SPIE Conference on *Advanced Optical Memories and Interfaces to Computer Systems II* July 1999 Paper **3802–06**
Proceedings of the SPIE, Vol. **3802,** pp. 18–29 (1999).

Coding tradeoffs for high–density holographic data storage

Geoffrey W. Burr[1] and Brian Marcus
IBM Almaden Research Center
650 Harry Road, San Jose, California 95120

1. ABSTRACT

We present an initial experimental evaluation of coding and signal processing tradeoffs in high–density holographic data storage. Block–based and low–pass modulation codes, predistortion of holographic pages during recording (pre–processing), and conventional equalization (post–processing) are compared using a few recorded holograms. The relative gain in number of stored holograms is obtained by measuring BER as a function of readout power; the effect on density is gauged by the size of the Fourier plane aperture in the holographic system. Results show that equalization provides a 20% density gain, and predistortion a 60% gain. The total improvement in density by combining small apertures with both of these signal processing options is >100% with an 8:12 strong balanced block code, a 6:9 lowpass/sparse code, and a parity thresholding technique with 9.1% overhead.

2. INTRODUCTION

By accessing the third dimension of storage media, volume holographic data storage can provide both high density and fast readout [1–3]. Thousands of holograms, each containing a page of data, are multiplexed into a common volume and accessed independently by Bragg–matched diffraction. As the size of this common volume is decreased, density (both volumetric and areal) increases. Since each data page can contain as many as a million pixels [4], reading one thousand pages a second results in a data rate of 1 Gbit/second.

As with any real-world data transmission or storage system, a holographic storage system is a noisy data channel. System optimization is a matter of getting data from input to output at the desired user bit-error-rate (user-BER), while maximizing the desirable properties of the system (density, capacity, and speed) and minimizing the undesirable properties (cost and complexity). To push these systems in density, the volume dedicated to a 'stack' of superimposed holograms must shrink. However, the small aperture through which the object beam enters the storage material causes diffraction: optical energy intended for a given detector pixel tends to spread to neighboring pixels. The resulting noise, when added to the background noise floor of the system, can cause errors in the retrieved digital data. If too much of this interpixel–crosstalk noise is present, then the signal levels must be increased (relative to the background noise) by decreasing the number of holograms, M. At some point, density reaches a maximum, because any further shrinking of the aperture would decrease density (by costing more in lost holograms than would be gained by smaller volume).

The effects of modulation [5] and error–correction [6] coding change this maximum by reducing the code–rate (number of digital bits per optical pixel) in return for improved decoding performance. In the same way that aperture size and the number of holograms M trade-off to maximize density, code-rate and M trade-off to maximize capacity. In contrast, signal post–processing techniques such as equalization [7,8] attempt to undo the blurring effects of the holographic data channel by deconvolution upon readout, while signal pre–processing [9] inverts the channel during the hologram recording process, affecting only bright pixels.

We have previously used simulation methods to explore the density tradeoffs associated with aperture size when using simple thresholding [10], and used experimental methods to explore the capacity tradeoffs of modulation codes at low–density, where interpixel crosstalk is insignificant [11]. In this paper, we combine these approaches to explore the effects of both block–based [5] and low–pass [12] modulation codes at high density, when used alone and in

[0]To contact G. W. Burr, Email: *burr@almaden.ibm.com*; Tel: (408) 927–1512; Fax: (408) 927–2100.

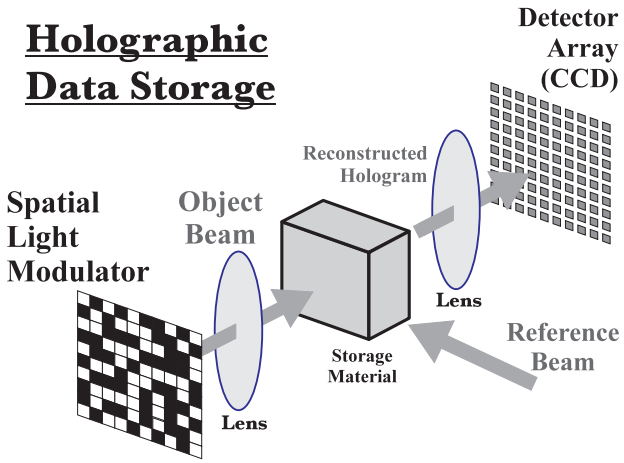

Figure 1: Components of a holographic storage system.

combination with zero–forcing equalization [8, 13] and predistortion [9].

3. NOISE SOURCES IN HOLOGRAPHIC DATA STORAGE

Figure 1 shows the basic components of a digital holographic storage system, in which a block of photosensitive storage material is surrounded by pixellated input and output components. To record data, one laser beam passes through the spatial light modulator (SLM) to pick up the input information (the 'object' beam), and meets a second coherent 'reference' beam in the material. A hologram is recorded in the index of refraction of the media. Re-illuminating the hologram with the original reference reconstructs a weak copy of the original information–bearing beam, which is then imaged onto a pixellated detector array. When each of the SLM pixels is accurately imaged to a detector pixel, the bright-or-dark state of a pixel during recording can be successfully detected at some later time during readout—hence, digital data has been stored and retrieved. If the hologram is stored in a thick material, the reconstruction will disappear when the readout beam is changed slightly in incident angle or wavelength. This new reference beam can then be used to store an independently accessible page of data. This has been used to store as many as 10,000 pages in the same $\sim 1cm^3$ block of material [14].

The basic noise trade-off in volume holography is between the finite dynamic range of the recording material and the fixed noise floor of the system. For instance, the electronic detection process at the camera tends to contribute the same amount of noise no matter how bright the hologram. However, as the number of holograms superimposed in the same volume (within the same 'stack' of holograms) increases, the amount of power diffracted into each hologram reconstruction and the resulting signal–to–noise ratio (SNR) decreases. The same problem tends to limit readout rate as well.

Even if all other noise sources are negligible, then there will be a certain hologram strength at which the SNR is inadequate for error–free detection. The number of detected electrons per pixel can be written as

$$n_{electrons} \propto M/\#^2 \, P_{readout} \, \frac{t_{readout}}{M^2 \, N_{pixels}}, \tag{1}$$

where M is the number of multiplexed holograms, N_{pixels} the number of pixels per hologram, $t_{readout}$ the integration time of the camera, $P_{readout}$ the power in the readout beam, and $M/\#$ is a material/system constant [15]. The storage capacity is MN_{pixel} and the readout rate is $N_{pixel}/t_{readout}$. (Storage density is then MN_{pixel} divided by the volume or area of each hologram 'stack'.) An increase in either capacity or readout rate leads to a decrease in the number of signal electrons [16]. As this signal strength approaches the number of noise electrons, the BER of the system will rise and the fidelity of the storage system will not meet the promised specifications.

While the constant noise floor is usually of primary importance, any additional noise sources will also use up part of the SNR budget. The presence of these additional noise sources causes the minimum acceptable number of signal electrons to get larger, reducing the capacity of the system. Noise sources in holographic storage include the following:

- *Change in the readout conditions.* This can occur, for instance, when the recording alters the properties of the

recording material, causing unwanted changes in the reference beam path between the time the hologram is recorded and the time it is reconstructed [17–19]. In some cases, the reference beam angle or wavelength can be tuned to optimize the diffraction efficiency and partially compensate for this effect [17].

- *The detector array doesn't line up with the array of pixels in the reconstructed hologram.* This includes errors in camera registration, rotation, focus, tilt and the magnification of the image, as well as any aberrations in the imaging system. Simple aberrations include spherical aberration (each spot gets uniformly bigger), coma (spots towards the outside of the array get stretched), and distortion (the imaged array of SLM pixels no longer falls on a square grid).

- *The detector is receiving undesired light,* either from light scattering off the storage material, crosstalk from other stored holograms (inter–page crosstalk [20]), or crosstalk between neighboring pixels of the same hologram (inter–pixel crosstalk [7,21]). Note that while crosstalk contributions scale with the strength of the holograms, the scattering depends only on readout power and the optical quality of the components. Inter–page crosstalk tends to build up as many closely–spaced reference beams are used within the same stack. Inter-pixel crosstalk is essentially diffraction–induced low–pass filtering of the pixellated data page, and occurs when an aperture is introduced to increase density by reducing the size of each stack within the material. The system then has a broad point-spread-function, and the sharply-defined input SLM pixels become blurred at the output detector array.

- *There are brightness variations across the detected image.* This can be a problem if a single threshold is used across the image to separate the pixels into bright and dark and assign binary values. These fluctuations can be caused by the SLM, the optical imaging, or the collimation and beam quality of the laser beams themselves. Such variations tend to be deterministic—they don't vary from hologram to hologram.

4. CODING AND SIGNAL PROCESSING

Given these many noise sources and the need to read back holograms and make bright–vs–dark distinctions with high fidelity, how can one maximize the desirable qualities of the system such as capacity and readout rate? The options we compare in this paper include:

1. Using a low–pass modulation code which avoids pixel combinations which are prone to inter–pixel crosstalk [12,21].

2. Using a decision scheme which produces fewer errors from the same SNR, either with adaptive thresholding [22], or by encoding at the SLM with a balanced modulation code [5].

3. Pre-processing at the spatial light modulator to reduce the deterministic variations which are reducing the SNR [9].

4. Post-processing at the detector array in order to remove a known point-spread-function estimation [7,8,13,23,24].

The following sections describe these options in more detail.

4.1 MODULATION CODING

A modulation code consists of an encoder and decoder which satisfy a desired modulation constraint. The encoder encodes each user data string of a fixed length m into a *coded rectangle* of fixed size $h \times w$, producing a constrained set of data pages that can be input to the holographic channel. The code rate of such an encoder is the fraction $m : hw$, indicating that hw SLM pixels represent m information bits. The coded rectangles are pasted together to exhaust a (coded) data page in such a way that the modulation constraint is satisfied not only within, but also across the rectangles.

A *block code* is a modulation code where the encoder is simply a one-to-one correspondence between m-bit data words and coded rectangles. Each rectangle is then independent of its neighbors. With the more general *finite-state encoder*, the coded rectangle is a function of the user data as well as an internal *state*. As the encoder produces each constrained rectangle, it also proceeds to a new internal state. Typically, coded rectanges are concatenated side by

Code/thresholding technique	r_{code}
Adaptive thresholding	1.0
Parity–assisted thresholding	0.90909
6:8 balanced block code	0.75
8:12 balanced finite-state code with enhanced minimum Hamming distance	0.666
6:9 combined lowpass/constant-weight finite-state code	0.666
8:9 lowpass finite-state code	
with adaptive thresholding	0.8888
with parity threshodling	0.80808

Table 1: Table of codes and thresholding techniques compared in this paper, along with the associated code rates.

side from left to right to form a long horizontal strip of height h; at the left end of each strip, the initial state is forced to a known value. In order to limit error propagation, we use a *sliding block decoder*: each coded rectangle B is decoded to a user data string by using the detected pixel values in rectangle B and a bounded number of coded rectangles to the right (*anticipation*) and left (*memory*), within the same horizontal strip. A *k-block decoder* is thus a sliding block decoder whose decoding window (i.e., memory + 1 + anticipation) consists of k coded rectangles.

Two modulation constraints are considered in this paper:

- a *lowpass* condition—the modulation constraint forbids certain patterns of high spatial frequency. Some simple lowpass codes were described in Reference [12], and shown in simulation to provide density improvements in Reference [10].
- a *constant-weight* condition—the proportion of ON pixels (binary 1's) is constant over all coded arrays. These constant weight codes can be either balanced (50% ON pixels [5]) or sparse (< 50% of the pixels are ON [25,26]).

Lowpass codes are desirable because they can mitigate the problem of inter-pixel interference at low aperture. Constant-weight codes are desirable because correlation detection [5] can be used, which alleviates problems with variations in intensity across the detected page. Sparse–coded pages contain fewer ON pixels, and thus reduce the optical exposure during recording while increasing the output power per pixel during readout.

In this paper, we consider several thresholding schemes and modulation codes—these are summarized in Table 1, along with their associated code rates, r_{code}.

The adaptive thresholding scheme uses a threshold which is continuously modified as the data page is detected [22]. A formula calculates the threshold using the pixel data from the last eight detected ON pixels, and the last four detected OFF pixels. The code rate is then 100%. In contrast to [22], the data page is detected in a normal raster fashion, not in an outward–expanding spiral.

The parity–assisted thresholding scheme [27,28] encodes user data directly into blocks over most of the page, and then uses the remainder of the page to encode the number of ON pixels in each block. In this paper, we used blocks of 9×9 pixels, and used the 6:8 code described below over the bottom 9.1% of the page to encode the number of ON pixels within each block. (Note that the parity value would be incorrectly encoded into 6 bits when the number of ON pixels per block is < 8 or > 72, but this is a very unlikely occurrence.)

The 6:8 and 8:12 codes are described in [5]. While the 6:8 code has higher code rate, the 8:12 code has enhanced robustness against noise because the minimum Hamming distance between codewords is larger.

The 8:9 code is a lowpass finite-state code that maps 8-bit data words to 3×3 coded arrays to fill in horizontal strips of height 3. This code forbids the pattern

$$\begin{array}{ccc} & 0 & \\ 0 & 1 & 0 \\ & 0 & \end{array}$$

from appearing anywhere in the coded page by forbidding the following patterns within each strip of height 3:

$$\begin{array}{ccc} & 0 & \\ 0 & 1 & 0 \\ & 0 & \end{array}, \qquad \begin{array}{ccc} 0 & 1 & 0 \\ & 0 & \\ & x & \end{array}, \qquad \text{and} \qquad \begin{array}{ccc} & & x \\ & 0 & \\ 0 & 1 & 0 \end{array},$$

where x indicates a 'don't care' bit. The code has a 3-state encoder and a 2-block decoder. It does not have a constant-weight feature, and so does not enable the use of correlation detection. Instead, for this code we have to use

adaptive thresholding or parity–assisted thresholding to detect (make binary decisions from the 8–bit camera data) and then decode each 3×3 data block to one user byte.

The 6:9 code satisfies the same lowpass constraint as the 8:9 code, but in addition satisfies a constant-weight condition: each coded array contains exactly three 0's and 6 1's, so a correlation detector is available. This code has a 3-state encoder and a 2-block decoder.

These particular codes were chosen because their small block sizes make the encoders and decoders simple to implement. It is possible to construct useful higher rate codes with low–pass or balanced properties, but only at the cost of larger block sizes and higher complexity.

4.2 PREDISTORTION

The 'predistortion' technique [9] is a recently developed technique for improving the SNR of data pages by removing deterministic (non–random) variations. Predistortion works by individually manipulating the recording exposure of each pixel on the spatial light modulator (SLM), either through control of exposure time or by relative pixel transmission (analog brightness level on the SLM). Any deterministic variations among the ON pixels, such as those created by fixed pattern noise, non-uniformity in the illuminated object beam, and even inter-pixel crosstalk can be suppressed, thus increasing SNR. Many of the spatial variations to be removed are present in an image transmitted with low power from the SLM directly to the detector array. Once the particular pattern of non-uniform brightness levels are obtained, the recording exposure for each pixel is simply calculated from the ratio between its current brightness value and the desired pixel brightness [9]. At low density, BER improvements of more than 15 orders of magnitude are possible [9] . More importantly, at high density, inter–pixel crosstalk (which is deterministic once the data page is encoded) can be suppressed and BER improvements from 10^{-4} to 10^{-12} have been shown [9].

4.3 EQUALIZATION

Signal post–processing is a common technique for improving SNR, extensively developed for the one–dimensional temporal channels found in conventional storage devices. These methods improve performance by manipulating the "analog" pixel values before they are quantized to binary 0 and 1 with thresholding or a modulation detector. Typically, these post-processing techniques perform a convolution with a small kernel designed to un-do the known broadening caused by the band-limiting optical channel. For the zero-forcing equalization [8] used in this paper, the kernel is derived by simply inverting the channel's spatial frequency response. While this removes the deterministic inter-pixel crosstalk, it can amplify random noise such as optical scatter or electronic detector noise. In addition, although the inter-pixel interference is occurring coherently (electric field amplitude), the detector pixels (and thus the reported camera counts) can only measure intensity. For small detector fill factors, this channel non-linearity can be compensated by taking the square root of the camera count values first, and then operating on these converted values with linear systems techniques [8, 13]. In this paper, the detector array we use has a large fill factor, so the camera values were used directly both in deriving the kernal and in convolving each received data page (the 'intensity' model). For each block of 25×25 detector pixels, we use a single 3×3 kernal. The kernal for each block of 625 pixels was derived by detecting a single isolated pixel in the center of the block, performing a 2–D FFT over the central 5×5 region, inverting in the Fourier domain, performing an inverse FFT, and then preserving the central 3×3 floating point numbers. The kernal is then ordered so that each pixel can be processed by correlation against its local neighborhood.

5. EXPERIMENTS

In this section, we describe the experimental apparatus and procedure for measuring density.

5.1 DEMON2 HOLOGRAPHIC TEST PLATFORM

The DEMON2 platform, shown in Figure 2, is a 90° geometry angle–multiplexed holographic storage platform. Light from a frequency–doubled diode–pumped Nd:YAG laser (λ=532nm) is expanded and split into reference and object beams. The object beam is apodized and further expanded, and directed through a polarizing beamsplitter onto the surface of a 1024×1024 liquid–crystal–on–silicon reflective SLM fabricated by IBM Yorktown through the HDSS program. The pixel pitch is δ=12.8μm and the areal fill factor 87%. Custom optics (effective focal length f=30mm) image the SLM pattern through the $LiNbO_3$:Fe storage material ($15\times15\times8mm^3$, **c**–axis at 45° in the horizontal plane, .02% Fe–doped, $\alpha \sim 0.8$ cm^{-1}) and onto the detector array (Dalsa CA–D4–1024). Detector pixel

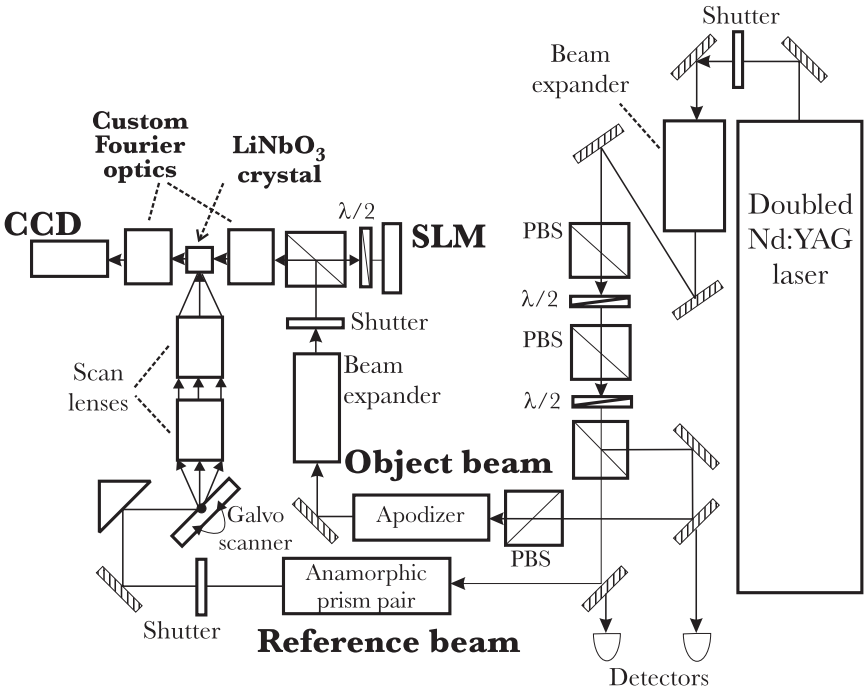

Figure 2: DEMON2 holographic storage platform.

pitch is 12μm with a fill factor of $> 95\%$, and the detected data page is pixel–matched 1:1 over the entire 1024×1024 pixels. The camera has a nominal frame rate of 41Hz, but integration times below and above 24.3 ms could be implemented. The camera response was calibrated using weak images of known power. The average pedestal was 5.4 camera counts (on its 8-bit scale), and the average gain such that 1 camera count per pixel corresponded to 1170 photons. Apertures were placed directly at the Fourier transform plane—the Nyquist aperture [10] for the spatial sampling rate on the SLM is

$$D_N = \frac{\lambda f}{\delta} = 1.246\text{mm} \tag{2}$$

The reference beam, 3.5mm in diameter, is directed off a galvanometric mirror positioner (Cambridge Instruments 602HC–6450), through a pair of scan lenses in a 4–F configuration, and onto the 8×15mm^2 face of the crystal. Shutters allow selection of object and reference beams independently, and half–wave plates and polarizing beamsplitters control total power and modulation depth.

Residual optical distortion (difference between the centers of the imaged SLM pixel and the CCD detector pixel) reaches approximately 0.3 pixels at the far corners of the page. In order to ensure that the ability of equalization to correct optical distortion did not affect the results of this paper, we only used the central 594×594 pixels to encode data. 1816 bad pixels within this region (caused by scratches on the SLM or dirt in the system) were remapped for transmission through the holographic system. When each page was displayed for hologram recording, the binary values of the bad pixels were duplicated in a rectangular buffer at the edge of the central region. When each page was received in the framegrabber, the data values from this remap buffer were swapped back to the bad pixel locations distributed across the data page. For this to work with equalization, the bad pixels were set to OFF after their binary data was saved in the buffer, and the convolutions performed on the received data page before swapping the pixel values from the remap buffer out to the bad pixels. The end result was that the equalization always operated on data that were detected by neighboring pixels, and the decoders always saw a contiguous page of good pixels. All the decoders were implemented in software. Each received data page (after equalization, if desired, and remapping) was processed twice: once with the received data values, and once with the known data page. The Hamming distance between the user data as decoded from received data and the known user data was computed and summed over the

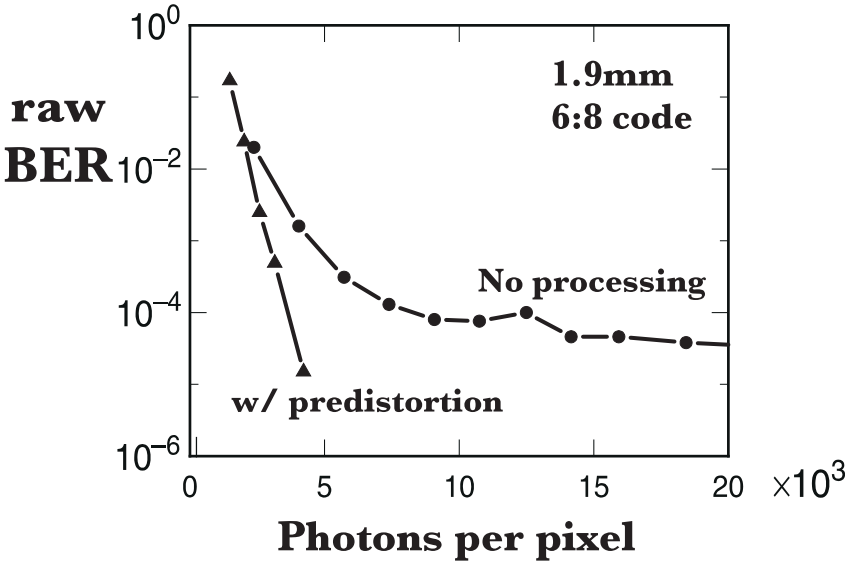

Figure 3: Intermediate experimental results: raw BER measured as a function of readout power (in photons per detector pixel)

page to calculate the number of raw bit errors.

5.2 EXPERIMENTAL PROCEDURE

For each combination of code and thresholding technique (9 all told), signal processing technique (4: none, equalization, predistortion, both), and aperture (from 1.1mm to 1.9mm), several holograms were recorded and read back. Each hologram was retrieved and decoded multiple times with decreasing camera integration times (from 24.4ms down to 0.25ms). All holograms were spaced by > 4 Bragg nulls to avoid inter–page crosstalk. For the measurements involving no signal processing and those with equalization, the same recorded hologram was read back twice. Similarly, the two thresholding techniques were implemented on the same detected data page. In contrast, two different holograms were recorded for the predistortion and 'both predistortion and equalization' cases.

Figure 3 shows two measured curves of raw–BER versus the average signal level in photons per pixel. Each curve rises to high BER at low input signal because of the fixed noise floor of the camera. At high signal levels, each curve tended to saturate at a constant BER (in the lower curve in Figure 3, this saturation level is well below 1 bit error per data page). This saturation is due to signal–dependent noise (noise contributions which scale with the signal power). Examples in our experiment included deterministic variations across the data page caused by uneven illumination of the SLM, and the inter–page crosstalk generated by the band–limiting aperture. The general effect of processing and stronger codes was to reduce this saturation level, thus decreasing the signal level at which the raw BER reached $\sim 10^{-3}$. As the aperture size was decreased, the BER floor rose for each code and signal processing technique, reaching 10% error for an aperture of 1.1mm (0.88 D_N).

5.3 ANALYSIS PROCEDURE

The capacity achieved by each code and signal processing technique depends on maintaining a low–BER at low signal levels. The tradeoff is between the number of holograms that can be stored (proportional to one over the square root of the signal level) and the ECC code rate. To optimize this tradeoff for each experimental curve like those shown in Figure 3, we plotted contours of constant capacity on the same axes of raw–BER vs. signal strength. Several such contours are shown in Figure 4, for a Reed–Solomon error–correcting code with 255 eight–bit symbols. As the number of bytes of overhead increases, the ECC code can correct a higher raw–BER at the cost of a lower code rate. In order to maintain the same overall capacity, the average signal level that can be tolerated must drop (thus allowing more holograms to be stored). The capacity numbers shown next to each curve are in arbitrary units (ECC code rate divided by the square root of signal in camera counts rather than photons per pixel).

For instance, three points are marked on the curve for "0.4" capacity. Point A corresponds to an ECC code

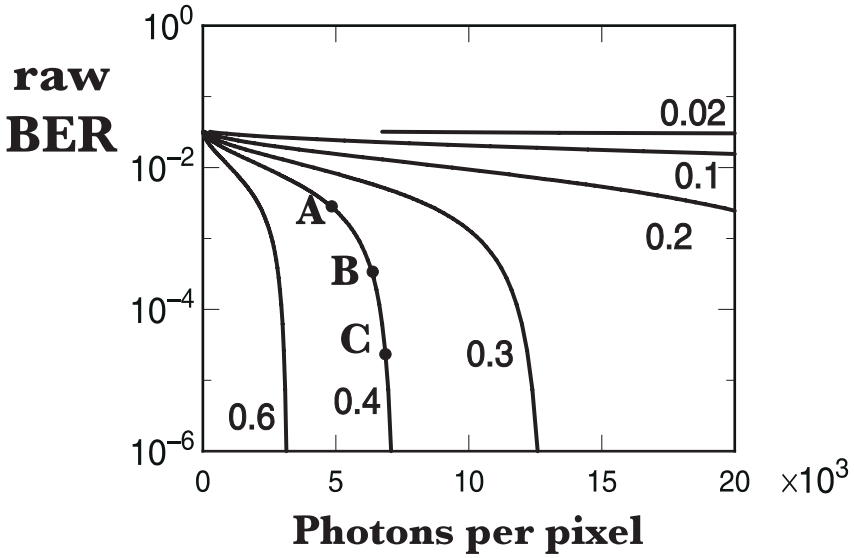

Figure 4: Curves of constant capacity with a 8-bit-per-symbol Reed–Solomon ECC code, plotted over the raw BER vs. detected photons axes.

capable of correcting t=39 byte errors with a code rate of 0.73, but requires a signal level of 3900 photons per pixel. In contrast, at point B the system expects a raw–BER of 5×10^{-4}, which it corrects with t=10 bytes of ECC, but requires only that holograms maintain 6200 photons per pixel. Similarly, point C corresponds to t=4 and 6900 photons per pixel. What these three points have in common is that the system has the same overall capacity of user data.

6. RESULTS AND DISCUSSION

To evaluate the density at the optimal choice of ECC coding, we merely plotted each measured data curve (like those in Figure 3) over the contours shown in Figure 4, and found the highest capacity curve which intersected the measured data. The total density was evaluated as

$$\text{Density (a.u.)} = \frac{1}{D^2}\, r_{\text{code}}\, \frac{r_{\text{ECC}}}{\sqrt{\text{signal}}}, \tag{3}$$

where D is the aperture size, r_{code} is the code rate of the modulation code (given in Table 1), and the ratio $(r_{\text{ECC}}/\sqrt{\text{signal}})$ is taken from the contour plots shown in Figure 4. For simplicity, all results were normalized to those measured with the 6:8 code at the 1.9mm aperture, without any signal processing.

The results of normalized density are shown in Figure 5. The 7 different coding options are shown as different curves in each of four plots, one for each signal processing option. The vertical axis is normalized areal density, and the horizontal axis is the aperture size. Note that in each instance, the density falls to zero at or just below the Nyquist aperture of 1.25mm. This shows that once the low–pass filter fails to pass all of the information in the data page, the BER rises very quickly and no amount of ECC can compensate.

Looking at Figure 5(a), the density gain from moving to a smaller aperture and retaining the 6:8 code is fairly small, approximately 17%. However, changing to the 8:12 code and using an aperture of 1.5mm in size results in a gain of 44%. Using equalization (Figure 5(b)) allows the system to push to smaller apertures such as 1.3mm, but does not appreciably increase the overall density. The use of predistortion alone (Figure 5(c)), or in combination with equalization (Figure 5(d)), however, provides a large improvement in density, topping out at 130% improvement for the 6:9 code.

Interesting trends in the results include:

- Vast improvement for the thresholding techniques are enabled by equalization and predistortion, because the signal processing improves the performance enough that the high code rate can become advantageous.

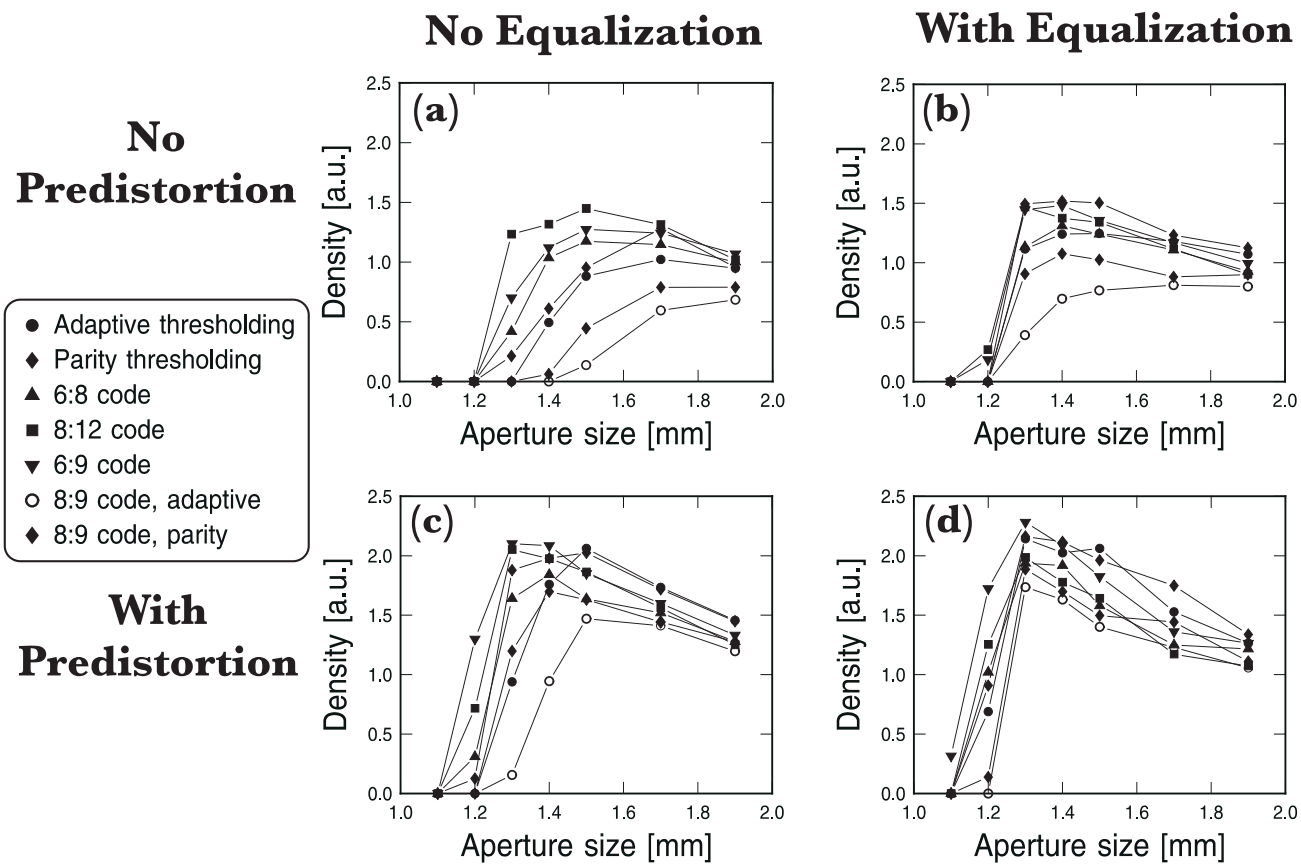

Figure 5: Density results for four different processing choices: (a) no equalization or predistortion, (b) equalization but no predistortion, (c) predistortion but no equalization, and (d) both equalization and predistortion.

- Equalization has the tendency to reduce the density achievable with the 8:12 code, while providing an improvement for most other codes. This is probably due to the fact that the the 8:12 code produces its high density by being able to work with data pages at extremely low signal levels. While the equalization reduces the inter–symbol interference between pixels, it also tends to amplify any random noise, thus increasing the signal level at which the 8:12 code can safely operate.

- The 6:9 code appears to be the winner for apertures smaller than the Nyquist aperture, and is the overall winner with an improvement of 130% with both equalization and predistortion. One disadvantage of using this particular code is that it is difficult to say how much of this performance is due to the correlation detector, how much to the low–pass character of the code, and how much to the sparseness. Certainly, the fact that the code is sparse already reduces the number of occurrences of the isolated–OFF pixel. This makes the code rate penalty associated with enforcing the lowpass condition much smaller than it would be for a balanced code. Further experiments will need to determine if the performance advantages are due to sparseness or the low–pass constraint.

- The 8:9 code performs extremely poorly, due to the lack of an associated modulation detector. As a result, every bit error caused by the thresholding scheme results in a block error for the subsequent 8:9 code, expanding the number of bit errors from one to four (on average). Some work might be done in the decoder to try to assign similar codewords to similar user byte sequences, which would reduce the average number of bit errors per block error. However, the improvement in decoded BER from such a move would most likely be fairly small. To be most fair, the ECC performance should probably be calibrated on the basis of the raw–*symbol*–error needed to maintain a given user–bit–error–rate; this would possibly reduce this problem.

- Although we do not show the results here, we also implemented and tested an inverted version of the 8:9 code, which forbids isolated ON pixels (an ON surrounded by four OFF pixels) instead of isolated OFF pixels.

Density (a.u.)	None	Equalization	Predistortion	Both
8:12	1.44	1.47	2.05	1.98 ($t = 14$)
Parity	1.28	1.49	1.88	2.16 ($t = 23$)
6:9	1.27	1.44	2.1	2.29 ($t = 11$)

Table 2: Best densities for the 8:12 code, parity thresholding, and the 6:9 codes, for the four different signal processing choices. The 'None' column results correspond to the 1.5 and 1.7mm apertures; all others to the 1.3mm aperture.

Contrary to the expectations of [10], this did not provide a noticeable improvement in density, but instead produced density curves quite similar to those shown here for the 8:9 code.

Numerical results for the best density with the 8:12, parity thresholding, and 6:9 codes are shown in Table 2. Also shown are number of bytes of error–correction associated with the best densities when using both predistortion and equalization, showing that while the codes require moderate error–correction, the parity thresholding technique requires fairly strong ECC.

7. FURTHER WORK

Since in this experiment, we are measuring holograms immediately after they are recorded, none of the noise sources associated with long exposure by the object beam are reflected in this data. These effects include noise gratings (written between the spatial frequency components of the object beam), broadening of selectivity curves and point–spread function caused by non–uniform erasure in the presence of absorption, and photovoltaic noise. The latter refers to the effects of the photovoltaic effect in $LiNbO_3$, which moves charge across the illuminated region to its edge during the recording process, changing the index of refraction at this boundary and leading to extra aberrations in transmitted images and stored holograms.

Since none of these effects are included, we would expect that these results probably undervalue the benefits of sparse codes (which reduce object beam exposure) and overvalue the benefits of predistortion (which increases total object beam exposure slightly but also includes more reference beam exposure and erasure). In addition, as we mentioned above, equalization can correct not only pixel broadening but also the effects of optical aberrations such as optical distortion.

To measure these effects, we plan to use the capacity measurement technique of [11] on a subset of these codes and signal processing options. This technique measures the dependence of raw-BER on initial recording exposure after long object beam exposures, and then uses this data to predict the number of holograms that can be recorded. To measure the number of holograms as a function of raw-BER would otherwise be an exhausting series of multiple hologram exposure experiments. Instead, a simple set of repeated measurements of a pair of holograms gives the relation between exposure time and raw-BER. Using the mathematics of a "flat–BER" recording schedule [11], this results in a unique value of M—the number of holograms that can be stored—for each target raw–BER.

In addition, we intend to introduce several sparse codes (to measure the advantages of sparseness and the lowpass constraints separately), and to combine the power of the 8:12 code with grayscale (more than two output signal levels) to create a robust, high–code–rate modulation code.

8. CONCLUSIONS

We have described and implemented an experimental procedure to compare seven different codes in combination with four signal processing options in terms of relative areal density. By reconstructing holograms stored with various band–limiting apertures located in the Fourier plane and varying camera integration time, we were able to measure the dependence of raw–BER on signal strength in photons per pixel. The optimal ECC solution for each measured data set was chosen by overlaying this raw data on a series of constant capacity curves, obtaining an overall density metric for each code, signal processing option, and aperture size.

Results showed that the presence of predistortion and equalization made high code rate more important than strong performance from the modulation decoder. This parallels the conclusions of [11] for low density holographic data storage. Using a typical block code of 6 bits to 8 pixels at a large aperture as a baseline, we found that a low–pass/sparse code encoding 6 bits in 9 pixels, a strong balanced block code encoding 8 bits in 12 pixels, and a parity–assisted thresholding scheme involving 9% overhead were able to provide a >100% improvement in areal density by using a smaller aperture and both predistortion pre–processing and zero–forcing equalization post–processing of data

pages.

9. ACKNOWLEDGEMENTS

We would like to thank the DEMON2 team (G. Burr, H. Coufal, C. Gollasch, J. A. Hoffnagle, C. M. Jefferson, M. Jurich, R. M. Macfarlane, R. M. Shelby) for designing, assembling, and implementing the hardware platform used in this paper. In addition, we acknowledge J. Ashley and B. Marcus for the modulation codes used in this paper, and C. M. Jefferson and the CSS model shop at Almaden for fabricating the precision apertures.

10. REFERENCES

[1] D. Psaltis and F. Mok. Holographic memories. *Scientific American*, 273(5):70, 1995.

[2] J. F. Heanue, M. C. Bashaw, and L. Hesselink. Volume holographic storage and retrieval of digital data. *Science*, 265:749, 1994.

[3] J. H. Hong, I. McMichael, T. Y. Chang, W. Christian, and E. G. Paek. Volume holographic memory systems: techniques and architectures. *Optical Engineering*, 34:2193–2203, 1995.

[4] R. M. Shelby, J. A. Hoffnagle, G. W. Burr, C. M. Jefferson, M.-P. Bernal, H. Coufal, R. K. Grygier, H. Günther, R. M. Macfarlane, and G. T. Sincerbox. Pixel–matched holographic data storage with megabit pages. *Optics Letters*, 22(19):1509–1511, 1997.

[5] G. W. Burr, J. Ashley, H. Coufal, R. K. Grygier, J. A. Hoffnagle, C. M. Jefferson, and B. Marcus. Modulation coding for pixel–matched holographic data storage. *Optics Letters*, 22(9):639–641, 1997.

[6] M. A. Neifeld and M. McDonald. Error correction for increasing the usable capacity of photorefractive memories. *Optics Letters*, 19:1483–1485, 1994.

[7] J. Heanue, K. Gurkan, and L. Hesselink. Signal detection for page–access optical memories with intersymbol interference. *Applied Optics*, 35:2431–2438, 1996.

[8] V. Vadde and B. V. K. Vijaya Kumar. Channel estimation and intra–page equalization for digital volume holographic data storage. In *Optical Data Storage 1997*, pages 250–255, 1997.

[9] G. W. Burr, H. Coufal, R. K. Grygier, J. A. Hoffnagle, and C. M. Jefferson. Noise reduction of page–oriented data storage by inverse filtering during recording. *Optics Letters*, 23(4):289–291, 1998.

[10] M.-P. Bernal, G. W. Burr, H. Coufal, and M. Quintanilla. Balancing inter–pixel crosstalk and thermal noise to optimize areal density in holographic storage systems. *Applied Optics*, 37:5377—5385, 1998.

[11] G. W. Burr, W.-C. Chou, M. A. Neifeld, H. Coufal, J. A. Hoffnagle, and C. M. Jefferson. Experimental evaluation of user capacity in holographic data storage systems. *Applied Optics*, 37:5431–5443, 1998.

[12] J. Ashley and B. Marcus. Two–dimensional lowpass filtering codes for holographic storage. *IEEE Transactions on Communications*, 46:724–727, 1998.

[13] V. Vadde and B. V. K. Vijaya Kumar. Channel modeling and estimation for intrapage equalization in pixel-matched volume holographic data storage. *Applied Optics*, 38(20):4374–4386, 1999.

[14] G. W. Burr, F. H. Mok, and D. Psaltis. Storage of 10,000 holograms in $LiNbO_3$:Fe. In *CLEO 1994*, page 9, 1994. paper CMB7.

[15] F. H. Mok, G. W. Burr, and D. Psaltis. System metric for holographic memory systems. *Optics Letters*, 21(12):896–898, 1996.

[16] K. Blotekjaer. Limitations on holographic storage capacity of photochromic and photorefractive media. *Applied Optics*, 18:57–67, 1979.

[17] R. DeVre, J. F. Heanue, K. Gürkan, and L. Hesselink. Transfer functions based on Bragg detuning effects for image–bearing holograms recorded in photorefractive crystals. *Journal of the Optical Society of America* **A**, 13(7):1331–1344, 1996.

[18] S. Campbell, S.-H. Lin, X. Yi, and P. Yeh. Absorption effects in photorefractive volume–holographic memory systems. i. beam depletion. *Journal of the Optical Society of America* **B**, 13(10):2209–2217, 1996.

[19] S. Campbell, S.-H. Lin, X. Yi, and P. Yeh. Absorption effects in photorefractive volume–holographic memory systems. ii. material heating. *Journal of the Optical Society of America* **B**, 13(10):2218–2228, 1996.

[20] C. Gu, J. Hong, I. McMichael, R. Saxena, and F. Mok. Cross–talk–limited storage capacity of volume holographic memory. *Journal of the Optical Society of America* **A**, 9(11):1–6, 1993.

[21] J. Hong, I. McMichael, and J. Ma. Influence of phase masks on cross-talk in holographic memory. *Optics Letters*, 21:1694–1696, 1996.

[22] X. A. Shen, A.-D. Nguyen, J. W. Perry, D. L. Huestis, and R. Kachru. Time–domain holographic digital memory. *Science*, 278:96–100, 1997.

[23] M. A. Neifeld, K. Chugg, and B. King. Parallel data detection in page–oriented optical memory. *Optics Letters*, 21:1481–1483, 1996.

[24] B. King and M. A. Neifeld. Parallel detection algorithm for page–oriented optical memories. *Applied Optics*, 37(26):6275—6298, 1998.

[25] M. A. Neifeld. Computer–generated holography for optical memory using sparse data words: capacity and error tolerance. *Applied Optics*, 32(26):5125–5134, 1993.

[26] A. Daiber. Sparse codes. Lecture at HDSS Coding Subgroup Meeting, April 1997.

[27] G. W. Burr, J. Ashley, B. Marcus, C. M. Jefferson, J. A. Hoffnagle, and H. Coufal. Optimizing the holographic digital data storage channel. In *Proceedings of SPIE: Advanced Optical Memories and Interfaces to Computer Storage*, volume 3468, pages 64–75, 1998.

[28] V. Vadde and B. V. K. Vijaya Kumar. Parity coding for page–oriented optical memories with intrapage intensity variations. *ol*, 24(8):546–548, 1999.