論文 97-22-9-23

(0, k) Run-Length Limited(RLL) Data Compression Codes for Digital Storage Systems

Jaejin Lee* *Regular Member*

디지털 저장시스템을 위한 (0, k) RLL 데이터 압축코드

正會員 이 재 진*

※본 논문은 1997년 동국대학교 신임교수 연구지원의 결과임

ABSTRACT

Much recent work has been done in the two related areas of source coding for data compression, and channel coding for data storage, respectively. We propose two (0, k) run-length limited(RLL) data compression codes for the storage that combine source and channel coding. It was shown that the proposed codes approach the maximum code rate of (0, k) code as k increases. Thus, the overall code rate of storage system can be increased by using the combined source/channel code as compared to the conventional 8/9 code which is popular in hard drive systems. Furthermore, one can also reduce the complexity of modulation coding procedure by using already RLL constrained data.

요 약

지금까지 소스코딩과 채널코딩은 서로 관계가 있으면서 독자적으로 많은 발전을 이룩하여왔다. 여기서는 스토리지 채널을 위해 소스코딩과 채널코딩을 접목시키는 두 가지의 압축코드를 소개한다. 실험 결과, 제안된 코드가 k 조건이 증가함에 따라 (0, k) RLL 코드의 채널용량에 근접함을 알 수 있었다. 따라서, 현재 하드드라이브 시스템에 적용되고 있는 8/9 코드와 비교할 때, 전체적으로 코드율을 증가시킬 수 있다. 더욱이, 변조되어야 할 데이터가 이미 k 조건을 만족하고 있으므로 변조코드 부분의 복잡도를 줄일 수 있게 된다.

*동국대학교 전자공학과
論文番號:97210-0621
接受日字:1997年 6月 21日

Ⅰ. Introduction

In many multimedia applications, rapid access/storage of image and video data in meeting the demands of real-time constraints is a basic necessity. Compression can be used to reduce the amount of data that is stored and retrieved, and can be realized through source coding. This source coding is in addition to the channel coding already utilized in storage channel to mitigate intersymbol interference and loss of timing.

A popular channel code for the storage channel is the run-length limited (RLL) code which constrains a run of the zeros to be at least d for preventing intersymbol interference(ISI), and at most k between two neighboring ones in NRZI signaling to preserve synchronization information. McLaughlin and Neuhoff [1] have proposed a source-channel code that increases the storage capacity of analog data samples on magnetic media using tree structured vector quantizers.

This paper proposes two new combined data compression codes (having the run-length constraints for the storage channel) using the Huffman code[2] which gives a minimum average codeword length for a finite source alphabet. One can objectively measure the performance of combined source/channel codes by comparing the maximum achievable code rate R_{new} (given by the average length of conventional source coded codeword divided by the average length of a new source/channel coded codeword) and conventional modulation code rate.

Without loss of generality, let the finite alphabet be the set of positive integers {1, 2, 3,..., N}. Let $\{p_1, p_2, \cdots, p_N\}$ be the Huffman probabilities of the set, with p_i ordered such that $p_i \geq p_j$ if $i < j$. If, in addition, these probabilities satisfy:

$$p_i \geq p_{i+1} + p_{i+2} \tag{1}$$

for $i = 1, 2, \cdots, N-2$, then the optimum codeword set is a comma code (in other word, unary code) such as

$$C_0 = \{c_1, c_2, \cdots, c_N\}$$

where

$$c_i = \begin{cases} 0^{i-1}1, & 1 \leq i < N \\ 0^N, & i = N \end{cases} \tag{2}$$

The notation b^n represents n consecutive b, for instance, "$(0^3 1)^2 1$" is equivalent to "000100011." In Table 1, the second column is an example of a codeword with alphabet size N = 10. The codeword length of each alphabet is $l_i = i$ for $1 \leq i < N$, and $l_N = N - 1$, and the average codeword length is

$$L_0 = \sum_{i=1}^{N} i p_i - p_N \tag{3}$$

Humblet [3] has shown that the optimum codeword has a unary tail if the probability decreases faster than $(0.618)^i$. The Rice coding technique of [4] is also based on the Fundamental Sequence code (which is equivalent to a comma code) for an ordered alphabet set. However, these data compression codes possess long zero string sequences which violate the maximum run-length constraint for the storage channels. Therefore, we propose two techniques that avoid loss of synchronization while minimizing redundancy of the code. Using the proposed source/channel codes, one can reduce the complexity of coding scheme of storage systems.[5]

These encoding and decoding procedures are presented in Section II. The average codeword length and redundancy of each code are also mentioned. Application of the proposed code to real data such as text file and an image file is presented in Section III. Section IV contains the conclusion.

Ⅱ. (0, k) RLL Data Compression Codes

The coding methods described in this section attempt to compress the source alphabet while satisfying the maximum and minimum run-length constraints for storage channels. Assume that the

source alphabet is a finite set of positive integers, and its probabilities are ordered and satisfy the condition (1). Then, any element of the alphabet i is expressed as a function of the maximum run-length constraint k such as

$$i = xk + y \tag{4}$$

where x is a quotient, and y, $0 \le y < k$, is a remainder. Also x, y and k are positive integers.

1. Code I

If $i < k$, the encoded codeword is equivalent to the Huffman code:

Table 1. Example of codewords: N = 10, $k = 4$

source i	Huffman code C_0	Code I C_1	Code II C_2
1	1	1	1
2	01	01	01
3	001	001	001
4	0001	00011	0001
5	00001	000011	000011
6	000001	0001011	0000101
7	0000001	00010011	00001001
8	00000001	000100011	000010001
9	000000001	0001000011	00001000011
10	000000000	00010001011	000010000101

$$c_i = 0^{i-1}1 \tag{5}$$

When $i \ge k$, this code utilizes "11" to mark the end of the code, and "1"s are inserted preventing a long sequence of zeros which violates k constraint.

$$c_i = \begin{cases} (0^{k-1}1)^x 1, & y = 0 \\ (0^{k-1}1)^{x-1}0^k 11, & y = 1 \\ (0^{k-1}1)^x 0^{y-1}11, & 1 < y < k \end{cases} \tag{6}$$

In Table 1, the third column is an example of a codeword for alphabet size N = 10 and maximum run-length constraint $k = 4$.

The average code length L_1 then, is

$$\begin{aligned} L_1 &= [1 \cdot p_1 + 2 \cdot p_2 + \cdots + (k-1)p_{k-1}] \\ &\quad + [(k+1)p_k + (k+2)p_{k+1} + \cdots + (N-1)p_N] \\ &= \sum_{i=1}^{N} i p_i + \sum_{i=k}^{N} p_i \\ &= L_0 + (1-\alpha) + p_N \end{aligned} \tag{7}$$

where

$$\alpha = \sum_{i=1}^{k-1} p_i, \tag{8}$$

and L_0 is given in (3). The redundancy of the Code I is $1 - \alpha + p_N$, and the maximum achievable code rate is

$$R_1 = \frac{L_0}{L_1} = \frac{1}{1 + (\frac{1 - \alpha + p_N}{L_0})} \tag{9}$$

Thus, R_1 converges to 1 as k increases (because α goes to 1 as k increases) for large N.

The decoding procedure for Code I is shown in Figure 1, and also outlined below.

Step 1. While counting the number of zeros, N_b, until a 1 is reached,

· If $N_b \ge k$, keep counting the number of bits until the string "11" occurs. Then, the decoded word is

$$\hat{i} = decode(c_i) = N_b - 1 \tag{10}$$

where N_b is the total number of bits including the string "11."

· If $N_b < k$, then the decoded alphabet is

$$\hat{i} = decode(c_i) = N_b \tag{11}$$

Step 2. Reset the counter, and repeat the Step 1.

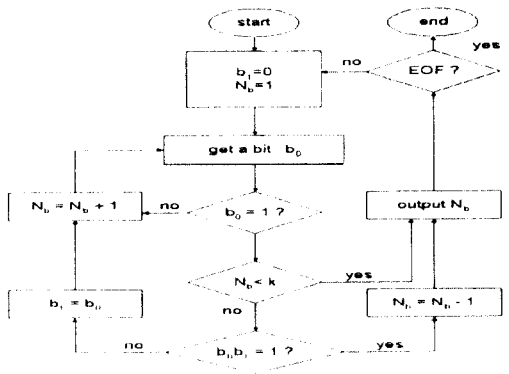

Fig. 1. Decoding procedure for (0, k) constrained data compression Code I

2. Code II

For all $i = 1, 2, \cdots, N$, let the source alphabet be expressed as

$$i - 1 = xk + y \tag{12}$$

where k is the maximum run-length constraint, x is a quotient, and y is a remainder such that $0 \le y < k$. To satisfy k constraint, "1"s are inserted after every k zeros. Thus, Code II encodes source alphabet i as following.

$$c_i = (0^k\ 1)^x\ 0^y\ 1 \tag{13}$$

In Table 1, the fourth column is an example of a codeword of type II, when the alphabet size is N = 10 and the maximum **run-length constraint $k = 4$**.

The code length of each codeword is

$$\begin{aligned} l_i &= (k+1)\,x + y + 1 = i + x \\ &= i + \lfloor \frac{i-1}{k} \rfloor \end{aligned} \tag{14}$$

where $\lfloor x \rfloor$ is the largest integer less than or equal to x. The average code length of the code is

$$L_2 = L_0 + R(L_2) \tag{15}$$

where the redundancy is

$$R(L_2) = p_N + \sum_{i=k}^{N} \lfloor \frac{i-1}{k} \rfloor\ p_i \tag{16}$$

The code rate of the Code II is

$$R_2 = \frac{L_0}{L_2} = \frac{1}{1 + (p_N + \sum_{i=k}^{N} \lfloor \frac{i-1}{k} \rfloor\ p_i)/L_0} \tag{17}$$

Thus, as k is increased, the code rate converges to one.

The coding procedure is illustrated in Figure 2. Given the maximum run-length constraint k, decoder checks the status of variables x and y where x increases every k bits, and y increases every bit and reset to 1 whenever x is increased or reset to zero. Every $(k+1)$st bit is discarded while a codeword is decoded. Then, the decoded alphabet is given by

$$\hat{i} = decode(c_i) = xk + y \tag{18}$$

for all $i = 1, 2, \cdots, N$.

This code is simpler than the Code I, even though the average codeword length is longer than that of Code I.

Ⅲ. Discussion

It appears that one can remove the condition (1) on the source alphabet. Then, the proposed codes would apply to arbitrary source probability distribution such that the "maximum run-length constraint violated sequences" in the binary Huffman tree are substituted by those sequences of Figure 1. In order words, one can substitute those consecutive zero sequences violating the maximum run-length constraint with the corresponding constrained codewords using Eq. (5) for Code I and Eq. (13) for Code II, while i in (4) and (12) represents the number of consecutive zeros between two neighboring ones.

Table II shows an example of code rates obtained

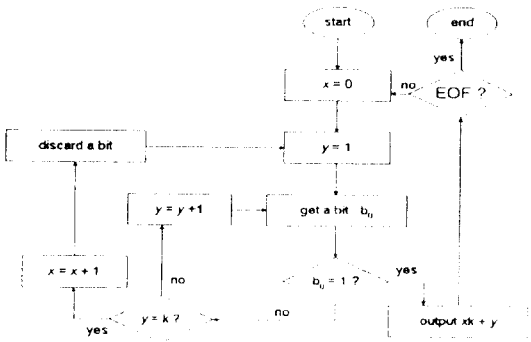

Fig. 2. Decoding procedure for (0, k) constrained data compression Code II

Table 2. Code rates of Code I and Code II depending on k and type of data file

k	capacity of (0, k) code	Text file		Image file	
		Code I	Code II	Code I	Code II
2	0.8791	0.7464	0.8245	0.7518	0.8351
3	0.9468	0.8696	0.9288	0.8769	0.9283
4	0.9752	0.9378	0.9635	0.9376	0.9668
5	0.9881	0.9653	0.9825	0.9692	0.9838
6	0.9942	0.9830	0.9918	0.9843	0.9917
7	0.9971	0.9920	0.9960	0.9918	0.9954
8	0.9986	0.9961	0.9983	0.9954	0.9977
9	0.9993	0.9984	0.9992	0.9977	0.9990
10	0.9996	0.9993	0.9996	0.9990	0.9996
11	0.9998	0.9996	0.9997	0.9996	0.9998
12	0.9999	0.9997	0.9999	0.9998	0.9999

via tests on a small text file and an image file. The text file is size of 13,941 bytes, and the image file is a black and white "Lena" image (262,176 bytes). The code rate is the size of conventional Huffman coded file divided by the size of the proposed coded file. The second column is the maximum achievable code rate of (0, k) code. From the table one can notice that Code II performs better than Code I. Since the probability of long zero sequence is less than that of short one, the code rates are almost same in the limit for $k > 9$. Thus, the overall code rate of storage system can be increased by using the combined source/channel code(especially, Code I) as compared to the conventional 8/9 code which is popular in hard drive systems. Furthermore, one can also reduce the complexity of modulation coding procedure by using already RLL constrained data.

Ⅳ. Conclusion

This paper has introduced two data compression codes that combine source/channel coding steps for digital data storage. It was shown that the proposed codes approach the maximum code rate of (0, k) code as k increases.

References

1. S. W. McLaughlin and D. L. Neuhoff, "Source-channel coding for digital magnetic recoding," Proc. of Asilomar Conference on Signal, Systems and Computers, pp. 20-24, 1991.
2. D. A. Huffman, "A method for the construction of minimum redundancy codes," Proc. of IRE, vol. 51, pp. 251-252, Sept. 1952.
3. P. A. Humblet, "Optimal source coding for a class of integer alphabets," IEEE Trans. on Inform. Theory, vol. IT-24, no. 1, pp. 110-112, Jan. 1978.
4. R. F. Rice, P. Yeh and W. H. Miller, "Algorithm for high-speed universal noiseless coding," Proc. of the AIAA computing in aerospace 9 conference, pp. 499-506, San Diego, CA, October 19-21, 1993.
5. Jaejin Lee, "Combined source/channel coding and a new coding scheme for data storage systems," submitted to IEEE Trans. on Magnetics.

이 재 진(Jaejin Lee) 정회원

1983년 2월 : 연세대학교 전자공학과 학사
1984년 12월 : University of Michigan, EECS(공학석사)
1994년 12월 : Georgia Institute of Technology, ECE (공학박사)
1993년 7월~1994년 12월 : Research Assistant, Georgia Tech
1995년 1월~1995년 12월 : Research Associate, Georgia Tech
1996년 1월~1997년 2월 : 현대전자 정보통신연구소 책임연구원
1997년 3월~현재 : 동국대학교 전자공학과 교수(전임강사)
※주관심분야 : 통신신호처리, 채널코딩, 자기 및 광기록 저장장치