

P5

3:30 pm – 5:30 pm

Guided Scrambling: A New Coding Technique for Holographic Storage

Wang Yong Hong Wilson, Kees A.S. Immink[†], Xu Bao Xi and Chong Tow Chong
Data Storage Institute, DSI Building, 5 Engineering Drive 1
(off Kent Ridge Crescent, NUS) Singapore 117608

Tel. (65) 8746600 Fax. (65) 7766527 e-mail: dsiyhw@dsi.nus.edu.sg

[†]Institute for Experimental Mathematics, Ellernstrasse 29, 45326 Essen, Germany
e-mail: immink@exp-math.uni-essen.de

1 Introduction

Holographic storage of digital data is a promising technique for reliably storing mass data [1]. In holographic storage entire two-dimensional (2D) pages of data are intermingled with each other such that numerous pages of data are recorded in a common volume of the storage medium. Certain pages with vexatious data patterns should be avoided as they could easily interact with other pages. In particular data with a periodic pattern give rise to distinct peaks of intensity in the Fourier transform plane. Vardy *et al.* [2] stipulated that the data patterns should have as many as transitions from light to dark and *vice versa* as possible in each column and row of a page. This may be achieved by requiring that in each row and column, there are at least t (t is a positive prescribed integer) transitions of the type “1” → “0” or “0” → “1”. Such an array is called a *conservative array* of strength t . There is another requirement for holographic storage that the number of “1”s and “0”s in a two-dimensional array should be balanced, so that the content of the data pages will be independent of the amount of signal light for recording.

Guided scrambling [3] is a member of a larger class of related coding schemes called *multi-mode* code. In multi-mode codes, each source word can be represented by a member of a selection set consisting of L codewords. The encoder evaluates the “quality” of each codeword in the selection set, and transmits the codeword that “best” matches the quality criterion at hand. The members of the selection sets are not judiciously chosen and stored in memory, and are randomly picked. The basic idea is that, provided the selection set is sufficiently large, we will find, with high probability, an adequate codeword fulfilling the constraints. The advantage of this approach is clear: at the encoder site we only need a) a simple mechanism for translating source words into “random” codewords and b) a mechanism for evaluating the “quality” of the candidate words. The stumble block of conventional large codes and the look-up table can be avoided, and replaced by a simple randomization algorithm.

2 Description of the Coding Scheme

For a data page of holographic storage, assume a source word of length n bits is given, the source word is preceded by p redundancy bits. In the final coding stage the $n + p$ bits are arranged in a matrix (page). We produce the set \mathcal{B}_x , which comprises all $n + p$ -bit words found by augmenting the source word with all possible 2^p redundant p -bit words. Each $(n + p)$ -bit word is forwarded to a scrambler, and the output of the scrambler is arranged in the required matrix form. The scrambling sequence can be obtained through the following feedback polynomial:

$$s_k = b_k + g_1 s_{k-1} + \dots + g_n s_{k-n}, \quad (1)$$

where s_k denotes the scrambled data, b_k the input data, and g_n the coefficients of the scrambler. Both the input data, output data, and coefficients $\in \{0, 1\}$, the multiplication and the addition are modulo-2. The largest index value of the non-zero coefficient is referred to as the *order* of scrambler. With a certain structure, the scrambling output sequence has a long periodicity and small correlation, and it is thus called *pseudo-random sequence*. After processing all possible arrays, a matrix taken from the set of 2^p generated matrices that best fulfills the two requirements for holographic storage will be chosen and stored in the holographic medium. In Section 3, we will show that such kind of sequence can be found with quite large probability.

Note that there is no necessity to transmit additional information to the receiver in order to decode the original data. The decoder just uses the same method to descramble the received data sequence with the following relation:

$$b_k = s_k + g_1 s_{k-1} + \dots + g_n s_{k-n}. \quad (2)$$

Clearly, such a kind of coding scheme has the advantages of reliability and simplicity. Error propagation is limited to the number of non-zero coefficient of descrambler, which is not a big deal. The number of preceding bits, p , is considered as redundancy. As p is relatively small, it offers high coding efficiency.

3 Results

We design the encoder for the two-dimensional 16×16 array. The scrambler order is 11, and the configuration of the scrambler is

$$s_k = b_k + s_{k-2} + s_{k-11}. \quad (3)$$

Note that the method cannot guarantee to find an array with specified constraints, though if the intermediate set is sufficiently large such an array is found with probability approaching unity. For example, Table 1 lists the probability Pr that a balanced array (an array with equal numbers of 'zero's and 'one's) is not found in a population of size 2^p of random 16×16 arrays.

Table 1: Probability Pr that a balanced 16×16 array is not found in a set of size 2^p .

p	Pr
6	3.80E-002
7	1.44E-003
8	2.08E-006
9	4.33E-012
10	1.88E-023
11	3.53E-046

In the simulation we first choose the balanced (equal numbers of 1's and 0's) array, then investigate the transition property of each row and column. As is disclosed in Table 1, the larger the value of p , the more balanced the arrays can be found. Thus there is more choice to find a better conservative array. Different values of p affect the "randomization" of the recorded data page, that is the strength of the conservative array. We investigate this characteristic by collecting the distribution of the strength t for different values of p . Table 2 shows the result. It is shown that as p is increased, the strength t will approach larger values with little variance. For example, when $p = 11$, all arrays have strength > 4 .

