly sensitive to channel errors when VLCs are transmitted

consecutively through a noisy channel. Any bit error in VLC bitstream can lead to lose synchronization and cause error propagation as shown in Fig. 1. Mostly, errors may propagate for a considerable period, with an ensuing loss of many dozens of words, before synchronization is re-established. Therefore, designing VLC that can easily recover the synchronization becomes very important for wireless image transmission. We propose a novel hybrid variable length code (HVLC) [2], it has good tolerance to random and burst errors in worsening channel conditions. The average codeword length is close to the optimal.

Figure 1 : VLC lose synchronization and error propagation

This paper is organized as follows. In section 2, we first introduce HVLC coding scheme and exception error handling. Section 3 will describe package structure alignment mechanism. Then we present HVLC codec architecture design in section 4. Simulation results and performance comparisons are provided in section 5. Conclusions will be made in section 6.

2. HVLC Coding Scheme

An error occurring in VLC bitstream can lead to a loss of synchronization, which in turn can result in the insertion or deletion of symbols at the decoder and cause error propagation. The synchronization capability needs an additional requirement for a code that leads to a loss of efficiency in terms of average codeword length. In fact, apart from trivial cases, no optimal binary code is synchronizable [3]. Therefore, it is interesting to consider codes that satisfy the weaker property of admitting decoders able to self-synchronize with probability approaching 1. Many authors have studied the ability of VLC code to synchronize. See [4] and [5] for a discussion of codes with bounded synchronization delay, and [6]-[7] for statistically synchronizable codes. Others [8]-[12] propose variable-length

* WORK SUPPORTED BY THE NATIONAL SCIENCE COUNCIL OF TAIWAN, R.O.C., UNDER GRANT NSC89-2218-E-009-080

code constructions, which enable the decoder to correctly synchronize. Nevertheless, the range and richness of the coding structures implying in variable-length codes makes the identification of general mechanisms for self-synchronization difficult, perhaps impossible.

In [2], we propose a novel HVLC constructed by multiple virtual channels. Each codeword of HVLC is combined by several codewords of the same wordlength. A code structure of HVLC is depicted in Fig. 2. There have two virtual channels called “base channel” (BC) and “main channel” (MC). For the BC code “10” and the MC code “01”, the constructed HVLC codeword is “1001”. MC and BC codes are interleaved in even and odd bit location respectively.

Figure 2 : HVLC Code Structure

Because BC code is used to align codeword boundary, we should choose codes with short ASD (average synchronization delay) [13]. Because BC code already has prefix code characteristic, which ensures HVLC is prefix code. MC codes do not need to satisfy prefix code condition. So, we can use some conventional fixed length error correctable coding schemes for MC code. Therefore, we can correct error, which is occurring in MC portion. In addition, it will not lead to lose synchronization even MC code is uncorrectable since the codeword boundary is determined by BC code.

With multiple channels VLC coding structure, HVLC will only have 50% probability to lose synchronization within one codeword when error happened in BC code. So, the average synchronization delay (ASD) is only 0.5 codeword. It greatly surpasses other synchronous optimal codes. In addition, HVLC has better signal-to-noise ratio compared to other VLC schemes because it has error detecting and correcting capability in MC code. Because of interleaving coding structure, burst error will be transformed into random error in BC and MC code that will be easier for handling. Another performance factor of HVLC is average codeword length (ACL). The ACL of HVLC is close to the optimal. For JPEG run/size VLC table, FLC is 8, Huffman code is 3.40 and HVLC is 4.15. Small ACL means high data compression capability. The proposed HVLC is efficient for using in bandwidth-limited and noisy channel where image quality is a requirement issue.

By observing HVLC coding scheme, we can discover that symbols with the same or similar probability will be grouped together and have the same codeword length. They will be assigned with the same BC code. The codeword in each group can be treated as a binary number. In order to applying arithmetic coding operation, each group requires four portions of information : base_addr, code_length, ECMC, and BC_code. “base_addr” is the base address of each group of codeword. “code_length” (n) indicates the codeword length (2n). “ECMC” is the MC coding schemes discussed in [2]. “BC_code” is the BC

code pattern of the group. For 256-entry symbol table with maximum 16-bit codeword length and partitioned into 8 group, total memory space requirement for storing group information is only $(8 \times 21) = 168$ bits (base_addr : 8 bits, code_length : 3 bits, ECMC : 2 bits, BC_code : 8 bits). Here, we give an example for describing the detail arithmetic coding operation.

Encoding :

Assume encoded symbol address is 15

- Group-searching
base_addr₅(8) < symbol_addr(15) < base_addr₆(20)
Matching group-5 (G5)
- Fetch G5 group information
code_length=3'b101=5, base_addr=8'b00001000
ECMC= 2'b01, BC_code=8'b00011110
- Since code_length=5, codeword length=10
BC code pattern=5'b11110 (LSB 5-bit of BC_code)
- enc_offset=symbol_addr - base_addr₅ = 8'b00000111
- Generate MC code using scheme indicted by ECMC
MC code pattern=5'b00111 (LSB 5-bit of enc_offset)
- BC and MC code are combined to form codeword :
10'b1010111101

Decoding :

Assume decoded bitstream is 1010111000101110

Extracted BC_code = 11110111, MC_code = 00100010

BC_code₅ = 8'b11110000, BC_code₆ = 8'b11111000

- Group-searching
BC_code₅ < BC_code (8'b11110111) < BC_code₆
Matching group-5 (G5)
- Fetch G5 group information
code_length=3'b101=5, base_addr=8'b00001000
ECMC=2'b01
- Correct MC_code using ECMC scheme
Generate dec_offset
dec_offset = 5'b00100 (Get MSB 5-bit of MC_code)
- Calculating decoded symbol memory address
symbol_addr = base_addr + dec_offset = 12
- Fetch decoded symbol from memory : sym_mem[12] = S55

3. Package Structure Alignment Mechanism

Many image compression algorithms split the input signal into fixed blocks of data. Then, these blocks of data are coded using a variety of compression techniques including VLC to produce a variable-length block of coded data. Having done this, the coded data will be multiplexed together and transmitted consecutively. The coding process always inserts synchronization codeword (SC) as marker to localize error propagation. But, it is true that even if the errors have been localized and regained synchronization within using SC, the number of decoded symbols might probably be changed. It is because the error may break codeword into more than one codewords or combine some codewords into one codeword. For both situations, the inconsistency of decoded symbol number will form an offset and cause the following decoded image information being spatially shifted. In order to keep the number of decoded symbol within a packet the same, a compensation mechanism needs to be added into decoding procedure.

For unexpected burst errors, it may form incorrect SC (dummy SC) or result SC undecipherable. Both cases will degrade the

image quality significantly. In order to avoid these situations, SC detection will only be valid between $0.8N$ and $1.2N$ window (N is package size) in our decoding procedure. Out of this valid detection window, the detected SC needs to be ignored since it mostly is dummy SC. In addition, we propose package structure alignment mechanism for four different cases discussed as below. The mechanism is based on image data continuity characteristic. Because package structure can be aligned correctly, signal-to-noise ratio (PSNR) of the image, under error occurrence, can be enhanced effectively. For package size of N , $0.2N$ storage elements are required by the alignment mechanism for temporally storing decoded symbols.

Case-1 : SC is detected in $0.8N \sim N$ since errors result in the number of decoded symbols smaller than package size N . We insert dummy symbols to form a package of size N . Thus, package structure can be re-aligned. The inserted dummy symbol is using the last prior decoded symbol.

Case-2 : For errors result the number of decoded symbols larger than package size N , we will not detect SC in N . After SC is not detected in N , we temporally store the following decoded symbols and do not send them to backend processing until SC is detected in $N \sim 1.2N$. If SC detected in $N \sim 1.2N$, we will discard the stored symbols and start to decode next codeword as next package symbol.

Case-3 : For errors result in SC undecipherable, we will not detect SC in $N \sim 1.2N$. When counting to $1.2N$ and SC is still not detected, we will decide to treat stored symbols (from N to $1.2N$) as valid symbols and sent them to backend processing. Then, package alignment will not be broken.

Case-4 : For errors result in additional dummy SC, we will detect two SC in $0.8N \sim 1.2N$. No matter which one is real SC, we will apply case-1 or case-2 mechanism when detecting first SC in $0.8N \sim 1.2N$. Then, the following decoded symbols will not be stored and simply sent to backend processing. If we detect another SC in $0.8N \sim 1.2N$, it requires to discard the same amount of symbols (symbol count between these two SC) after this SC for aligning the package structure. Fortunately, it is seldom to detect this situation.

4. HVLC Codec System Architecture

In order to meet diverse applications and standard-defined tables, codec system highly needs the programmability to load various HVLC tables without redesign the hardware. Constant processing rate is also an important design issue for reducing the I/O interface complexity. Besides, high throughput rate is required in real time application. The proposed HVLC codec system is designed to target these issues and performances. Based on efficient memory mapping strategy, encoding and decoding processes can be completed by applying numerical properties and simple arithmetic operations. So, encoding and decoding one codeword can be done within one clock cycle. Related memory space requirement can also be optimized.

Block diagram of the HVLC codec system is shown in Fig. 3. It mainly consists of seven parts : group locator and database, arithmetic logic unit (ALU), stall detector, symbol memory, programmable parameters controller, encoder, and decoder architecture. Group locator and databases are used to parallel searching the encoded symbol or decoded codeword is belonging which group. There have two sets of group locator and database

for supporting DCT-based coding schemes (DC-Coefficient and AC-Coefficient Table). After group searching is confirmed, ALU performs arithmetic and logic operations of encoding and decoding procedures. Variable length coding will result inconsistent data rate between input and output interface. Stall detector is required to detect overflow and underflow situation, which depending on encoding output FIFO and decoding input FIFO status. It will send stall signal to prior stage to stop sending data into codec system. The FIFOs depth has been designed to minimize the stalling effect. Symbol memory is synthesized by in-house cell library and stored with related symbol table. Before starting operation, it needs to load relative parameters, group information, operation mode (encode/decode, pixel-based/DCT-based), package size, and symbol table. The programmable parameters controller will control the loading sequence and encoding/decoding operations.

Figure 3 : Block Diagram of HVLC Codec System

Encoder part consists of MC code generator, SC insertion block, Enc_bitstream concatenator buffer, and serial-to-parallel output FIFO. Enc_bitstream concatenator buffer will combine the MC code and BC code fetched from group database to form HVLC codeword structure. SC insertion block will insert SC into HVLC bitstream when package size or block size is counted. After certain amount (programmable) of HVLC codewords are collected, encoder starts to send out the encoding bitstream (8-bit) continuously. Bus utilization rate can be optimized. Decoder architecture is more complicated than encoder. Received HVLC bitstream will be fed into serial-to-parallel interface FIFO. Dec_bitstream separator will extract BC code and MC code from the HVLC bitstream. Group determinator will decide group property of the codeword. Then, codeword boundary alignment block can shift and align the received bitstream correctly. Main channel code EDC block will correct the extracted MC code using ECMC scheme defined in the group database. Finally, symbol address calculator will derive symbol address and fetch the decoded symbol from symbol memory. Concurrently, SC detector and package alignment block will perform mechanisms described in section 3. Similar to encoder, decoding output FIFO will start to output results continuously after collecting certain amount (programmable) of decoded symbols. Receiving bitstream, decoding operation, and sending out decoded symbols can be performed at the same time.

The designed HVLC codec system can support pixel-based and DCT-based image coding algorithm. In addition, it provides programmable package size and SC detection valid window. The system includes a symbol memory (256 x 12), which can be exploited for table up to 256-entry 12-bit wordlength symbols. These flexibilities can be utilized to meet various application requirements and achieve high image quality. It is core-limited designed and consists of 88,000 gate count. The codec system chip is implemented using CMOS 0.35um SPQM technology and occupying a silicon chip area of 3300um x 3300um. The layout view of the codec system chip is shown in Fig. 4.

Figure 4 : HVLC Codec System Chip Layout View

5. SIMULATION RESULT

In order to examine the effectiveness and performance of HVLC for real coding scheme, we have simulated the Lena image (512 x 512, 256-level grayscale) using pixel-based and DCT-based coding schemes. Both coding schemes are simulated for different package size (pixels) and frame size (DCT-blocks) at BER of 10^{-3} . The PSNR results are shown in Fig. 5 and 6. On the average, PSNR of HVLC is at least 3dB ~ 10dB better than T-code, B2-code, and Huffman code.

Figure 5 : Pixel-based PSNR with Different Package Size

Figure 6 : DCT-based PSNR with Different Package Size

5. CONCLUSION

Image transmission over wireless channel requires higher error resiliency than required in other channels. We propose a novel HVLC codec system, it exhibits high synchronization (ASD = 0.5 codeword), error correction capability and low redundancy. The average codeword length is close to the optimal. Simulation results show that HVLC can achieve high signal-to-noise ratio (Pixel-based : 28dB; DCT-based : 22dB) compared to other VLC schemes at BER of 10^{-3} environment. The HVLC codec system can support pixel-based and DCT-based coding algorithm. The memory-based architecture provides flexibility to handle different coding tables up to 256-entry 12-bit symbols. Based on 0.35um CMOS SPQM technology, the codec system occupies 3300um x 3300um silicon area and consists of 88,000 gate count. It can operate at 3V power supply with clock rate up to 133MHz. So, 133M-Symbol/sec constant encoding/decoding rate can be achieved for real-time application requirements.

6. REFERENCES

- [1] D.A. Huffman, " A method for the construction of minimum-redundancy codes, " in *Proc. IRE*, vol. 40, pp.1098-1011, Sept. 1952.
- [2] Y.S.Lee, C.M. Yu, C.H. Chang, and C.Y. Lee, " HVLC : Error Correctable Hybrid Variable Length Code for Image Coding in Wireless Transmission, " in *IEEE ICASSP'2000*, June 2000.
- [3] J.J. Stiffler, *Theory of Synchronous Communications*. Englewood Cliffs, NJ : Prentice-Hall, 1971.
- [4] S.W. Golomb and B. Gordon, " Codes with bounded synchronization delay, " *Inform. Contr.*, vol. 8, pp.355-372, 1965.
- [5] V.I. Levenshtein, " Certain properties of code systems, " *Sov. Phys. - Dokl.*, vol. 6, no. 6, pp.858-860, 1962.
- [6] R.M. Capocelli, L. Gargano, and U. Vaccaro, " On the characterization of statistically synchronizable variable-length codes, " *IEEE Trans. Inform. Theory*, vol.34, pp.817-825, July 1988.
- [7] R.M. Capocelli, A.A. De Santis, L. Gargano, and U. Vaccaro, " On the construction of statistically synchronizable codes, " *IEEE Trans. Inform. Theory*, vol.38, pp.407-414, Mar. 1992.
- [8] V.K.W.Wei and R.A. Sholtz, " On the characterization of statistically synchronizable codes, " *IEEE Trans. Inform. Theory*, vol. IT-26, pp.733-735, Nov. 1980.
- [9] T.J. Ferguson and J.H. Rabinowitz, " Self-synchronizing Huffman codes, " *IEEE Trans. Inform. Theory*, vol. IT-30, no. 4, pp.687-693, July 1984.
- [10] B.L. Montgomery and J. Abrahams, " Synchronization of binary source codes, " *IEEE Trans. Inform. Theory*, vol. IT-32, no. 6, pp.849-854, Nov. 1986.
- [11] J.C. Maxted and J.P. Robinson, " Error recovery for variable length codes, " *Bell Syst. Tech. J.*, vol. 31, no. 6, Nov. 1985.
- [12] M.R. Titchener, " Generalised T-codes : Extended construction algorithm for self-synchronizing codes, " in *Proc. Inst. Elec. Eng.*, pt. F, *Computers and Digital Techniques*, vol. 143, no. 3, June 1996.