Near-Optimal Parallel Distributed Data Detection for Page-Oriented Optical Memories

Xiaopeng Chen, *Student Member, IEEE,* Keith M. Chugg, *Member, IEEE,* and Mark A. Neifeld, *Member, IEEE*

***Abstract*—Volume optical storage systems suffer from numerous sources of noise and interference, the effects of which can seriously degrade retrieved data fidelity and produce unacceptable bit-error rates (BER's). We examine the problem of reliable two-dimensional data retrieval in the context of recently developed soft-decision methods for iterative decoding. We describe a novel near-optimal algorithm in which each pixel on the page is treated as a starting point for a simple iterative procedure so that a highly parallel, locally connected, distributed computational model emerges whose operation is well suited to the page-oriented memory (POM) interface format. We study the use of our two-dimensional distributed data detection ($2D^4$) algorithm with both incoherent (linear) and coherent (nonlinear) finite-contrast POM channel models. We present BER results obtained using the $2D^4$ algorithm and compare these with three other typical methods [i.e., simple thresholding (THA), differential encoding (DC) and the decision feedback Viterbi algorithm (DFVA)]. The BER improvements are shown to have a direct impact on POM storage capacity and density and this impact is quantified for the special case of holographic POM. In a Rayleigh resolved holographic POM system with infinite contrast, we find that $2D^4$ offers capacity improvements of 84%, 56%, and 8% as compared with DC, THA, and DFVA respectively, with corresponding storage density gains of 85%, 26%, and 9%. In the case of finite contrast ($C = 4$), similar capacity improvements of 93%, 18%, and 4% produce similar density improvements of 98%, 21%, and 6%. Implementational issues associated with the realization of this new distributed detection algorithm are also discussed and parallel neural and focal plane strategies are considered. A 2 cm^2 $\lambda = 0.1$ μm digital VLSI real estate budget will support a 600 $\times$ 600 pixel $2D^4$ focal plane processor operating at 40 MHz with less than 1.7 W/cm^2 power dissipation.**

***Index Terms*— Distributed detection, iterative method, maximum likelihood detection, optical memories, parallel algorithm.**

I. Introduction

CONVENTIONAL optical storage systems utilize two-dimensional (2-D) planar media (e.g., optical disks) together with serial recording and retrieval techniques (i.e., a single optical head) to achieve high storage capacity, low cost, and a convenient removable format. The CD-ROM is based on this paradigm and has proven important for use as a software distribution medium. Multilayer methods (e.g., DVD) have recently been developed in which the storage capacity of conventional optical disks is increased through the use of multiple recording layers and improved optical resolution [1]–[2]. These quasi-planar storage systems represent a type of volume optical memory in which the third dimension is utilized with only coarse resolution. In contrast to these planar (2-D) and quasiplanar methods, three-dimensional (3-D) volume optical storage methods utilize all three spatial dimensions at high resolution, for the recording and retrieval of data. Recent progress has produced a number of promising material systems (e.g., photorefractives, photochromics, and photopolymers) as well as the required 2-D parallel supporting device technologies (e.g., CCD's and SLM's) [3]–[17]. Renewed interest in these 3-D storage methods has been fueled by the promise of very high capacity (>Tbyte), short access time (<1 μs), and high sustained data transfer rate (>1 Gb/s). The simultaneous combination of these three desirable characteristics is achieved in part by virtue of the 2-D parallel nature of the memory access paradigm, in which pages (e.g., 1000 $\times$ 1000 pixels) of data can be stored and recalled from each memory address. Because volume optical storage depends upon this page-oriented access to achieve its desirable performance characteristics, special attention must be paid to the 2-D nature of the memory interface and to fidelity issues related to the 2-D parallel storage and retrieval subsystems. In addition to these various performance related issues, the degree of commercial success of volume optical memory will of course depend upon additional considerations such as cost, convenience, and reliability.

A real-world optical storage system operating near its capacity, will be subject to numerous sources of noise and interference. These corrupting phenomena can arise from simple noise sources such as CCD thermal noise or from more complex sources of interference such as image dependent photovoltaic distortion [18]–[24]. These noise and interference effects can seriously corrupt the retrieved data thus producing unacceptably large bit-error rates (BER's). Conventional optical memories utilize a number of interface processing techniques both during storage and retrieval, to insure an acceptable level of data fidelity in the presence of these various corrupting phenomena. Conventional interface processing techniques include error correction coding/decoding, modulation coding/decoding, equalization, and sequence estimation. The utilization of these same methods within a volume optical storage environment however, will require that we consider the 2-D nature of the stored data and adapt the conventional sequential formulation of such meth-

Manuscript received March 31, 1998; revised July 16, 1998. This work has been supported in part by the National Science Foundation under Contract NCR-9616663 and Contract NCR-9628405.

X. Chen and K. M. Chugg are with the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA 90089-2565 USA.

M. A. Neifeld is with the Department of Electrical and Computer Engineering and The Optical Sciences Center, University of Arizona, Tucson, AZ 85721 USA.

Publisher Item Identifier S 1077-260X(98)08531-1.

ods accordingly. Considerable efforts have been undertaken to accomplish this adaptation to the page oriented memory (POM) environment. 2-D parallel methods of equalization, error correction, modulation coding, and sequence estimation have all been discussed [25]–[36]. It is possible however, to formulate a "best" (i.e., maximum likelihood) method for processing the parallel 2-D data retrieved from optical memory. Such a method will produce the lowest possible error rates. In this paper we will reexamine the problem of reliable 2-D data retrieval in the context of some recently developed soft-decision methods for iterative decoding [37]–[41]. These soft-decision iterative techniques will be shown to produce a convenient parallel distributed data detection algorithm whose performance closely approximates that of the best possible algorithm.

The basic problem of reliable page-oriented data detection is to decide on the most likely user data given the observed memory output page or pages. This problem is made difficult by the noise and interference that characterizes the optical storage channel and the resulting interpixel dependencies. A conceptually simple page-wise optimal method of accomplishing this task would involve the use of a look-up-table (LUT) containing all possible data pages together with their corresponding expected retrieved pages as corrupted by the storage and retrieval processes. An arbitrary retrieved page could be compared against elements in the LUT to decide on the original user data. This method is unfortunately not feasible because for a storage system with page size of $N \times N$ the required LUT would contain $2^{N \times N}$ entries! Traditional sequential (1-D) systems utilize a low complexity algorithm (i.e., the Viterbi algorithm (VA) [42]) to realize this optimal detection strategy; however, such an optimal sequential approach is unsuited to the page-oriented detection problem owing to the lack of a natural ordering of pixels in the 2-D array [43]. We note that there have been attempts to impose an ordering on the 2-D data page so that the Viterbi algorithm might be applied, and some reasonable success has been achieved with these methods [29]. It is also important to note that the optimal framework described above can be applied to received pages whose correlations are intentional (i.e., coded data pages) as well as unintentional (i.e., channel effects). The LUT can therefore include the mappings imposed via modulation and error correction codes as well as the channel; however, here we will focus on mitigating unwanted interference and noise without explicitly considering coding.

The maximum-likelihood data detection approach described above is computationally unreasonable for use within POM systems due to the highly parallel nature of the retrieved data pages. Various methods to combat this complexity are possible and recent attention has been paid to linear methods (e.g., equalization) and likelihood-based techniques (e.g., VA). A recent study of minimum-mean-squared-error (MMSE) equalizers for POM applications has found that these linear solutions can be useful for both coherent and incoherent POM [44]. The nonlinear nature of the coherent channel however, sets a limit to the benefit of linear equalization and it is expected that likelihood-based methods will be superior in the coherent case. Although this expectation is natural owing to the convenient mechanism through which nonlinear channel behavior may be captured within a likelihood-based algorithm, to date, the application of these methods has been restricted to the incoherent case [29], [37], [45].

In this paper, we develop a new 2-D likelihood-based algorithm and apply it to both incoherent and coherent POM channels. We present a highly parallel solution in which each pixel on the page is treated as a starting point for a simple iterative procedure. In this new algorithm there is no sense of a *sequence* associated with pixels on a page and the relevant soft information (likelihood) is computed in parallel at each pixel by collecting information from its nearest neighbors. In this way a highly parallel, locally connected, distributed computational model emerges whose operation is well suited to the POM interface format. In the next section we describe the POM channel models and the new distributed algorithm. Both coherent optical POM and incoherent POM are included. Section III presents the BER results obtained using our two-dimensional distributed data detection ($2D^4$) algorithm and several other typical algorithms. The effect of finite contrast is also quantified for both the coherent and incoherent models. The BER improvements presented in Section III have a direct impact on POM storage capacity and density and this impact is quantified in Section IV for the special case of holographic POM. Implementational issues associated with the realization of the new distributed detection algorithm are discussed in Section V. This section focuses on parallel neural and focal plane strategies and is followed by Section VI that provides insight concerning the implications of this research along with our conclusions and suggestions for future work.

II. $2D^4$ Algorithm

A. Two-Dimensional ISI/AWGN Channel Models

A volume optical memory system represents a complex data environment that includes many sources of noise and interference. Each type of POM (i.e., holographic, two-photon, hole-burning, etc.) will be characterized by its own dominant source(s) of data corruption, the specifics of which will depend on the exact implementation details of the system (e.g., focal lengths, CCD fill factor, exposure schedules, etc.). Despite the expected variations among various POM implementations, it is possible to identify two common factors that will contribute to reduced fidelity in such systems. The first such factor concerns the imaging behavior of the supporting optical system. Because such a system (coherent or incoherent) will impose a spatial-frequency lowpass response on the 2-D data array, the resulting optical blur will cause 2-D intersymbol interference (ISI) to appear in the retrieved page. In contrast to the serial (1-D) case, even small amounts of blur can produce significant effects in 2-D channels owing to the large number of nearest neighbors (i.e., 8) supported by the 2-D topology. In addition to a lowpass optical system, all POM implementations require some (finite temperature) detector array and supporting electronics from which thermal noise will arise, the magnitude of which will become larger as the memory system strives to operate at lower latency and higher sustained data transfer rate. It is of course

possible that shot noise or field noise will dominate some POM systems; however, the general behavior of such systems can be well understood through a study of the Gaussian noise limited case [44].

The POM model described above can be easily formulated in terms of a discrete 2D ISI channel whose output is corrupted by additive white Gaussian noise (AWGN). We represent the user data as a $N \times N$ pixel digital page of independent and identically distributed (iid) Bernoulli random variables $a(i,j)$, each taking on the real values $\alpha_0 \geq 0$ and $\alpha_1 > \alpha_0$ with equal probability to convey a binary digit. These signal levels represent the "on" and "off" states of a spatial light modulator (SLM) pixel with $\alpha_0 \neq 0$ representing the effect of finite contrast. At the output of each detector we define $w(i,j)$ to be an AWGN process with variance σ_w^2. For the case of an incoherent optical system we interpret α_0 and α_1 as intensity quantities and we obtain an expression for the memory output (observation) at pixel (i,j)

$$z_I(i,j) = \sum_{l,m=-L}^{L} a(i-l,j-m)f(l,m) + w(i,j) = x_I(i,j) + w(i,j) \quad (1)$$

$$f(l,m) = \int_{-\Delta/2}^{\Delta/2}\int_{-\Delta/2}^{\Delta/2} h_{l,m}(x,y)\,dx\,dy \quad (2)$$

where $2L+1$ represents the number of pixels over which the channel blur function extends, Δ represents the spatial dimension of the (square) CCD detector over which the received intensity is integrated, and $h_{l,m}(x,y)$ represents the incoherent point spread function (PSF) of the optical system centered at the (l,m) CCD pixel position (i.e., $h_{l,m}(x,y) = h(x+l\Delta, y+m\Delta)$ where $h(x,y)$ is the on-axis incoherent PSF). Note that the observation $z_I(i,j)$ in (1) is linear in the data $a(i,j)$.

A coherent optical system gives rise to a somewhat more complicated discrete 2-D ISI channel in which the observation is a quadratic function of the user data. For a coherent system, we interpret α_0 and α_1 as field quantities and the memory output at pixel (i,j) becomes

$$z_C(i,j) = \sum_{l,m,p,q=-L}^{L} [a(i-l,j-m)a(i-p,j-q)] \cdot R_h(l,m;p,q) + w(i,j) = x_C(i,j) + w(i,j) \quad (3)$$

$$R_h(l,m;p,q) = \int_{-\Delta/2}^{\Delta/2}\int_{-\Delta/2}^{\Delta/2} h_{l,m}(x,y)h_{p,q}^*(x,y)\,dx\,dy \quad (4)$$

where in this case $h_{l,m}(x,y)$ and $h_{p,q}(x,y)$ represent two copies of the coherent PSF centered on pixels (l,m) and (p,q), respectively, and the * superscript indicates the complex-conjugate operation.

B. A 2D[4] Algorithm

As described in the introduction, an optimal algorithm would require the storage of 2^{N^2} page likelihoods. Even within the framework of this conceptually simple, yet prohibitively complex, processing, there are algorithmic variations. One may consider maximum *a posteriori* (MAP) detection[1] of the *bit* $a(i,j)$, or the *page*[2] $A = \{a(i,j)\}$ based on the observation Z (i.e., Z_I in the incoherent system and Z_C in the coherent system). The former minimizes the BER, while the latter minimizes the probability of a page error. In practice, either criterion is logical and the resulting algorithms provide good performance under either measure [46]. Since an efficient implementation of either approach is unknown, the difference is conceptual; however, we will present algorithms for approximating both approaches. Under the current assumption of equal *a priori* statistics (i.e., $P[a(i,j)=\alpha_0] = P[a(i,j)=\alpha_1]$), the MAP rule reduces to the maximum likelihood (ML) rule. Thus, we will refer to the BER-minimizing algorithm as ML bit detection (MLBD) and the page-error-minimizing algorithm as ML page detection (MLPD). Our motivation for distinguishing between MLBD and MLPD is that an intuitive explanation of algorithms aimed at approximating MLBD is simpler while numerical procedures that approximate MLPD are computationally simpler. Thus, in this section we briefly describe a distributed, iterative algorithm for approximating MLBD and in Appendix I-A we describe the analogous processing for an approximate MLPD algorithm.

The MLBD algorithm selects the hypothesized data bit $\hat{a}(i,j)$ that maximizes the bit likelihood $P[Z|\hat{a}(i,j)]$ which is performed for each location in the data page. To compute this exactly, one would need to average the page likelihood $P[Z|A]$ over all pages A consistent with the hypothesized value of $a(i,j)$ (i.e., 2^{N^2-1} entries of a LUT). Such a task is computationally unreasonable; however, computing the bit likelihood based on a subset of the observation page is computationally feasible. For example, first consider computation of the likelihood of $a(i,j)$ based on the single pixel observation $z(i,j)$. This can be accomplished by combining a finite number of terms. Specifically, let the *support* S_{ij} of the observation $z(i,j)$ be the subset of input data such that $P[z(i,j)|A] = P[z(i,j)|S_{ij}]$. For the 2-D ISI models described above, the support of $z(i,j)$ is the $(2L+1)\times(2L+1)$ array of inputs centered at $a(i,j)$. Also, define the *neighborhood* of $a(i,j)$ to be $N_{ij} = S_{ij} - \{a(i,j)\}$. Notice that the neighborhood N_{ij} can take on one of $2^{[(2L+1)^2-1]}$ values in the binary-valued space $\Omega = \{0,1\}^{[(2L+1)^2-1]}$. Thus, the likelihood is computed via

$$P[z(i,j)|a(i,j)] = E_{N_{ij}}\{P[z(i,j)|a(i,j),N_{ij}]\} \quad (5)$$

$$= \sum_{N_{ij}\in\Omega} P[z(i,j)|S_{ij}]P[N_{ij}] \quad (6)$$

$$= \sum_{N_{ij}\in\Omega} P[z(i,j)|S_{ij}]\left(\prod_{a(l,m)\in N_{ij}} P[a(l,m)]\right) \quad (7)$$

[1] The MAP decision criterion yields minimum error probabilities. The ML decision criterion is equivalent to MAP when the *a priori* probabilities of 0 and 1 levels are equal.

[2] Upper case variables are used to denote 2-D arrays of corresponding lower case quantities.

where the $P[z(i,j)|S_{ij}]$ is completely characterized by the channel and the last step follows from the assumption of independent data bits.

The likelihood $P[z(i,j)|a(i,j)]$ can be thresholded to make an optimal decision based only on $z(i,j)$. However, a significantly more reliable decision will result from a likelihood based on other observations. A method of combining the likelihoods $P[z(l,m)|a(l,m)]$ to approximate this goal is motivated by the relation in (7). Specifically, note that $P[z(i,j)|a(i,j)]$ is proportional to the *a posteriori* probability $P[a(i,j)|z(i,j)]$ because of the equal priors assumptions. Thus we propose to infuse likelihood information from the neighboring pixels by using (7) with $P[a(l,m)]$ replaced by $P[z(l,m)|a(l,m)]$. Although this assignment appears nonintuitive, the approach is commonly referred to as "propagation of the extrinsic information" and is widely used for turbo decoding [39]–[40]. More specifically, based on (7), we suggest the following iterative likelihood propagation rule

$$L_U^{(k)}[a(i,j)] = \sum_{N_{ij}\in\Omega} C_{ij}[S_{ij}] \prod_{a(l,m)\in N_{ij}} L^{(k-1)}[a(l,m)] \quad \text{(Likelihood Exchange)} \tag{8}$$

$$L^{(k)}[a(i,j)] = (1-\beta)L^{(k-1)}[a(i,j)] + \beta L_U^{(k)}[a(i,j)] \quad \text{(Likelihood Filtering)} \tag{9}$$

where $L^{(k)}$ is the likelihood at kth iteration, and $C_{ij}[S_{ij}]$ is a set of $2^{(2L+1)^2}$ combining coefficients determined at initialization for each bit location (i,j). We refer to this iterative rule as the 2D^4 algorithm. The exchange (combining) in (8) is not optimal because the *a posteriori* statistics of the data are *not* independent. However, the intuitive notion is that, beginning from $L^{(0)}[a(l,m)]$ based on some small portion of the observation page, the exchange mechanism will approach the desired $P[Z|a(i,j)]$ for large k. For example, one may choose to set $C_{ij}[S_{ij}] = P[z(i,j)|S_{ij}]$ and $L^{(0)}[a(l,m)] = P[a(l,m)]$, which would be based exactly on the relation in (7). Many iterative algorithms exhibit a tradeoff between speed of convergence and accuracy. This motivates the filtering process at each pixel as defined in (9). The filtering parameter $\beta \in (0,1]$ determines the bandwidth of a single-pole low-pass filter which rejects abrupt changes in the updated likelihood $L_U^{(k)}[a(i,j)]$. Specifically, the smaller the value of β, the greater the rejection of fluctuations in $L_U^{(k)}[a(i,j)]$, which is expected to increase the time to convergence. One iteration of the algorithm is defined by the combination of likelihood exchange with neighboring bit locations and the filtering update, which takes place independently at each pixel (i.e., k measures the iteration number). The iterations may be continued until some stopping criterion is met (e.g., $|L^{(k)}[a(l,m)] - L^{(k-1)}[a(l,m)]| < \epsilon$) or for a fixed number of iterations. In either case, if stopping occurs after K iterations, the decisions are made according to: $\hat{a}(i,j) = \alpha_1$ if $L^{(K)}[a(i,j) = \alpha_1] > L^{(K)}[a(i,j) = \alpha_0]$, otherwise $\hat{a}(i,j) = \alpha_0$.

The preceding description is intended to provide ample motivation for the family of 2D^4 algorithms and we will refer to the steps of the algorithm as defined in (8) and (9). However, the algorithm used for simulations differed in two ways. First, we used an algorithm inspired by MLPD rather than MLBD. This corresponds to replacing the expectation in (5) by a maximization, which has certain computational advantages. This may be interpreted as replacing the (average) likelihood $L[a(i,j)]$ by a generalized likelihood [47] $G[a(i,j)]$; so we will refer to L-2D^4 and G-2D^4 algorithms when the distinction is relevant. Second, an effort is made to set the combining coefficients based on a larger portion of the observation page. These details are discussed in Appendix I-A. Also, a method of complexity reduction is investigated in Section III-B.

The knowledge about the channel is incorporated only through the combining coefficients $C_{ij}[S_{ij}]$ in the update rule (8). An important feature of the 2D^4 algorithm is its ability to include channel effects of arbitrary complexity within these probabilities. This algorithm is therefore well suited to the nonlinear channel models that characterize coherent optical storage systems. It is often the case for experimental POM systems, that an explicit channel model is unavailable owing to the complexity associated with the various relevant noise and interference sources. However, an explicit model is not required by the 2D^4 algorithm and experimental measurements can be used to estimate $C_{ij}[S_{ij}]$ (e.g., $P[z(i,j)|S_{ij}]$) for use in likelihood update rule (8). Also, it is conceptually straightforward to compute these probabilities so as to also include the effects of modulation and/or error control codes.

In summary, under the POM models described above, a bit-wise optimal algorithm would require the use of an entire page of observations in order to decide on the digital value of a single pixel. Such a procedure is not computationally feasible and the algorithm described in this section has motivated the use of a local neighborhood of observations as a substitute for the use of an entire page. This concept defines one iteration of the 2D^4 algorithm. The proposed algorithm is iterative, however, with each pixel using the instantaneous values of its neighboring likelihoods to update its own likelihood estimates. This iterative process can be viewed as a mechanism through which likelihood information is *propagated* in 2-D throughout the retrieved data page. During each successive iteration, each pixel is influenced by a growing number of observations. This likelihood propagation process is the key to facilitate a highly parallel 2-D detection algorithm. In Section V, we will use a neural network analogy to further clarify the operational principals of the proposed 2D^4 method. In the next section we quantify the performance of this algorithm.

III. BER Results

In this section, we examine the BER performance of the 2D^4 algorithm. Monte Carlo simulations were run for two incoherent channels and two coherent channels with a page size of $N = 128$. Under both kinds of channels, BER's are plotted against the inverse noise variance $(1/\sigma_w^2)$ in decibels (i.e., INV $= 10\log_{10}(1/\sigma_w^2)$) with the channels normalized to unit optical intensity. The two incoherent channels are representative of a POM system operating near and beyond the Sparrow resolution limit with a Gaussian blur. Specifically, $h_I(x,y) = (1/\sqrt{2\pi\sigma_b^2})\exp\,[-(x^2+y^2)/2\sigma_b^2]$ is used to compute $f(i,j)$, which was then truncated to $L = 2$ (5×5) and renormalized so that $\Sigma_{(i,j)}\, f(i,j) = 1$. Incoherent Channels A

and B were so defined for $\sigma_b = 0.45$ and $\sigma_b = 0.623$, respectively. For the coherent case, we consider an optical imaging system containing a square Fourier plane aperture of length $1/W$ on each side. The coherent impulse response of such an imaging system is $h_C(x,y) = \mathrm{sinc}(x/W)\mathrm{sinc}(y/W)/W$, where W measures the degree of blur experienced by images passing through the system. The coherent channels were computed by determining $R_h(i,j;l,m)$, truncating to $L = 2$ (5×5) and renormalized so that $\Sigma_{(i,j)}\ R_h(i,j;i,j) = 1$. Coherent Channels A and B, were computed in this manner for $W = 0.4$ and $W = 1.0$, respectively. Although the forward channel models have been simulated using 5×5 support, the computational complexity of the 2D^4 algorithm with $L = 2$ is unacceptable, requiring 2^{25} neighborhood configurations per pixel. In order to reduce this computational burden we will simulate the required 2D^4 interface processing using a reduced $L = 1$ (3×3) support. The omitted entries on the boundary (i.e., 16 pixels) are assigned a fixed input value, i.e., $(\alpha_0 + \alpha_1)/2$. This mismatch will cause a slight reduction in overall performance (i.e., more pronounced for larger blurs); however, the results will more accurately represent the performance of the 2D^4 algorithm within an actual realtime POM system environment.

All simulations are based on the G-2D^4 algorithm with combining coefficients and initialization as described in Appendix I-A. We compare the 2D^4 algorithm with three other approaches: the simple pixel-by-pixel threshold decision algorithm (THA), the decision feedback Viterbi algorithm (DFVA) [29] and the use of differential encoding/decoding (DC) [48]. The THA makes decisions by thresholding on the single observation: $\hat{a}(i,j) = \alpha_1$ if $z(i,j) \geq \overline{z}$; otherwise $\hat{a}(i,j) = \alpha_0$, where $\overline{z}$ is the mean value of $z(i,j)$. The DFVA runs the VA row-by-row with decisions fedback from previous rows. Although in [29] the DFVA is only applied to incoherent channels, it is straightforward to adapt the DFVA for a coherent channel model. Due to the same complexity constraint as the 2D^4, the DFVA is also simulated with a reduced 3×3 support. The assignment of a fixed input value on the ignored edge entries, as used with the 2D^4 algorithm, is also applied. Using DC combats the ISI by using two pixels in the page to record a single bit. Specifically, DC records a binary "1" by "10" and a binary "0" by "01." For example, an input bit $b(i,j) = 1$ sets $a(2i,j) = \alpha_1$ and $a(2i+1,j) = \alpha_0$. At the detector, the decision is made according to the simple rule: $\hat{b}(i,j) = 1$ if $z(2i,j) \geq z(2i+1,j)$; otherwise $\hat{b}(i,j) = 0$. Thus, the use of DC results in a 50% decrease in the number of bits per page. This fact does not impact the BER results presented in this section, but is included when translating the BER results to storage capacity/density in Section IV.

Before presenting the BER comparisons for the incoherent and coherent Channels A and B, we demonstrate the effects of β and complexity reduction techniques.

A. Convergence Properties

The convergence rate of the iterative procedure in (8) and (9) determines the time complexity of the 2D^4 algorithm. Fixing INV $= 27$ dB, the BER curves for various values of β are plotted for the incoherent Channel B in Fig. 1. The convergence rate is dependent on the value of β. More specifically, when $\beta \leq 0.5$, the convergence is slower with decreasing β, although the performance after convergence is similar. For example, for $\beta = 0.1$, convergence is achieved for $k > 50$, but for $\beta = 0.5$, convergence occurs for $k > 15$. This behavior is reversed for $\beta \geq 0.5$—i.e., the performance, not the convergence rate is more sensitive to varying β. For $\beta \geq 0.5$, convergence occurs at approximately 15 iterations with the steady-state BER varying by up to 25% (see the insert in Fig. 1). Also, oscillations in the BER results are observed after a certain number of iterations, especially when $\beta \geq 0.9$. Based on the results in Fig. 1, we use $\beta = 0.3$ in the following simulations as a reasonable compromise. For this choice, no more than 20 iterations are required; even fewer for less severe ISI channels (i.e., see Sections III-C and III-D).

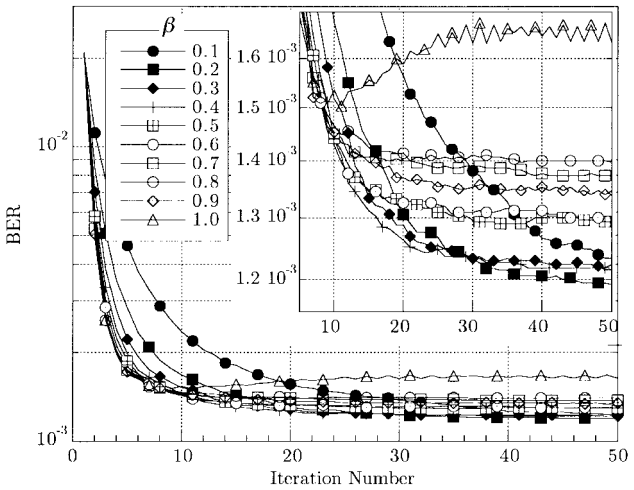


Fig. 1. The convergence properties of the 2D^4 algorithm parametered by various values of β. The insert is a magnified version of the lower portion. The incoherent channel B is simulated at a INV of 27 dB.

B. Effect of Reduced Connection Complexity

The main computation task within the 2D^4 algorithm, comes from the update rule (8), the complexity of which is determined by the size of the neighborhood N_{ij}. For most low-pass channels, the ISI channel coefficients decrease in magnitude rapidly at the edge of the neighborhood. For example, even in the relatively severe incoherent channel B, $f(1,1)/f(0,0) = 0.0993$ is relatively small. Thus, we consider omitting some entries of N_{ij} in the likelihood update of (8) to simplify the computation. In the 2D^4 algorithm, the connection complexity is reduced by 50% for each entry that is omitted. However, it is expected that this complexity reduction will be realized at the cost of performance and/or convergence rate because the likelihood exchange-update process is weakened. The connection complexity is an important factor in the implementation of 2D^4 algorithms (see more discussion in Section V).

We define the *connection mask* of the 2D^4 algorithms as the the structure of the connectivity used during the update step. The "fully connected" connection mask suggested by (8) is a special case. For the 3×3 processing used in this paper we

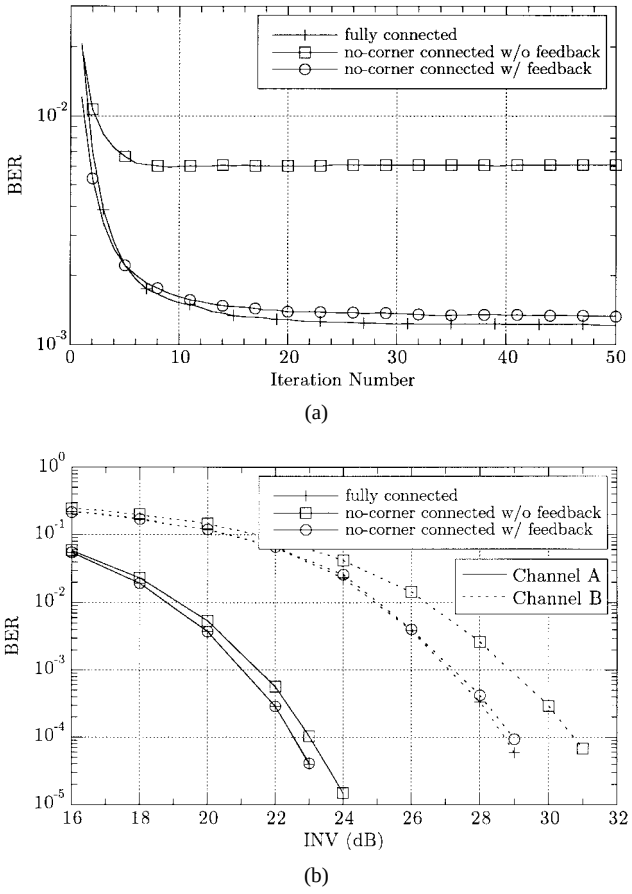
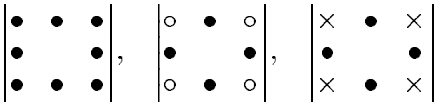


Fig. 2. Impact of the connection complexity on the performance of the $2D^4$ algorithm. (a) Convergence properties for the incoherent channel B at INV = 27 dB. (b) BER performance for the incoherent channels A and B. The number of iterations used is 5 and 20 for channels A and B, respectively.

have considered three different connection masks:



where "•" indicates the connection is included, "○" indicates hard-decision feedback and "×" indicates omission. From left to right, these are referred to as the fully connected (FC), no-corner connected with feedback (NCCF) and no-corner connected without feedback (NCC) masks in order of reducing complexity. In the NCCF case, temporary hard-decisions are made after each iteration based on current likelihood quantities, and are fedback at the corners of the mask where connections are omitted. This approach is similar to well-known state reduction techniques for 1-D data detection [49], [50], as is the DFVA. Note that using feedback only slightly increases the connection complexity. Compared to the FC scheme, the connection complexity is reduced 16 times in both the NCC and NCCF schemes. The incoherent channels A and B are simulated for all three connection masks. Fixing INV = 27 dB in Channel B, Fig. 2(a) shows that the impact of the connection complexity on the convergence rate is negligible. However, a certain performance degradation due to complexity reduction is observed for both channels. Specifically, at a BER of 10^{-4}, the NCC scheme suffers a loss of 0.4 and 2.0 dB in INV on channels A and B, respectively.[3] However, with hard decision feedback, there is virtually no performance degradation for channel A, while the degradation is reduced to only 0.3 dB for channel B. Actually, even the NCC $2D^4$ algorithm outperforms the DFVA algorithm, the performance of which is discussed in Section III-C.

For all subsequent results we have used the fully connected mask.

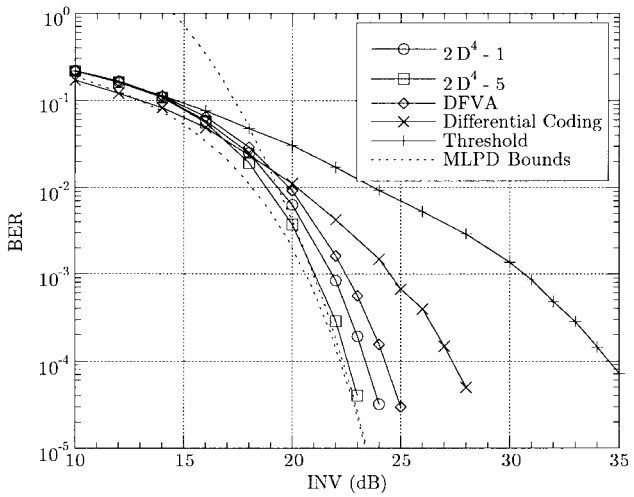


Fig. 3. Comparison of the BER performance for various detection algorithms for the incoherent channel A. Note that "K" in $2D^4 - K$ is the number of iterations used.

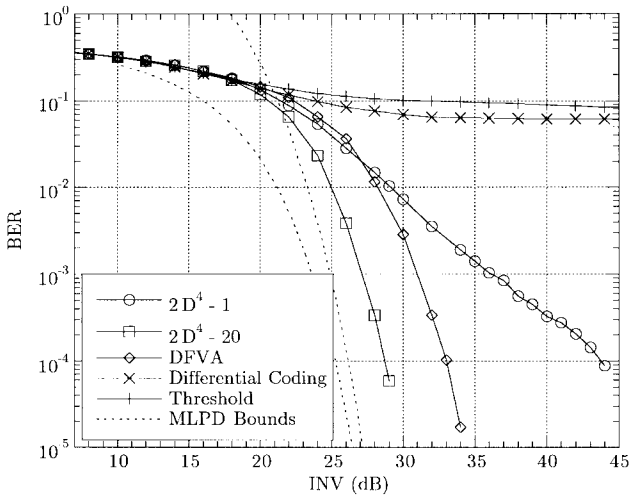


Fig. 4. Comparison of the BER performance for various detection algorithms for the incoherent channel B.

C. Incoherent System Performance

The BER of the infinite contrast ($\alpha_0 = 0$ and $\alpha_1 = 1$) incoherent system with Channels A and B is plotted as a function INV in Figs. 3 and 4, respectively, for various data detection approaches. The theoretical performance bounds associated with MLPD [43] are plotted for reference, which represents the best achievable performance. For channel A, $2D^4$ works virtually optimally with only 5 iterations. Because

[3]Henceforth, unless otherwise specified, the value of INV is always discussed at a BER of 10^{-4}.

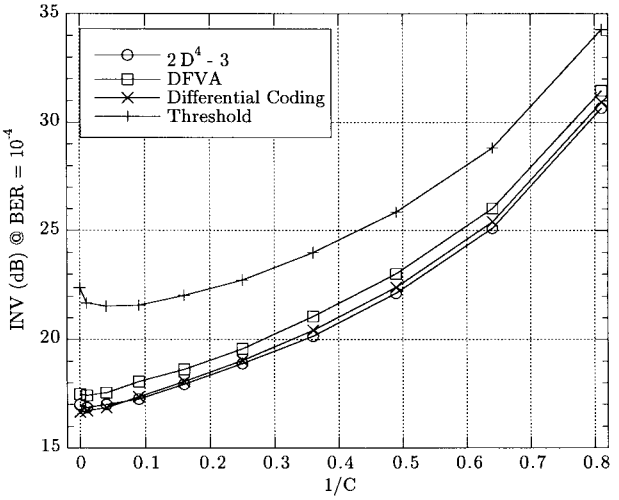


Fig. 5. Comparison of the impact of contrast on various algorithms for the coherent channel A. The x-axis is the inverse of contrast and the y-axis is the value of INV (dB) at which a BER $= 10^{-4}$ is achieved.

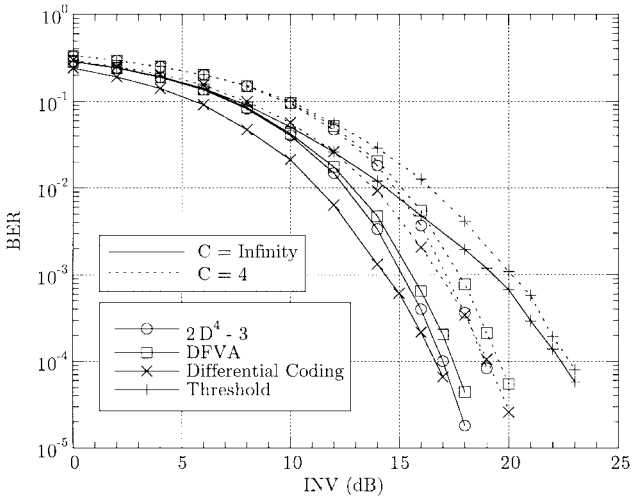


Fig. 6. Comparison of the BER performance for various detection algorithms for the coherent channel A and various contrast levels.

the performance of the 2D^4 algorithm closely approximates the optimal MLPD performance, we refer to the 2D^4 algorithm as "near-optimal." Its performance is 1.7, 4.9, and 12 dB better than that of DFVA, DC, and THA, respectively. The THA and DC are inoperable for the more severe channel B. Both exhibit an error floor around BER $= 0.08$, far above the desired 10^{-4}. The performance of 2D^4 is approximately 2.5 dB worse than the achievable MLPD performance and 4.2 dB better than that of the DFVA. While the 2D^4 algorithm with one iteration provides acceptable performance for channel A (i.e., only 0.9 dB worse than 5-iteration version), it results in a INV cost of 15 dB relative to the 20-iteration version on Channel B.

Due to the linearity of the incoherent channel, the impact of reduced contrast (i.e., $\alpha_0 \neq 0$) in such systems can be directly associated with an increase in noise [44]. Thus, the effects of finite contrast can be determined by translating the curves in Figs. 3 and 4 and are not presented explicitly.

D. Coherent System Performance

The BER performance for the coherent channels A and B is presented in this section. Since the coherent channel is nonlinear, the impact of finite contrast (defined as $C = \alpha_1^2/\alpha_0^2$) does not translate to an equivalent INV degradation. Fig. 5 presents the variations of the value of INV required to achieve a BER of 10^{-4} due to variations in C for channel A. As in the incoherent case, the THA yields the worst BER. Moreover, its behavior with C is complicated due to the nonlinearity of the channel—i.e., the best performance does not occur at $C = \infty$, but rather at $C \cong 25$. For channel A, the performance of the DFVA, DC, and 2D^4 approaches are similar for all levels of contrast. For the remaining coherent channel results, we choose to examine two representative values of C: infinite contrast ($\alpha_1 = 1$ and $\alpha_0 = 0$) and a more typical experimental value of $C = 4$ ($\alpha_1 = 1$ and $\alpha_0 = 0.5$).

The BER results for the coherent channel A are plotted in Fig. 6. With three iterations, the 2D^4 algorithm achieves a gain of $\sim$5 dB over the THA and is slightly better than the DFVA for both infinite contrast and $C = 4$ systems. The DC approach is slightly better than the 2D^4 algorithms in terms of BER on this channel for infinite contrast and comparable for the $C = 4$ case.

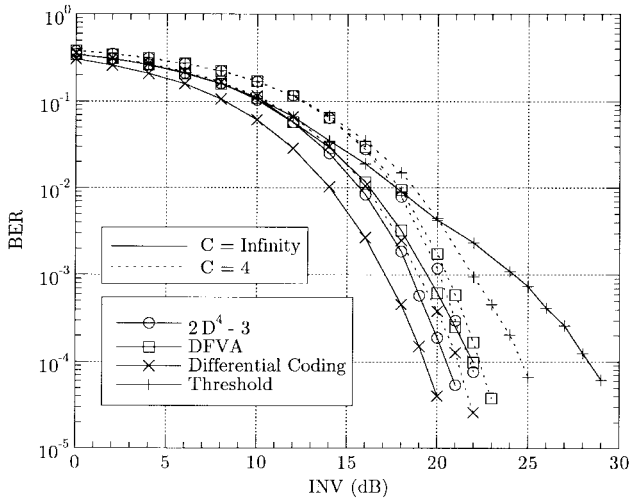
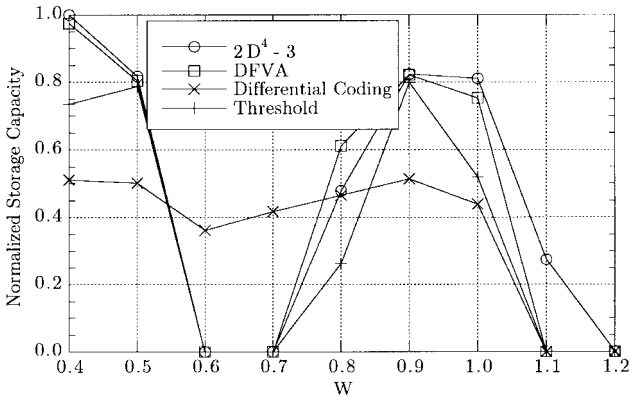


Fig. 7. Comparison of the BER performance for various detection algorithms for the coherent Channel B and various contrast levels.

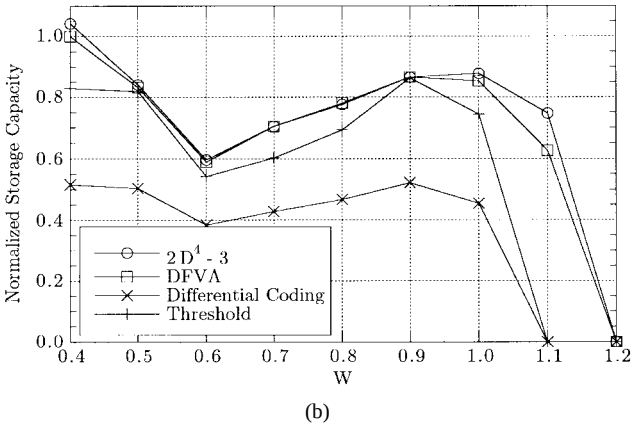
The BER results for the coherent channel B are plotted in Fig. 7. In infinite contrast systems, the 2D^4 algorithm using 3 iterations performs 8 and 1.5 dB better than THA and DFVA, respectively. When $C = 4$, the channel nonlinearity helps the THA to improve the performance by 3.6 dB. The 2D^4 performs 0.5 and 2.8 dB better than the DFVA and THA, respectively. Again, at the cost of 50% coding redundancy, the DC performs slightly better ($\sim$1 dB) than the 2D^4 algorithm.

IV. Holographic Storage Capacity and Density

The results of Section III characterize the BER performance of various approaches, including the 2D^4 algorithm, on particular channels. In this section, we describe how the BER gains realized by more sophisticated processing may be translated to an increase in storage capacity and density. Consider the data fidelity of a coherent POM system as



(a)



(b)

Fig. 8. Comparison of the normalized storage capacity as a function of system blur. (a) C = infinity. (b) $C = 4$.

represented by the BER curves such as those presented in Section III. For the fidelity requirement of BER $= 10^{-4}$, there is a minimum value of the INV which can be tolerated for a given algorithm. Define this value of noise standard deviation as $\sigma_{w,\max}(\mathrm{THA}), \sigma_{w,\max}(2\mathrm{D}^4), \sigma_{w,\max}(\mathrm{DFVA})$, and $\sigma_{w,\max}(\mathrm{DC})$ for the four algorithms considered. Since under the postdetection AWGN model an increase in noise strength is equivalent to a decrease in optical signal intensity, we observe that each $\sigma_{w,\max}(\cdot)$ corresponds to some minimum acceptable signal level $P_{\min}(\cdot) \propto 1/\sigma_{w,\max}(\cdot)$. Due to the diffraction efficiency scaling behavior of holographic POM, the signal power at each detector decreases as $1/M^2$ where M is the number of stored pages [51], [52]. This implies that the maximum number of pages will be limited by $M_{\max}(\cdot) \propto \sqrt{\sigma_{w,\max}(\cdot)}$ for a given algorithm. Thus, a gain in $\sigma_{w,\max}(\cdot)$ results in an increase in $M_{\max}(\cdot)$. The storage capacity is then $M_{\max}(\cdot)N^2$ for all approaches except for DC, for which it is $M_{\max}(\mathrm{DC})N^2/2$ due to the use of two pixels to convey a single bit. The relative capacity achieved using each of these methods is illustrated in Fig. 8 as a function of system blur for $C = \infty$ and $C = 4$. These capacity curves have been normalized to the capacity achieved using the 2D^4 algorithm on a coherent POM system with a blur of $W = 0.4$ and infinite contrast. The normalization constant is therefore determined by the maximum number of pages that can be stored while maintaining a fidelity constraint of BER $= 10^{-4}$ using our proposed algorithm on an infinite contrast, relatively blur-free channel. Although the performance of the 2D^4 algorithm is degraded due to the simple approximation strategy used, it is still the best over most of the range of various blur for two typical values of contrast ratio.

Differential coding is the only approach that meets the fidelity requirement for the infinite contrast case when $W \in [0.6, 0.7]$. However, the storage capacity associated with DC is relatively low for other values of W due to the inherent 50% redundancy. For an infinite contrast system with $W = 1.0$ (the Rayleigh resolution), the 2D^4 provides increases in capacity of approximately 8%, 56%, and 84% relative to that associated with the DFVA, the THA, and DC, respectively. For the $C = 4$ case, these capacity increases are 4%, 18%, and 93%, respectively. Also, the 2D^4 capacity for $C = 4$ is approximately 8% greater than that for the infinite contrast case.

Based only on the curves in Fig. 8, it is tempting to conclude that the performance of a POM is optimized by designing for very small blur and, hence, reducing the need for sophisticated interface processing. However, storage density is another important characteristic of POM and a large blur corresponds to a narrow signal bandwidth which is an indication of area efficiency. In particular, the Fourier plane area associated with a particular value of blur is characterized by $1/W^2$. Since storage density is inversely proportional to this area, we find that storage density may not be optimized for small blur as $1/W^2$ can become large. In order to quantify these trends we define $\rho(\cdot)$ as the storage densities for a given detection algorithm. As described above we have $\rho(\cdot) \propto M_{\max}(\cdot)/A \propto M_{\max}(\cdot)W^2$, where A is the storage area required to record M pages of data while maintaining BER $= 10^{-4}$. Thus, the storage density metric will benefit from interface processing both through an increase in noise margin (i.e., $M_{\max}(\cdot)$) and the ability to tolerate larger W. Again, the DC approach suffers an inherent 50% reduction in storage density. Fig. 9 shows the normalized storage density for all algorithms as a function of blur for two values of contrast ratio. It is clear from the data in Fig. 9 that an optimum blur exists for which density can be maximized. For the $C = \infty$ case, this occurs at approximately $W = 0.9$ for the THA and $W = 1.0$ for all other approaches. The 2D^4 algorithm provides the maximum storage density with relative gains of 9%, 26%, and 85% in storage density relative to the DFVA, THA, and DC approaches, respectively.

Similar results are observed for the $C = 4$ case, with the exception that all approaches are able to meet the fidelity criterion for $0 < W < 1.1$, with the DFVA and 2D^4 approaches also meeting this criterion at $W = 1.1$. In fact the storage density associated with the 2D^4 algorithm is maximized at $W = 1.1$ with $C = 4$ (this occurs at $W = 1.0$ for all other approaches). The relative gains in maximum storage density for the 2D^4 algorithm with $C = 4$ are 6%, 21%, and 98% relative to the DFVA, THA, and DC approaches, respectively.

V. Implementational Issues

The 2D^4 algorithm has been shown to offer superior BER performance for a variety of channels, both coherent and incoherent. The capacity and density gains associated with these

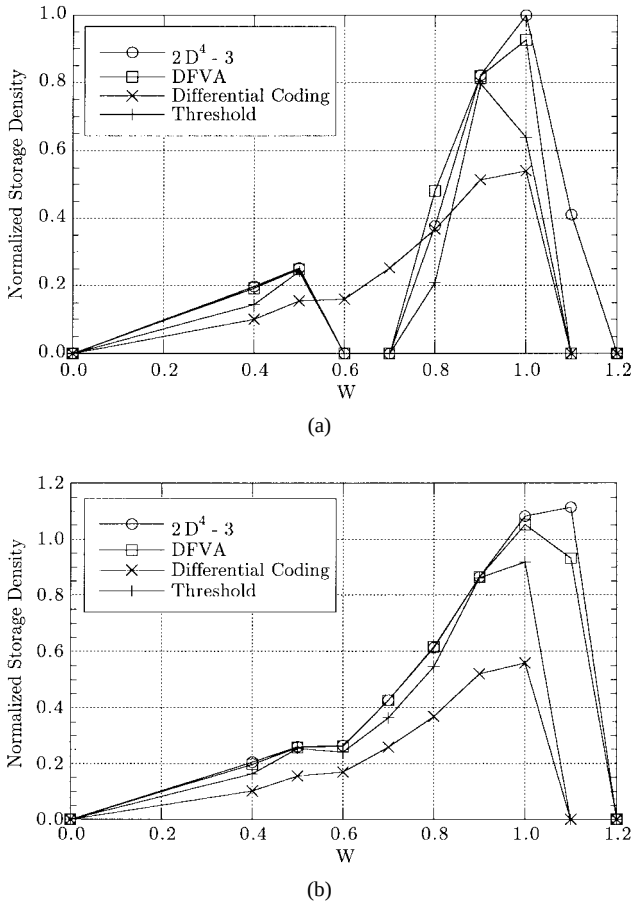


Fig. 9. Comparison of the normalized storage density as a function of system blur. (a) C = infinity. (b) $C = 4$.

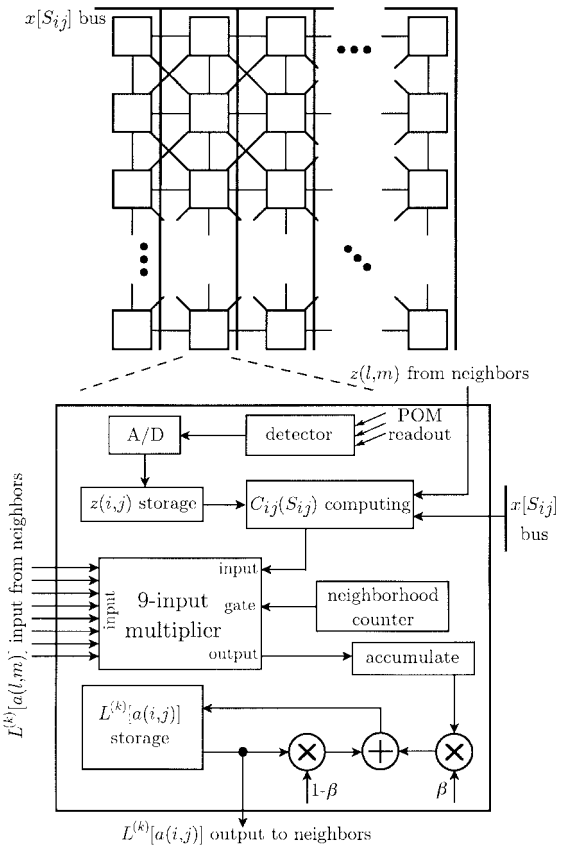


Fig. 10. Schematic of a pixel-parallel neighborhood-serial digital VLSI implementation of the L-2D^4 algorithm. The details of a neighborhood-serial processor are illustrated in the lower portion.

BER improvements were quantified in the previous section. Capacity and density, however, are only two of the relevant storage system metrics, and algorithms that offer increased capacity or density at the expense of latency or transfer rate are of limited use. This implies that there is a bound on the acceptable computational complexity for this application. In this section we investigate the implementational cost of the 2D^4 algorithm. We consider a 2-D parallel implementational model that is a convenient match with the natural 2D data format that characterizes POM. In this section, we measure the implementational cost of 2D^4 in terms of VLSI area and power for a parallel digital solution and suggest a potential analog implementation that exploits the relationship between the 2D^4 algorithm and neural networks.

The likelihood exchange rule (8) represents the basic operation that must be performed at each pixel in a 2D array. For the cases studied here with $L = 1$, the likelihood exchange rule (8) represents a locally connected (8 neighbors) computation in which each pixel first collects information from its neighbors, and then computes an update. We consider these tasks to comprise one iteration-clock so that 1 iteration clock $= 1$ communication clock $+ 1$ computation clock. It is of course also possible to realize such a solution on a mesh-connected array in which two communication clocks are used. According to this pixel-parallel model the 2D^4 algorithm will converge in K iteration clocks, where K is the convergence time represented in Fig. 1 (typically $K \leq 10$). For each computation clock it is necessary for each pixel to accomplish the likelihood exchange (8). This computation requires consideration of all 2^8 possible neighborhood configurations and we propose a sequential approach to this task at each pixel. The 2^8 terms of the sum in (8) are computed sequentially and an accumulation register is used to collect the intermediate results. This requires at least 2^8 basic clock cycles to realize one computation clock. Fig. 10 represents a schematic of the proposed pixel-parallel neighborhood-serial implementation. Each pixel contains an optical detector, 5-bit A/D converter (serial), neighborhood counter (produces 8-bit binary word $\mathbf{n}$), neighborhood-gated 9-input 8-bit multiplier (forms product of only those inputs corresponding to "on" positions of $\mathbf{n}$), accumulation register, two 2-input 8-bit multipliers, one 8-bit adder, and some storage and initialization circuitry. Prior to beginning the iteration procedure some overhead may be required for A/D conversion (32 clocks) and algorithm initialization which we do not include in our cost estimates. The channel information $x(S_{ij})$ (i.e., the noise-free signal conditioned on the value of the support) is communicated via the column busses shown in the figure and together with the stored value of the observation $z(i,j)$, are used to compute the combining coefficients $C_{ij}(S_{ij})$ with a single 8-bit multiplier (e.g., $C_{ij}(S_{ij}) = [z(i,j) - x(S_{ij})]^2$).

Using simple $\lambda = 0.1$ μm CMOS models for the functional units described above, the area cost of the pixel-parallel neighborhood-serial L-2D^4 algorithm has been computed. For this architecture the pixel area is dominated by the 9-input multiplier and is roughly 3.4×10^{-4} cm^2. This value is quite large, and with a reasonable chip size of 2 cm^2, supports page sizes of only 2900 pixels. An alternative to the parallel 9-input multiplier would utilize additional clock cycles ($8\times$) to realize this computation. Such a solution can reduce the area per pixel to 6.0×10^{-5} cm^2 offering array sizes up to 33 340 pixels.

The page sizes that can be supported with the likelihood-based L-2D^4 algorithm are limited to roughly 180 $\times$ 180 pixels. We have also considered an implementation of the metric-based version (G-2D^4) that is described in Appendix I-A. In this case the multipliers are replaced by adders and significant area and power savings are obtained. The G-2D^4 algorithm offers an implementational area per pixel of 2.1×10^{-5} cm^2 which leads to more useful array sizes of nearly 10^5 pixels. We note that additional parallelism can be obtained through the use of resource sharing at the cost of higher clock rates. The 10^5 pixel implementation must operate at a basic clock rate that is roughly 128×8 times the iteration clock rate. Given a desired page access time of 1 ms, we find a required iteration clock rate of 10 kHz and a resulting basic clock rate of 10 MHz. This clock rate is reasonable for the technology that we have assumed. At this clock rate, the power dissipation per pixel for the G-2D^4 algorithm is roughly 0.13 mW, and is $6\times$ less than the power requirements of the likelihood-based method. The total power density associated with G-2D^4 is not unreasonable at 6.9 W/cm^2.

In Section III-B, we investigated the performance of the 2D^4 algorithm when fewer neighbors are utilized in the update computation. Fig. 2, for example, indicates the performance cost of eliminating corner connections from the neighborhood for a severe channel. Consider using such a strategy for the purposes of reducing power dissipation in the associated implementation. For an implementation of G-2D^4 that utilizes no corner connections, the previous 128 $\times$ 8 clock multiplication factor becomes 16 $\times$ 4, reducing the basic clock rate and the power dissipation both by a factor of 16. The resulting power density of 0.43 W/cm^2 is easily manageable and additional multiplexing can now be used to achieve greater area savings through resource sharing. For example, the number of computational blocks can be reduced by a factor of four and shared among groups of four pixels. Such a strategy would facilitate array sizes of nearly 600 $\times$ 600 pixels operating at a 40-MHz basic clock rate, with a reasonable power dissipation requirement of 1.7 W/cm^2. We have ignored the small additional overhead that will be required to manage the resource sharing in this case.

Although the parallel digital solutions described above will benefit from additional cost reductions via more careful circuit design, it is also possible to envision an analog solution to the computational problem. Such a highly parallel, analog implementation may offer significant reductions in area and power. While such a design study is beyond the scope of this work, an interesting point of departure for such an effort might use the similarity between the likelihood propagation rule (8) and (9), and the dynamic behavior of feedback neural networks. These nonlinear dynamical systems have been proven useful in the solution of various optimization problems (e.g., traveling salesman) and the detection problem is yet another in this class of problem [53]–[56]. In particular, the iterative likelihood-based method described here (L-2D^4) can be shown to minimize the criterion function $J = E\{(L[a(i,j)] - L_U[a(i,j)])^2\}$ and the update rule (7) can be cast in the form of a fully parallel network as shown in Fig. 11. Fig. 11 depicts a locally connected network of subnets. Each subnet can be thought of as a two-layer nonlinear (8th order) network [57]–[59]. Alternately, each subnet may be considered to be a single piecewise linear perceptron operating on 2^8 so-called Phi-functions, each of which is one of the 2^8 possible products of 8 neighboring likelihoods [60]. It is also possible to consider the neural analog of the G-2D^4 algorithm. Although it becomes difficult to write a simple criterion function in this case, the recent development of morphological networks does provide a framework for the study of a parallel distributed min-sum approach [61]–[62].

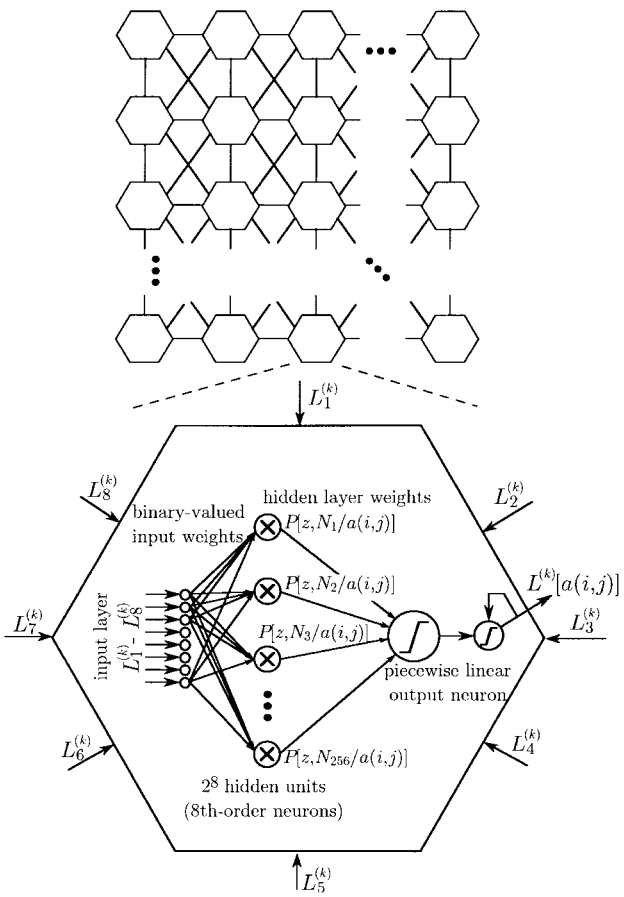


Fig. 11. Schematic of a fully parallel neural network implementation of the L-2D^4 algorithm. The details of a subnet are illustrated in the lower portion.

VI. Conclusion and Discussion

In this paper, we have described a new data detection algorithm for use within the highly parallel POM interface environment. This algorithm is fashioned after other iterative, soft-decision, maximum-likelihood techniques and represents

a highly parallel, distributed computational model that is well suited to the 2-D data format characteristic of POM systems. The new 2D^4 algorithm is motivated from a likelihood-based perspective and its relationships with MLPD and MLBD have been discussed. Both likelihood-based (L-2D^4) and generalized likelihood-based (G-2D^4) versions have been described. The add-compare-select based computation of the G-2D^4 algorithm is shown to offer significant reductions in implementational cost as compared with the multiplication-intensive L-2D^4 algorithm. The BER performance of the G-2D^4 algorithm has been compared with three other approaches (THA, DC, and DFVA) for finite contrast POM channels based on both incoherent (linear) and coherent (nonlinear) optical systems. Varying degrees of coherent and incoherent optical blur have been studied. The BER advantages of the G-2D^4 algorithm as compared with the other three methods, are translated directly into gains in storage capacity and density. In particular, for the case of a Rayleigh resolved holographic POM system with infinite contrast we find that the G-2D^4 algorithm offers capacity improvements of 84%, 56%, and 8% as compared with DC, THA, and DFVA respectively, with corresponding storage density gains of 85%, 26%, and 9%. In the case of finite contrast ($C = 4$), comparable capacity improvements of 93%, 18%, and 4% produce similar density improvements of 98%, 21%, and 6%.

The holographic POM model for which the above results are obtained, describes an observation $z(i,j)$ that is nonlinear in the input data $a(i,j)$. In this case therefore, it is natural to expect that likelihood-based methods will be superior owing to the ease with which such methods can incorporate accurate channel information (i.e., the neighborhood-conditional expected received signal $x(S_{ij})$). The results described above bear out this expectation with the G-2D^4 algorithm offering significant capacity/density advantages as compared with THA and DC and only modest improvements as compared with the other likelihood-based technique (DFVA).

Various algorithmic and implementational issues related to the use of the new 2D^4 algorithm have also been discussed. A study of algorithm convergence has shown the relationship between convergence time and final BER, with the best BER performance obtained for the slowest convergence parameter ($\beta \ll 1$). A reasonable compromise between BER and convergence time is found to be $\beta = 0.3$ for the case of a fully connected neighborhood. Convergence in these cases was generally obtained with fewer than 10 iterations. The neighborhood connection complexity was also studied and it was found that using a reduced connection mask allows significant reductions in complexity with little performance degradation. The use of hard-decision feedback may be required with reduced connection algorithms in order to maintain the performance for more severe ISI channels (e.g., the incoherent channel B).

Implementational issues associated with the realization of the G-2D^4 algorithm have also been discussed and a parallel digital focal plane architecture is considered in detail. A 2-cm^2 $\lambda = 0.1$ μm digital VLSI real estate budget is shown to support a 600 $\times$ 600 pixel G-2D^4 focal plane solution operating at 40 MHz with less than 1.7 W/cm^2 power dissipation. We have also suggested the use of a highly parallel analog implementation and have outlined the relationship between such a solution and other neural network based optimization circuits.

While this paper has introduced a useful algorithm offering both good BER performance and reasonable implementational cost, several outstanding issues remain. The most critical of these concern the recovery of algorithm performance in cases for which there is a mismatch between the physical channel support and the neighborhood connectivity used during 2D^4 processing. The results presented here are based on a simple averaging procedure that allows the use of $L = 1$ processing on $L = 2$ channels. The reduced connection and hard-decision feedback results presented in Section III-B suggest that such techniques may provide improved capacity and density gains with marginal increases in processing complexity. Additional approaches to complexity reduction for particularly severe channels (e.g., coherent POM systems with $0.5 \leq W \leq 0.8$ and $W \geq 1.0$) could perhaps be based on other existing approaches for 1D sequence detection (e.g., pre-equalization, and variable complexity tree-search algorithms). Effective methods for complexity reduction are also necessary to include the memory associated with signal encoding (e.g., modulation coding and error control coding).

A second set of issues concerns the implementation of these likelihood-based methods in experimental POM systems. Various imperfections associated with experimental POM will cause the actual system to depart from the system model used during 2D^4 processing. Understanding the BER degradation in this case represents an important tolerancing exercise and may result in the need to use empirical channel data within the 2D^4 algorithm. We are pursuing experimental verification of such a strategy. Additional design work must also be completed before a final implementational strategy can be established. The results of the baseline design study presented here suggest that the highly parallel and distributed nature of the 2D^4 algorithm make it suitable for a parallel 2-D focal plane implementation; however, detailed digital and analog circuit designs must be established and compared with competing algorithmic approaches.

APPENDIX I
COMPUTATIONAL DETAILS OF THE 2D^4 ALGORITHM

A. Generalized-Likelihood Based 2D^4 Algorithm

The generalized likelihood (GL) of $a(i,j)$ based on $z(i,j)$ may be computed by

$$P_G[z(i,j)|a(i,j)] = \max_{N_{ij} \in \Omega} \left(P[z(i,j)|S_{ij}] \prod_{a(l,m) \in N_{ij}} P[a(l,m)] \right) \tag{10}$$

which is directly analogous to (7). The advantage of using GL is that, unlike the averaging operation in (7), the maximization operation commutes with any strictly monotonic function.

Thus, we may rewrite (10) as

$$-\log(P_G[z(i,j)|a(i,j)]) = \min_{N_{ij}\in\Omega}\left(-\log(P[z(i,j)|S_{ij}]) + \sum_{a(l,m)\in N_{ij}} -\log(P[a(l,m)])\right). \quad (11)$$

In this form, the GL computation includes only Addition-Comparison-Select (ACS) operations, which are computationally simple. Thus, we define a *metric* as $M(\cdot) = -\epsilon_1 \log P(\cdot) + \epsilon_2$ where $\epsilon_1 > 0$ and ϵ_2 are constants. For example, in the AWGN channel, ϵ_1 and ϵ_2 may be selected so that the metric is the squared Euclidean distance (e.g., $(z(i,j) - x(S_{ij}))^2$). Based on these observations, we suggest a metric-based GL propagation rule analogous to (8)–(9)

$$M_U^{(k)}[a(i,j)] = \min_{N_{ij}\in\Omega}\left(C_{ij}^M[S_{ij}] + \sum_{a(l,m)\in N_{ij}} M^{(k-1)}[a(l,m)]\right) \quad (12)$$

$$M^{(k)}[a(i,j)] = (1-\beta)M^{(k-1)}[a(i,j)] + \beta M_U^{(k)}[a(i,j)] \quad (13)$$

where $C_{ij}^M[S_{ij}]$ is the metric version of $C_{ij}[S_{ij}]$ in log-domain. After K iterations are accomplished, the decisions are made according to: $\hat{a}(i,j) = \alpha_1$ if $M^{(K)}[a(i,j) = \alpha_0] > M^{(K)}[a(i,j) = \alpha_1]$; otherwise $\hat{a}(i,j) = \alpha_0$.

2) Setting the Likelihood Combining Coefficients

Since the objective of the iterative algorithms is to converge to the likelihood of $a(i,j)$ based on the entire page Z, it is desirable to set the likelihood combining coefficients to the likelihood of S_{ij} based on the largest possible region of the observation data to achieve stronger combining effects. The simplest case is to use $P[z(i,j)|S_{ij}]$ in the L-2D^4 algorithm. With only slightly higher complexity, the combining coefficients may be set to the likelihood of S_{ij} based on $\boldsymbol{z}_+(i,j) = \{z(i,j), z(i\pm 1,j), z(i,j\pm 1)\}$—i.e., the set of observations located on a cross $R = \{(0,0),(\pm 1,0),(0,\pm 1)\}$. Thus, we calculate $C_{ij}[S_{ij}] = P[\boldsymbol{z}_+(i,j)|S_{ij}]$. With the definitions of $V(i,j) = \{a(i,j), a(i\pm 1,j)\}, V(i,j) = \{a(i,j), a(i,j\pm 1)\}$ and $HV(i,j;m) = \{H(i\pm m,j), V(i,j\pm m)\}$, the initialization scheme is given by

$$\begin{aligned} C_{ij}[S_{ij}] &= P[\boldsymbol{z}_+(i,j)|S_{ij}] \\ &= \sum_{HV(i,j;2)} P[\boldsymbol{z}_+(i,j), HV(i,j;2)|S_{ij}] \\ &= c \sum_{HV(i,j;2)} P[\boldsymbol{z}_+(i,j)|HV(i,j;2), S_{ij}] \\ &= c \sum_{HV(i,j;2)} \prod_{(l,m)\in R} P[z(i+l,j+m)|S_{i+l,j+m}]. \end{aligned} \quad (14)$$

The constant c is independent of S_{ij}. The last equality follows from the conditional independence of the observations in $\boldsymbol{z}_+(i,j)$ and the fact that $z(i+l,j+m)$ is only dependent on $S_{i+l,j+m} \subset \{HV(i,j;2), S_{ij}\}$. After completing the calculation of $P[z(i,j)|S_{ij}]$'s, we use (14) to obtain $P[\boldsymbol{z}_+(i,j)|S_{ij}]$. Note that by carefully accounting for common terms in (14), one can simplify the evaluation further. For G-2D^4, the corresponding rule is

$$\begin{aligned} C_{ij}^M[S_{ij}] &= M[\boldsymbol{z}_+(i,j)|S_{ij}] \\ &= \min_{HV(i,j;2)} \sum_{(l,m)\in R} M[z(i+l,j+m)|S_{i+l,j+m}]. \end{aligned} \quad (15)$$

While this operation does not require significantly more computation than simply using $C_{ij}[S_{ij}] = P[z(i,j)|S_{ij}]$, attempting to compute the likelihood of S_{ij} based on larger regions of the observation results in an exponential increase in complexity.

APPENDIX II
LIKELIHOOD RATIO PROPAGATION

Since binary data is assumed, the decision statistic required is the ratio of the likelihoods of the "1" and "0" hypotheses. This fact may be exploited in order to reduce the computational and storage requirements of the 2D^4 algorithm by a factor of two. Specifically, instead of updating both $L^{(k)}[a(i,j) = \alpha_1]$ and $L^{(k)}[a(i,j) = \alpha_0]$ as in (8), we only need to update the likelihood ratio $\Lambda^{(k)}[a(i,j)] = L^{(k)}[a(i,j) = \alpha_1]/L^{(k)}[a(i,j) = \alpha_0]$ by

$$\Lambda^{(k)}[a(i,j)] = (1-\beta)\Lambda^{(k-1)}[a(i,j)] + \beta \frac{L_U^{(k)}[a(i,j) = \alpha_1]}{L_U^{(k)}[a(i,j) = \alpha_0]} \quad (16)$$

and the decision rule is modified to be $\hat{a}(i,j) = \alpha_1$ if $\Lambda^{(K)}[a(i,j)] > 1$; otherwise $\hat{a}(i,j) = \alpha_0$. By only storing $\Lambda^{(k)}[a(i,j)]$, the memory requirements for the instantaneous soft information quantities are cut in half. Also, the likelihood exchange rule (8) is changed to

$$L_U^{(k)}[a(i,j)] = \sum_{N_{ij}\in\Omega} C_{ij}[S_{ij}] \prod_{\substack{a(l,m)\in N_{ij} \\ a(l,m)=\alpha_1}} \Lambda^{(k-1)}[a(l,m)]. \quad (17)$$

Since the likelihood information from $a(i,j)$ is included only when it is α_1 in a given N_{ij}, on average this saves half of the multiplication operations. Consequently, in a fully parallel implementation of the L-2D^4 algorithm, it also saves half of the connections. Similarly, we can define the generalized log-likelihood ratio $\Gamma^{(k)}[a(i,j)] = M^{(k)}[a(i,j) = \alpha_1] - M^{(k)}[a(i,j) = \alpha_0]$ in the G-2D^4 algorithm. The corresponding metric propagation scheme is

$$M_U^{(k)}[a(i,j)] = \min_{N_{ij}\in\Omega}\left(C_{ij}^M[S_{ij}] + \sum_{\substack{a(l,m)\in N_{ij} \\ a(l,m)=\alpha_1}} \Gamma^{(k-1)}[a(i,j)]\right) \quad (18)$$

$$\Gamma^{(k)}[a(i,j)] = (1-\beta)\Gamma^{(k-1)}[a(i,j)] + \beta(M_U^{(k)}[a(i,j)=\alpha_1] - M_U^{(k)}[a(i,j)=\alpha_0]). \quad (19)$$

The corresponding decision rule is $\hat{a}(i,j) = \alpha_1$ if $\Gamma^{(K)}[a(i,j)] < 0$; otherwise $\hat{a}(i,j) = \alpha_0$. The same relative computational and storage savings are achieved in G-2D^4 version as in L-2D^4 algorithm.

REFERENCES

[1] M. Moriya and T. Tokura, "Outline of DVD standard," *Nat. Tech. Rep.*, vol. 43, pp. 3–9, 1997.

[2] D. R. Guenette and D. J. Parker, "CD, CD-ROM, CD-R, CD-RW, DVD, DVD-R, DVD-RAM: The family album," *EMedia Professional*, vol. 10, pp. 30–34, 1997.

[3] P. J. van Heerden, "Theory of optical information storage in solids," *Appl. Opt.*, vol. 2, pp. 393–400, 1963.

[4] G. Sincerbox, "Holographic storage revisited," in *Current Trends in Optics*, C. Dainty, Ed. New York: Academic, 1994.

[5] M.-P. Bernal, G. W. Burr, H. Coufal, R. K. Grygier, J. A. Hoffnagle, C. M. Jefferson, R. M. McFarlane, R. M. Shelby, G. T. Sincerbox, and G. Wittman, "Holographic-data-storage materials," *MRS Bull.*, vol. 21, pp. 51–60, 1996.

[6] L. Hesselink and M. C. Bashaw, "Optical memories implemented with photorefractive media," *Opt. Quantum Electron.*, vol. 25, pp. 611–661, 1993.

[7] S. Campbell, X. Yi, and P. Yeh, "Hybrid sparse-wavelength angularly multiplexed optical data storage system," *Opt. Lett.*, vol. 19, pp. 2161–2163, 1994.

[8] I. McMichael, W. Christian, J. Hong, T. Y. Chang, R. Neurogaonkar, and M. Khoshnevisan, "Compact volume holographic memory system with rapid acoustooptic addressing," in *Proc. LEOS/OSA IEEE Nonlinear Optics: Materials, Fundamentals, and Applications, 1994*, 1995, pp. 424–426.

[9] H.-Y. S. Li and D. Psaltis, "Three dimensional holographic disks," *Appl. Opt.*, vol. 33, pp. 3764–3774, 1994.

[10] E. S. Maniloff, S. B. Altner, S. Bernet, F. R. Graf, A. Renn, and U. P. Wild, "Recording of 6000 holograms by use of spectral hole burning," *Appl. Opt.*, vol. 34, pp. 4140–4148, 1995.

[11] F. Mok, "Angle-multiplexed storage of 5000 holograms in lithium niobate," *Opt. Lett.*, vol. 18, pp. 915–917, 1993.

[12] G. W. Burr, F. H. Mok and D. Psaltis, "Angle and space multiplexed holographic storage using the 90 degree geometry," *Opt. Commun.*, vol. 117, pp. 49–55, 1995.

[13] G. W. Burr, F. H. Mok, and D. Psaltis, "Large scale volume holographic storage in the long interaction length architecture," *Proc. SPIE*, vol. 2297, pp. 402–414, 1994.

[14] A. Pu and D. Psaltis, "High-density recording in photopolymer-based holographic three-dimensional disks," *Appl. Opt.*, vol. 35, pp. 2389–2398, 1996.

[15] S. Hunter *et al.*, "Three-dimensional optical image storage by two-photon recording," *Optical Memory Neural Net.*, vol. 3, pp. 151–166, 1994.

[16] A. S. Dvornikov, I. Cokgor, F. McCormick, R. Piyaket, S. Esner, and P. M. Rentzepis, "Molecular transformations as a means for 3D optical memory devices," *Opt. Commun.*, vol. 128, pp. 205–210, 1996.

[17] J. Kikas and K. Leiger, "Effect of geometry on storage density in spectral hole burning memories," *Opt. Commun.*, vol. 94, pp. 557–560, 1992.

[18] M. P. Bernal, G. W. Burr, H. Coufal, R. K. Grygier, J. A. Hoffnagle, C. M. Jefferson, E. Oesterschulze, R. M. Shelby, G. T. Sincerbox, and M. Quintanilla, "Effects of multilevel phase masks on interpixel crosstalk in digital holographic storage," *Appl. Opt.*, vol. 36, pp. 3107–3115, 1997.

[19] G. W. Burr, W. C. Chou, M. A. Neifeld, H. Coufal, J. A. Hoffnagle, and C. M. Jefferson, "Experimental evaluation of optimal coded user capacity in holographic data storage systems," *Appl. Opt.*, submitted for publication.

[20] W. C. Chou and M. A. Neifeld, "Interleaving and error correction in volume holographic memory systems," *Appl. Opt.*, submitted for publication.

[21] M. C. Bashaw, J. F. Heanue, and L. Hesselink, "Organization of data for monochromatic multiplexed volume holography," *J. Opt. Soc. Amer. A*, vol. 13, pp. 2174–2186, 1996.

[22] C. Gu, J. Hong, I. McMichael, R. Saxena, and F. Mok, "Cross-talk limited storage capacity of volume holographic memory," *J. Opt. Soc. Amer. A*, vol. 10, pp. 2547–2550, 1993.

[23] X. Yi, P. Yeh, and C. Gu, "Statistical analysis of cross-talk noise and storage capacity in volume holographic memory," *Opt. Lett.*, vol. 19, pp. 1580–1582, 1994.

[24] E. S. Maniloff and K. M. Johnson, "Effects of scattering on the dynamics of holographic recording and erasure in photorefractive lithium niobate," *J. Appl. Phys.*, vol. 73, pp. 541–547, 1993.

[25] K. M. Chugg, X. Chen, and M. A. Neifeld, "Two-dimensional linear MMSE equalization for page-oriented optical memories," presented at the 31st Annu. Asliomar Conf. Signals, Systems, and Computers, 1997, paper MP6-7.

[26] J. Heanue, M. Bashaw, and L. Hesselink, "Channel codes for digital holographic data storage," *J. Opt. Soc. Amer. A*, vol. 12, pp. 2432–2439, 1995.

[27] B. Olson and S. Esener, "Partial response precoding for parallel readout optical memories," *Opt. Lett.*, vol. 19, pp. 661–663, 1994.

[28] B. Olson and S. Esener, "Multidimensional partial response for parallel readout optical memories," *Proc. SPIE*, vol. 2297, pp. 331–344, 1994.

[29] J. F. Heanue, K. Gurkan, and L. Hesselink, "Signal detection for page access optical memories with intersymbol interference," *Appl. Opt.*, vol. 35, pp. 2431–2438, 1996.

[30] J. F. Hutton, G. A. Betzos, M. Schaffer, and P. A. Mitkas, "Error correcting codes for page-oriented optical memories," *Proc. SPIE*, vol. 2848, pp. 146–156, 1996.

[31] B. J. Goertzen and P. A. Mitkas, "Error-correcting code for volume holographic storage of a relational database," *Opt. Lett.*, vol. 20, pp. 1655-1657, 1995.

[32] 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," *Opt. Lett.*, vol. 22, pp. 639–641, 1997.

[33] A. Vardy, M. Blaum, P. H. Siegel, and G. Sincerbox, "Conservative arrays: Multidimensional modulation codes for holographic recording," *IEEE Trans. Inform. Theory*, vol. 42, pp. 227–229, 1996.

[34] M. A. Neifeld and J. D. Hayes, "Parallel error correction for optical memories," *J. Optical Memory and Neural Networks*, vol. 3, pp. 87–98, 1994.

[35] M. A. Neifeld and M. McDonald, "Error correction for increasing the usable capacity of photorefractive memories," *Opt. Lett.*, vol. 19, pp. 1483–1485, 1994.

[36] M. A. Neifeld and J. D. Hayes, "Error correction schemes for volume optical memories," *Appl. Opt.*, vol. 34, pp. 8183–8191, 1995.

[37] X. Chen and K. M. Chugg, "Near-optimal data detection for two-dimensional ISI/AWGN channels using concatenated modeling and iterative algorithms," in *Proc. IEEE 1998 Int. Conf. Communications*, 1998, paper S27P04.

[38] K. M. Chugg and X. Chen, "Efficient architectures for soft output algorithms," *Proc. IEEE 1998 Int. Conf. Communications*, 1998, paper S04P04.

[39] C. Berrou and A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes (1)," in *Proc. IEEE 1989 Int. Conf. Communications*, 1993, pp. 1064–1070.

[40] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *IEEE Trans. Inform. Theory*, vol. 44, pp. 909–926, 1998.

[41] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as An instance of Pearl's "Belief propagation" algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 140–152, 1998

[42] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, 1973.

[43] K. M. Chugg, "Performance of optimal digital page detection in a two-dimensional ISI/AWGN channel," presented at the 30th Asilomar Conf. on Signal, Systems and Computers, 1996, paper TP4-8.

[44] K. M. Chugg, X. Chen, and M. A. Neifeld, "Two-dimensional equalization in coherent and incoherent page oriented optical memory," to appear in *J. Opt. Soc. Amer. A*, 1998.

[45] C. L. Miller, B. R. Hunt, M. W. Marcellin, and M. A. Neifeld, "Bilevel image reconstructions via 2D Viterbi algorithm," in *Proc. IEEE Int. Conf. Image Processing*, 1997.

[46] J. F. Hayes, T. M. Cover, and J. B. Riera, "Optimal sequence detection and optimal symbol-by-symbol detection: Similar algorithms," *IEEE Trans. Commun.*, vol. COM-30, pp. 152–157, 1982.

[47] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*. New York: Wiley, 1968.

[48] J. F. Heanue, M. C. Bashaw, and L. Hesselink, "Channel codes for digital holographic data storage," *J. Opt. Soc. Amer. A*, vol. 12, pp. 2432–2439, 1995.

[49] A. Duel-Hallen and C. Heegard, "Delayed decision feedback estimation," *IEEE Trans. Commun.*, vol. 37, pp. 428–436, 1989.
[50] M. V. Eyuboğlu and S. U. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Trans. Commun.*, vol. 36, pp. 13–20, 1988.
[51] A. Strasser, E. Maniloff, K. Johnson, and S. Goggin, "Procedure for recording multiple-exposure holograms with equal diffraction efficiency in photorefractive media," *Opt. Lett.*, vol. 14, pp. 6–8, 1989.
[52] D. Brady and D. Psaltis, "Control of volume holograms," *J. Opt. Soc. Amer. A*, vol. 9, pp. 1167–1182, 1992.
[53] J. J. Hopfield and D. W. Tank, "Computing with neural circuits," *Science*, vol. 23, pp. 625–633, 1986.
[54] M. Budinich, "A self-organizing neural network for the traveling salesman problem that is competitive with simulated annealing," *Neural Computation*, vol. 8, pp. 416–424, 1996.
[55] M. Ohlsson, C. Peterson, and B. Soderberg, "Neural networks for optimization problems with inequality constraints: The knapsack problem," *Neural Computation*, vol. 5, pp. 331–339, 1993.
[56] A. H. Gee and R. W. Prager, "Limitations of neural networks for solving traveling salesman problems," *IEEE Trans. Neural Networks*, vol. 6, pp. 280–282, 1995.
[57] D. Burshtein, "Long-term attraction in higher order neural networks," *IEEE Trans. Neural Networks*, vol. 9, pp. 42–50, 1998.
[58] T. Kanaoka, R. Chellappa, M. Yoshitaka, and S. Tomita, "A higher-order neural network for distortion invariant pattern recognition," *Pattern Recognition Lett.*, vol. 13, pp. 837–841, 1992.
[59] J. Li, Y. Yu, and W. Li, "The studies on the higher order neural network using for pattern recognition and optimization problem," *Proc. SPIE*, vol. 2492, pp. 748–755, 1995.
[60] M. L. Minsky and S. A. Papert, *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA: MIT Press, 1969.
[61] G. X. Ritter, P. Sussner, and J. L. Diaz-de-Leon, "Morphological associative memories," *IEEE Trans. Neural Networks*, vol. 9, pp. 281–293, 1998.
[62] J. L. Davidson and F. Hummer, "Morphology neural networks: An introduction with applications," *J. Circuit Syst. Signal Processing*, vol. 12, pp. 177–210, 1993.

Xiaopeng Chen (S'98) was born in Sichuan Province, China, in 1971. He received the B.E. (with honors) and M.E. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1993 and 1995, the M.S. degree in electrical engineering from the Pennsylvania State University, University Park, PA, in 1996, and is currently working toward the Ph.D. degree in electrical engineering at the University of Southern California, Los Angeles, CA.

His research interests are in the general area of data detection and estimation for digital communication and image processing systems, and all-optical signal processing in high-capacity optical fiber transmission systems and networks.

Keith M. Chugg (S'88–M'95) was born in Los Angeles, CA on March 16, 1967. He received the B.S. degree (high distinction) in engineering from Harvey Mudd College, Claremont, CA, in 1989 and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California (USC), Los Angeles, CA, in 1990 and 1995, respectively.

He served as an instructor in the Electrical Engineering-Systems Department, USC, from 1993 to 1995. During his graduate study, he held summer positions at TRW, Redondo Beach, CA (1991, 1992), Lincom Corp., Los Angeles, CA (1990), and Qualcomm Inc., San Diego, CA (1989). During the 1995–1996 academic year, he was an Assistant Professor in the Department of Electrical and Computer Engineering, University of Arizona. Since 1996, he has been with the EE-Systems Department, USC, where he is an Assistant Professor. His research interests are in the general area of signaling, detection, and estimation for digital communication and data storage systems.

Mark A. Neifeld (S'84–M'91) received the B.S. degree in electrical engineering from Georgia Institute of Technology, Athens, GA, in 1984, and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, CA, in 1987 and 1991, respectively.

In July of 1991, he joined the faculty of the University of Arizona, Tucson, where he is now an Associate Professor in the Department of Electrical and Computer Engineering and the Optical Sciences Center. His research interests include multidimensional coding and signal processing, parallel access to optical memory, space-radiation effects in optoelectronic devices, optical computing and interconnects, neural networks, and pattern recognition.