Adaptive code modulation for 2D optical memories

Leo Selavo, Donald M. Chiarulli, Steven P. Levitan
Department of Computer Science, University of Pittsburgh, 316 Mineral Industries Building, Pittsburgh, PA 15260
Phone: (412)-624-8839 Fax: (412)-624-5249
Email: elo@cs.pitt.edu, don@cs.pitt.edu, steve@ee.pitt.edu

1. ABSTRACT

Volumetric and page oriented optical storage systems have a great potential for high data density and high speed through page oriented data access [1]. However, in order to be practical, they have to maintain data integrity using error correction and modulation codes. The coding comes at a cost of spatial overhead of codeword size over the source data size, and processing time overhead for encoding and decoding. Our proposed adaptive modulation scheme strives to minimize both overheads. It achieves utilization of 74% with constant time complexity, which is competitive with other recently proposed schemes for page-oriented memories.

2. INTRODUCTION

Figure 1 illustrates a binary source data page (a) being written to an optical media and then read (b) and sampled with a detector array (c). The detectors apply a threshold to the received intensity of every pixel, thus converting it to binary data. Ideally the resulting data array should mach the source array (a). However, it happens seldom due to such destructive effects as inter-symbol interference (ISI). Therefore, error coding and modulation mechanisms must be introduced. In this paper we focus on the latter.

a) Binary data   b) Detector image   c) Sampled image

Figure 1: Raw data images

ISI is observed in Figure 2, where intensities of read bits are plotted against number of bits read at this intensity. There are two curves, one representing bits encoded as ‘0’ and the other for ‘1’. The curves intersect between intensities A and B; therefore it is impossible to determine if the bit was ‘0’ or ‘1’. The goal of the adaptive encoding scheme is to avoid such situations, and achieve ISI-free readout as seen in Figure 3. This is achieved by spatially separating potentially conflicting symbols on the media so that the readout is always below or above threshold and noise margin.

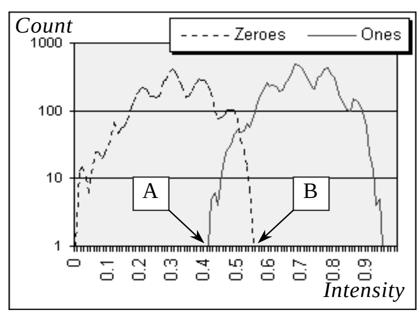

Figure 2: Raw data histogram

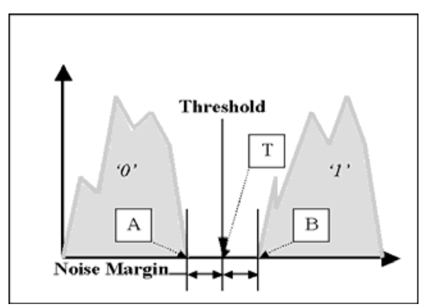

Figure 3: Adaptive encoding goal

The spatial overhead of a code is usually measured by its *code rate*, expressed as the source data word size over the codeword size. In our approach the code rate may become variable, since several codes will be used adapting to the current situation. Therefore, we introduce a code quality metric called *media utilization*, expressed as the source data block size over the encoded data block size in percentage. Higher utilization indicates more efficient codes. For all fixed code rate codes, the code rate and utilization are the same.

One of the most recent two-dimensional encoding schemes for page-oriented memory is the 4/9-modulation scheme developed at the University of Southern California [2]. It is a static code, which uses a fixed translation table and has utilization of 45%. The alternative to fixed codes are dynamic codes, which take into account the codeword of the surrounding data on the data page and choose an optimal code for each codeblock location. In [3] we showed that dynamic encoding could potentially achieve over 80% media utilization.

Unfortunately, dynamic encoding comes at a cost of higher computational complexity. Reaching the best utilization costs exponential time in the size of code block. This paper proposes alternative dynamic encoding and decoding techniques, which can be done in constant time, while achieving over 70% utilization.

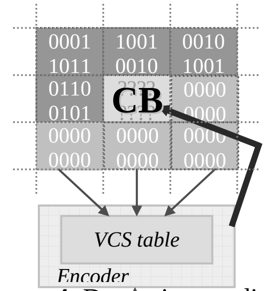

Figure 4: Dynamic encoding

3. APPROACH

The static encoding uses one fixed code, which allows it to be performed in constant time. However, it always assumes the worst-case scenario, and thus has relatively low utilization. Dynamic encoding, on the other hand, always strives to adapt to the current data layout on the page, and chooses the optimal code for each particular code block (CB) location. It examines the surrounding data, and then constructs a valid code set (VCS), which does not violate the defined noise margins. Finally, the source data is encoded using the VCS table (Figure 4).

Dynamic decoding reads and applies the threshold to all pixels. The resulting binary array is computed into an array of light intensities, considering ISI. The same VCS, as was used for encoding, is computed for each CB location, and the CB is decoded to the original data.

The VCS for dynamic encoding can be tailored by changing the policy and heuristics of how the code set is chosen for each code block, trading utilization for computational complexity. An extreme case is to try all possible code sets and determine the most efficient set, alas, at a cost of exponential time complexity for both encoding and decoding [3]. The adaptive encoding, proposed in this paper, offers a compromise, providing slightly lower utilization, but performed in constant time. This allows for the implemention of fast hardware for encoding and decoding.

Adaptive encoding works as follows: Each source data word has a set of one or several codewords, which can encode it. The pair-wise intersection of all these sets must yield an empty set in order to avoid ambiguities during decoding. As a source data word is received at the input, all the codewords are tested on the memory page until a valid codeword is found. This codeword is used for encoding. If a situation occurs when none of the codewords test valid (due to high density of surrounding data intensity), a blank codeword is written, indicating that this memory location is to be skipped.

For decoding a codeword is read from the media, and immediately it is determined to which set it belongs, and thus what source data word it was used to encode.

The adaptive coding is designed by constructing the sets of codewords for each source data word. The codewords are grouped in sets so that they maximize the likelihood of the following event: if a codeword tests invalid, then some other codeword from the same set (for the same source word) will test valid. The likelihood is maximized across all the sets, thus maximizing the likelihood that it will be possible to encode any incoming source data word.

4. SIMULATION MODEL AND RESULTS

The encoding performance was evaluated using a memory model, which represents a two-photon optical memory. The memory was organized in 2-dimentional pages. Encoded ‘1’ bits were interpreted as light sources and modeled using a point spread function. Thus, the light intensity pattern for each page was computed by taking into account ISI. The data page readout was modeled by applying a threshold to the calculated light intensity of each data pixel. The memory model used the following data page sizes for experiments: 64x64, 128x128, 256x256 and 512x512. The simulation parameters were identical to those in [2,3] in order to achieve comparable results. The model can be easily modified to accommodate other types of page-oriented memories.

The experiments were conducted to determine and compare performance of different page oriented memory modulation techniques: Static encoding (4/9 from USC) [2], Dynamic encoding, using full code set enumeration, a linear-time "either-neither" algorithm [3], and the new adaptive encoding. Different sizes of code blocks were tried for the dynamic encoding algorithms. The same data source with a uniform distribution of '1' and '0' bits was used for all encoding experiments. The results of the most promising code block sizes are shown in Table 1.

Algorithm	Code block size	Utilization	Time complexity (CB size)
4/9 static encoding (from UCSD)	3 x 3	< 45%	Constant
Dynamic encoding: Full enumeration	2 x 2	71%	Exponential
-//-	3 x 3	83%	Exponential
Dynamic encoding: "Either-Neither"	3 x 3	55%	Linear
Adaptive encoding: Source-word sets	2 x 2	**64%**	**Constant**
-//-	3 x 3	**74%**	**Constant**

Table 1. Comparison of different modulation schemes for page oriented storage systems.

As the table shows, the adaptive encoding outperforms static and dynamic linear-time encoding. The parameters for the adaptive encoding were as follows: (a) 3 bits of source data were encoded for each available 2x2 code block, resulting in 3/4 encoding as the best case, and (b) 7 bits were encoded with 9 for 3x3 code block, resulting in 7/9 encoding as the best case. This leads to utilization upper bounds of 75% and 77% respectively. Apparently the upper bound was not reached because of the cases when no appropriate codeword was possible for the code block without creating destructive ISI, and the code block location had to be skipped.

5. CONCLUSIONS AND FUTURE WORK

Our experiments have shown that dynamic encoding modulation outperforms static modulation in terms of spatial overhead, providing up to 83% utilization of the media versus 45% for the static encoding. However, the static encoding can be done in constant time, while the dynamic encoding requires exponential time to reach maximum utilization, or linear time for a medium utilization of 55% as in a "either-neither" algorithm case.

There is a gap between a utilization of 45% for typical static encoding schemes and 83% for the dynamic full code enumeration approach. This suggests that there are good algorithms between these bounds with time complexity less than exponential. The described adaptive encoding modulation is one such algorithm.

A novel adaptive modulation encoding was presented, which uses multiple codeword sets for each source data. This encoding has constant time complexity and provides very competitive utilization of 74%, yielding only to the exponential time dynamic encoding. This is the best tradeoff found, which can be considered for practical implementation for page oriented optical storage systems.

In future we plan to investigate improvements of the adaptive algorithm, which would lead to better utilization without a major increase in time complexity. Also, we will investigate smart read-write head architectures with the optical sensor and transmitter array and processing fabric for the implementation of such algorithms. It is also possible to adapt the described dynamic and adaptive algorithms to volumetric 3D memory applications, taking into consideration inter-page symbol interference.

REFERENCES

1. Haichuan Zhang et al., "Multi-layer Optical Data Storage Based on Two-photon Recordable Fluorescent Disk Media", *Eighteenth IEEE Symposium on Mass Storage Systems*, 2001.
2. Dhawat E. Pansatiankul, Alexander A. Sawchuk, "Multidimensional modulation codes and error correction for page oriented optical data storage," *Optical data storage topical meeting 2001*, p.94-97, 2001.

3. Leo Selavo, Steven P. Levitan and Donald M. Chiarulli, “Real-time adaptive encoding for 3D optical memories,” SPIE Annual Meeting, Vol. 4459B-57, 2001.