Real-time adaptive encoding for 3D 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

ABSTRACT

There are a variety of factors that can limit the set of allowable code words that are useable on an optical memory block. In this paper, we will primarily consider inter-symbol interference (ISI) and the noise margins required to represent an individual bit. For example, code words must maintain a specific topological separation of "1" bits so that ISI does not raise the intensity of neighboring "0"'s above a pre-set threshold. Typically this is accomplished by a static encoding that uses a pre-selected set of code words based on these properties of the storage media and the optical system. Alternatively, our approach provides for a dynamic analysis of all data currently stored in the region surrounding a particular block and defines the allowable code words uniquely for each block.

We assume the existence of a "smart" read head that is capable of analyzing a page of data and calculating the allowable codes in real-time based on the actual data in the surrounding region. We use point-spread function based mathematical model for optical readout system to evaluate and carry out data encoding. Our experiments show 81% spatial utilization while recent publications present only 45% utilization.

Keywords: Page oriented memory, data encoding, dynamic data modulation, 2D data modulation.

1. INTRODUCTION

With the increasing popularity of multimedia, storage systems require larger capacity and access speed. There are a few volumetric optical storage systems in development, for instance two-photon volumetric memory [1].

Conventional magnetic and optical storage systems have data organized linearly in tracks. A limited number of tracks can be read or written in parallel. This is the approach used in most magnetic and magneto-optical disks, CDs and DVDs. Volumetric optical memories usually store data in blocks of two-dimensional pages. They outperform conventional data storage systems in two ways. First, they store information bits thorough the volume of the medium thus increasing the capacity. Secondly, the volumetric memories have parallel block-oriented readout, thus increasing the information throughput.

Data reliability is an essential requirement for any memory system. There are a few factors for optical memories that may corrupt the data while being recorded, retained or read out from the optical memory. For example, the readout system and the material noise, optical effects such as reflection, diffusion, and inter-pixel crosstalk due to additive noise in gaussian amplitude. Mechanical noise sources are also present, such as jitter due to positioning the readout mechanism or the optical media.

The data reliability is typically ensured by using error correcting codes and media-specific data modulation. Our research is focused on the modulation schemes. Error correcting codes have a common characteristic that reliability is achieved at the cost of code density. For example, magnetic storage devices use encoding schemes such as EFM (eight to fourteen) modulation, which results in low utilization, 43% of the media is lost to overhead. There are several data encoding schemes proposed for page-oriented memories, such as array codes [2] and crosstalk minimization [3,4]. However, just like their counterparts from magnetic storage they also suffer from very low utilization of the media.

One reason for utilization being so low in these codes is that code words are assigned statistically taking into account the worst-case scenario. In our approach we assign codes dynamically by taking advantage of the fact that the whole memory page is available to us for encoding and decoding. Specifically we are able to consider the actual data on regions of each page and compute the codes dynamically. We assign codes that maximize utilization while satisfying the system constraints. Therefore we assume intelligence in the read/write head with ability to process information at the source in real time. The result will be higher code densities and therefore higher media utilization while retaining the same reliability level of the data. We can also expect better performance due to parallel information processing.

In the next section we state the problem and give some analysis on data reliability. It is followed by description of our proposed approach to the problem and data encoding and decoding algorithms. The next section describes the memory model we used for simulations followed by simulation results. We conclude with comparison of our research to results to other recently published work in the area and an insight into future work.

2. PROBLEM STATEMENT

A typical setup for a two-photon optical memory [5] is shown in Figure 1. The data is read from memory in pages. The page addressing laser beam is shaped to a sheet of light and targeted to the memory page we want to read. The storage media is transparent to the laser light. The 3D pixels (voxels) that represent value ‘1’ are exited by the laser light and fluoresce. The optical readout system projects the image to the detector array and decoding system. Each pixel in the 2D page has a light intensity. The system uses an intensity threshold for each pixel value.

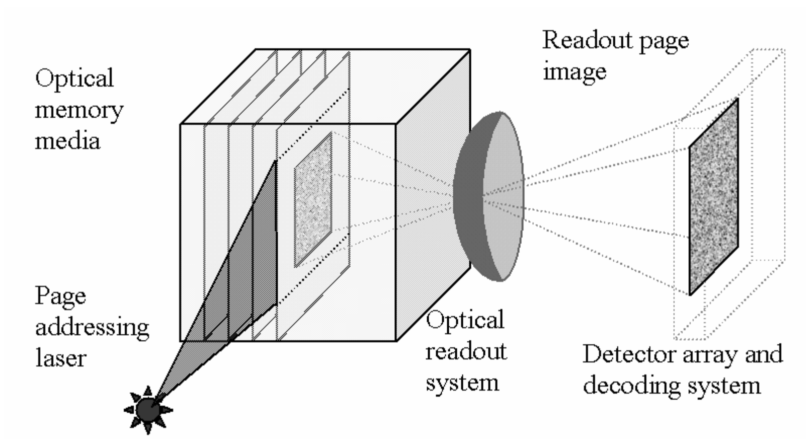


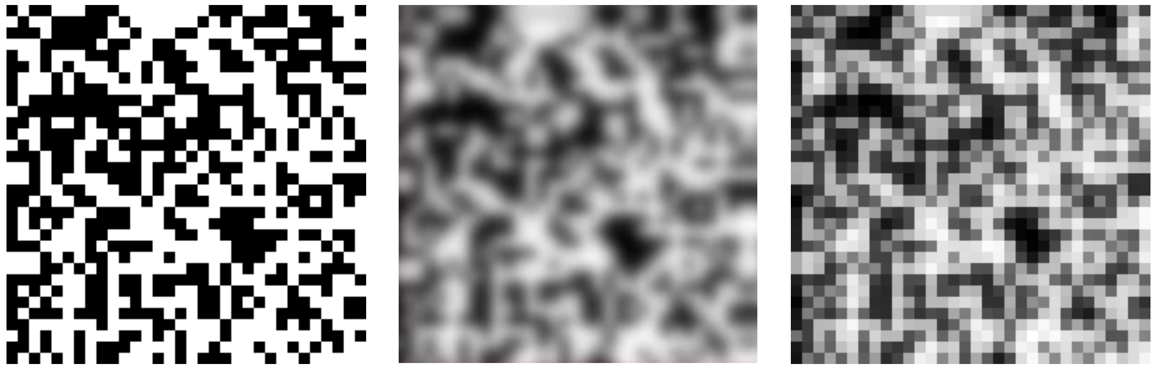
Figure 1: Two-photon optical memory

Several effects can result in corrupted information during readout:

- *Inter Symbol Interference.* The light output for each pixel almost always contributes some optical power to neighboring pixels on the detector array. This type of crosstalk should be minimized by the optical system, however a cluster of “1” pixels near a “0” pixel can produce enough crosstalk power such the “0” is detected in error as above threshold.

- *Weak ones*. If a ‘1’ pixel is surrounded by many ‘0’ pixels, and the readout threshold is fairly high, the ‘1’ pixel may be too week to read out as “high” bit.

- *Misalignment and jitter*. The readout head, laser or the media may not be aligned perfectly due to mechanical nature of addressing a page. Therefore the image may be slightly shifted and pixels may lack some intensity or contribute some intensity to their neighbors.



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

Figure 2: Raw data images

We will collectively refer to the combined effect of each of the three mechanisms as noise in the optical system. Figure 2 illustrates this with three versions of 32x32 bit image: a) the binary data written to memory; b) image as the detector array sees it and c) image as the detector array samples it. No matter what threshold you apply there is likely to be significant number of noise induced errors to the image between Figure 2b and Figure 2c.

In order to correctly detect ones and zeroes from the sampled image the relative intensity of each pixel must be clearly distinguishable according to value the pixel holds. To measure this relative intensity we gather statistics for each of the pixels on a page and plot a histogram of the number of pixels at each intensity. An example of one such histogram is shown in Figure 3. The two regions in this picture represent histograms for zeroes and ones. We observe that the regions overlap between intensities A and B. Thus in this region the same intensities may be interpreted as either ‘0’ or ‘1’, which results in unreliable data.

The goal of this research is to attain a histogram similar to Figure 4. The encoding should manipulate data so that the intensity values for ‘0’ and ‘1’ are sufficiently below or above our detection threshold T. This is achieved by ensuring that there is sufficient spatial separation between ones and zeroes on a 2D code block to limit the noise effects.

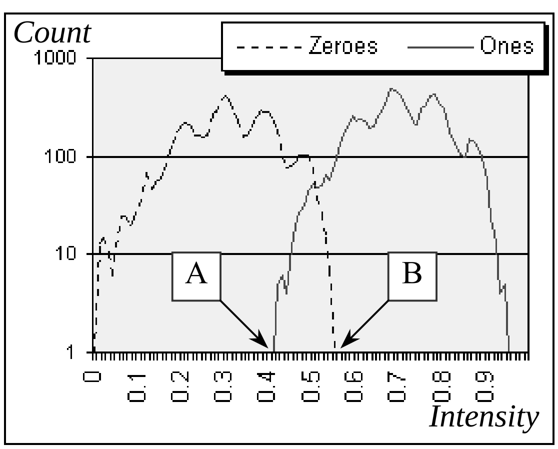


Figure 3: Raw data graph

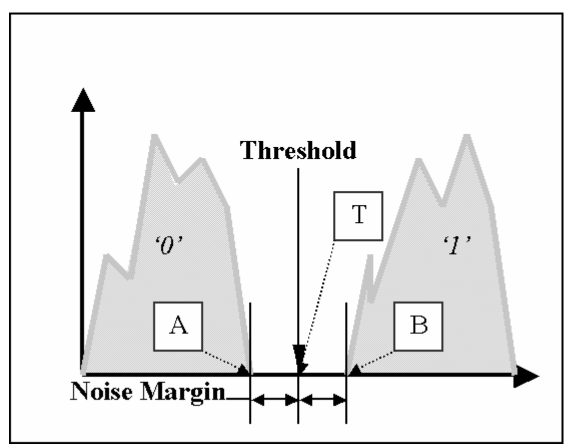


Figure 4: Noise margins

3. APPROACH

In this section we describe our mechanism for generating dynamic codes that minimize inter symbol interference (ISI) and other system noise. Specifically, ISI can cause a '0' pixel (that has many neighboring '1' pixels) to become "too bright". Similarly if zeroes surround a '1' pixel, it may become too weak to be correctly read as '1'. Therefore it will be a good idea to cluster '1' pixels by placing them in close proximity.

Since we cannot totally prevent system noise, we introduce "noise margin" as a parameter for our encoding. In this context the noise margin is a relative intensity above and below our detection threshold. Our encoding will provide that no pixels read between intensities A and B in the histogram of Figure 4. All "ones" will be encoded in such a way that their intensity should fall only above B, and "zeroes" only below A. The size of the noise margin is determined by the quality of the optics and memory system. Thus in realistic optical memory, if there will be additional intensity noise in the readout system, our encoding will tolerate it without loss of data correctness as long as it falls in the noise margin (A to T for zeroes and T to B for ones).

Figure 5 shows a section of data page during the encoding operation. The page is partitioned in rectangular code blocks (CB). In this example the light gray CB is being encoded. The dark gray CBs have been encoded previously, and the white blocks are assumed to be all zeroes (i.e. no light sources) since they will be subsequently encoded.

11001 00101 01010	01011 11010 11011	10111 10011 11010
10101 10001 11000	????? ????? ?????	00000 00000 00000
00000 00000 00000	00000 00000 00000	00000 00000 00000

Figure 5. Encoding scheme

We start with a blank page. The first CB to be encoded is the left-top most. There are no CBs to its left or above since it is beyond page boundary, therefore we assume zeroes there. We subsequently encode the whole line block by block and page line by line. Based on the surrounding data we compute intensity values and then valid codes for the CB to be encoded that will minimize cross talk and be within set noise margins. Once the valid code set is enumerated we select the k-th code from the code set depending on source data.

Decoding is done very similarly. We read digital data; compute intensities and valid code sets for each CB. The difference is that now we can use the code sets for reverse lookup of the encoded data from the read data. One advantage of the decoding is that it can be done in parallel for all CBs. The algorithm suggests reconfigurable mesh-connected parallel architectures, such as our optically reconfigurable field programmable gate array, developed in our laboratory [6]. Also, left-to-right approach for page fill is not necessarily the best. Filling page in a random or uniform order may actually yield better results.

There are several ways that we can generate valid codes for CBs. The most effective from the code density point of view is to enumerate all possible codes that do not violate noise margins. Unfortunately the worst time complexity for full enumeration approach is order of 2^{N-1} where N is the number of bits in a CB. The central focus of this research is to devise alternative algorithms that exploit topological properties and heuristics that are both efficient and well suited to parallel processing architectures.

4. MEMORY MODEL

We evaluate our encoding performance on two-photon volumetric optical memory model, described above. The evaluation of the encoding for other types of volumetric memories such as holography and spectral hole burning [1,6] is possible with adjustment of the model parameters and formulas to the chosen technology.

The optical memory storage is organized in 2D pages. The size and number of pages is a parameter of the model and we have run experiments for 64x64, 128x128 and 256x256 sized pages. To model the readout of the memory page the model computes an image that represents light intensities received by the readout detector array. Each of the light source intensities is computed to model light distribution assuming square aperture using the continuous point-spread function (PSF):

$$H(x, y) = 1/\sigma^2 \, \mathrm{sinc}^2(x/\sigma, y/\sigma)$$

Where σ is a parameter that models the degree of blur in the system, and

$$\mathrm{sinc}(x, y) = \sin(\pi x)/\pi x \;*\; \sin(\pi y)/\pi y$$

The optical memory page image is then generated using 2D convolution operator ● on binary data stored in the page and the PSF function.

$$r[i, j] = a[i, j] \bullet h[i, j] + noise[i, j]$$

where r[i, j] is the resulting intensity, a[i, j] is the source binary data, noise[i, j] is additive white Gaussian noise and h[i, j] is the effective discrete PSF computed as follows:

$$h[i, j] = \iint h(x, y)\, dxdy \qquad \textit{for each square [i, j] of PSF matrix.}$$

The surface integral is approximated with discrete over-sampling method using 64x64 samples per pixel. The binary output is determined by simple threshold on intensities. The degree of blur represents the quality of the optical system and is set at 1.4.

For optimization purposes we pre-compute discrete PSF matrix. We limited the dimensions of the matrix to 11x11 with the light source at the center. For every pixel beyond the dimensions of the matrix the additive intensity is assumed to be null, i.e. insignificant. This improved performance of the simulation system without significant loss of precision.

We ran experiments using a compressed zip file as a data source. It was chosen for the random distribution of 0 and 1 bits due to data compression. Thus we may encounter wide variety of bit patterns (or else the file could be further compressed).

5. SIMULATION RESULTS

In this section we first discuss the algorithm and then our experimental results. Our intention is to measure the improvement in utilization between conventional static encoding and our proposed dynamic encoding systems. We compute light intensity values and then build a valid code set for CB, which minimizes cross talk and complies with predefined noise margins.

The straightforward approach is to try all possible patterns of CB and *enumerate* only the valid ones. Then a bit-string from the source can be encoded using the enumeration as a lookup table. This takes about 2^N computations per CB where N is the number of pixels in a CB. One could pre-compute the lookup table to speed up the process, however the table is huge even for reasonable N values. Even though this approach is computationally very intensive, it gives us empirical upper bound for maximum utilization we could expect with the dynamic encoding approach.

An alternative way to generate a valid code is to evaluate CB *bit by bit* and decide if it can assume both '0' or '1' values. When this is true, we take a bit from the source data; copy it to this location and move to the next bit. Otherwise we write zero (nothing) to this pixel and go to the next. This method is much faster (order of N), however it does not generate codes as dense as the full enumeration method.

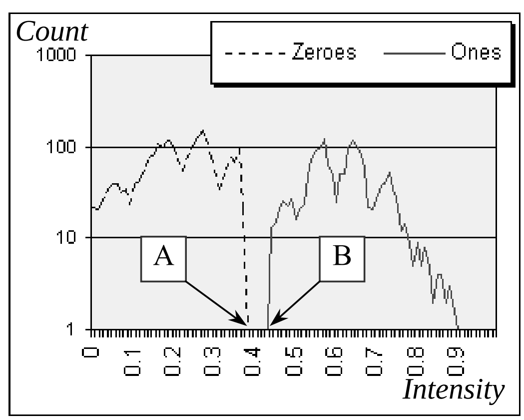


Figure 6: CB size 4x4 encoding results

We ran both algorithms for different sizes of CB, starting 1x2 up to 5x5 bits. The histogram in Figure 6 shows that the noise margins are maintained and there is no significant ISI. The utilization for the enumeration algorithm was 49-81% (as shown in Figure 7) with higher utilization for larger CBs. We also noticed that "tall" CBs do not perform as well as "wide" ones. For the same number of bits in

CB, square rectangles yielded better results than non-square.

There are two interesting features shown in Figure 7. First, the best results for utilization are over 80%. This is much higher than reported results for static encoding, namely 45% for 4/9 modulation scheme [3,4]. This suggests that there may be modulation algorithms that will give good utilization and relatively low time complexity. The second feature is the trend of the surface to flatten out at CB sizes larger than 4x4. This suggests that it is not needed to go into large CB sizes to achieve good utilization results. Thus better time performance can be achieved for encoding schemes with smaller code blocks.

Utilization results for various CB sizes.

Y \ X	1	2	3	4	5	6
1		49	62	65	68	72
2	49	66	71	74	74	75
3	61	72	74	77	81	76
4	61	71	76	78	81	
5	57	73	77	80	80	
6	64	74	80			

Threshold: 0.40
Noise margin: 0.03

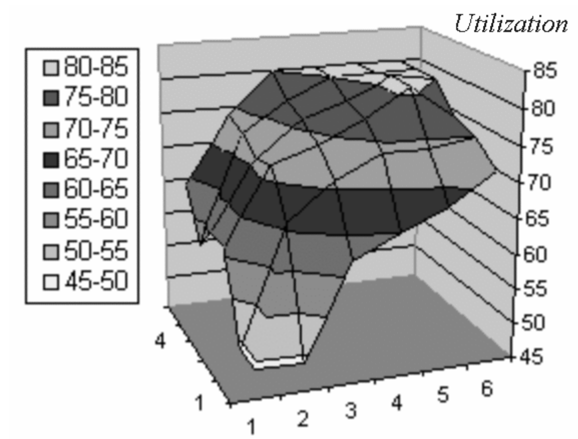


Figure 7: Full enumeration approach: Utilization vs. CB size

Returning to figure 6 the histogram for 4x4 CB version encoding simulation shows that noise margin is maintained and there are no ambiguously read bits, i.e. no destructive crosstalk. The code density was 78%. Figure 8 presents a) binary, b) detector and c) sampled images of memory page fragment after using 4x4 encoding. There are no single isolated dark dots surrounded by bright light as an effect of encoding and thus the crosstalk is minimized. Even the detector image b) seems to have better contrast the raw data image in figure 2b.

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

Figure 8: Encoded data images

The bit-by-bit approach gave good speed results while the utilization was between 55% and 60%. This algorithm slightly outperforms others recently published in terms of utilization. The algorithm's time complexity is linear to the number of bits in a code block. All code blocks can be decoded in parallel.

Another set of experiments determined how the dynamic encoding responds to parameter change. There are two parameters that influence the encoding: the threshold level and the noise margin. The experiments for the same input data set at different thresholds for 2x2 and 3x3 size code blocks generated the results in figure 9. Some threshold values yield better utilization as others. The best utilization was 83% for the threshold value of 0.45 units of normalized light intensity. Even relatively small 2x2 blocks gave good utilization of 70%.

More experiments for 2x2 and 3x3 size code blocks showed how well the encoding performs for different noise margins. In the ideal case maximally wide noise margin is desirable. Unfortunately the experiments in figure 10 show that the utilization drops quite steeply if we increase the noise margin over 0.04 units of normalized light intensity.

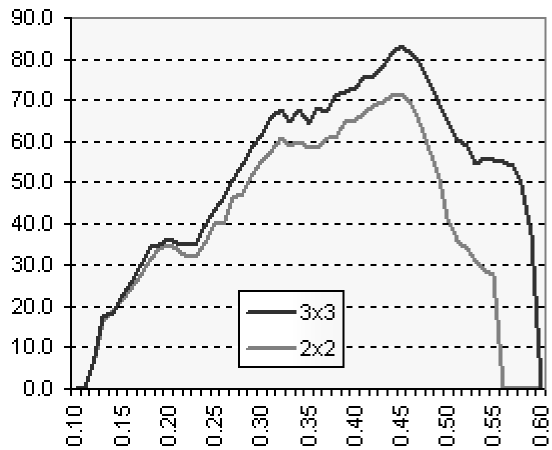


Figure 9: Utilization vs. Threshold (NM=.03)

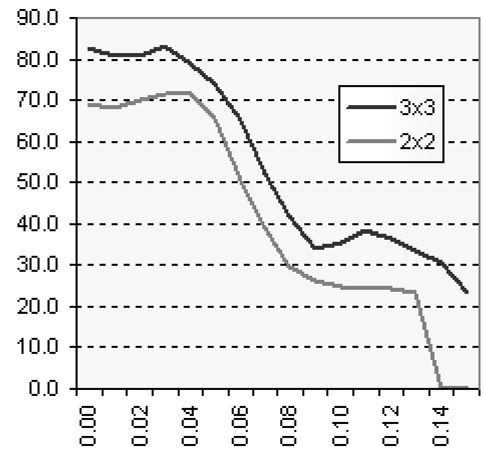


Figure 10: Utilization vs. Noise margin (T=.45)

6. CONCLUSIONS AND FUTURE WORK

Optical memories are good candidates to match requirements for large, fast and reliable data storage. We have presented dynamic data modulation encoding technique to attain the reliability with lower spatial overhead than conventional methods. Our experiments show utilization up to 81% while recent publications [3,4] present 4/9 encoding (45% utilization). The 2x2 version of the full enumeration approach seems to be a good candidate for implementation due to its utilization of 70% and relatively good performance (only 16 different codes to test for each code block).

Unfortunately the computational complexity of full pattern enumeration algorithm is higher than desired; therefore we need to search for alternative algorithms and heuristics to speed up the encoding and decoding. We have shown one such algorithm achieving 60% utilization. We will search for more algorithms with good utilization and reasonable time complexity in the future.

We assumed existence of “smart” optical readout head for our method implementation. We have previously presented a proof of concept of an optically reconfigurable FPGA [6], which may serve as a prototype for the readout head. To further improve the performance of the algorithms we plan to look into parallel implementations of the encoding and the decoding for mesh type architectures.

Since we are dealing with volumetric memories, inter-page crosstalk and other 3D noise effects may occur. For instance, neighboring page voxels may also get excited and radiate to some degree. We would like to adapt our algorithm to 3D environment, taking in account neighboring page information.

REFERENCES

1. Joshua L. Kann et al., "Mass Storage and retrieval at Rome Laboratory," *Proceedings of 5th NASA Goddard Mass Storage Systems and Technologies Conference*, pp. 389-406, 1996.
2. John F. Hutton, George A. Betzos, Maureen Schaffer, Pericles A. Mitkas, "Error correcting codes for page-oriented optical memories," *Proceedings of SPIE, Materials, Devices, and Systems for Optoelectronic Processing, Volume 2848*, 1996.
3. Dhawat E. Pansatiankul, Alexander A. Sawchuk, "Two-dimensional modulation codes for page-oriented optical data storage systems," *Proceedings of Optics in Computing*, p.79-81, 2001.
4. 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.
5. 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.
6. Leo Selavo, Steven P. Levitan and Donald M. Chiarulli, "An Optically Reconfigurable Field Programmable Gate Array," *Proceedings of Optics in Computing*, 1999.