On the Capacity of 2-D Constrained Codes and Consequences for Full-Surface Data Channels

William Weeks IV
Department of Electrical and Computer Engineering
University of Missouri–Rolla
1870 Miner Circle
Rolla, MO 65409
USA

Abstract

The use of constrained binary two-dimensional codes can provide significant density improvement for data-storage devices that are modeled as two-dimensional intersymbol-interference channels with additive noise. The two-dimensional capacity of such codes is bounded by considering strips in the plane of width n and constructing matrix recursions that simplify bound computations with increasing n. Curve-fitting and bounding techniques allow precise estimates of the capacity to be made. Finally, by assuming that capacity-achieving encoders and decoders exist, density improvements for full-surface channels by use of checkerboard codes are calculated, and those calculations verified by numerical simulation.

1 Introduction

Two-dimensional constrained codes (i.e. checkerboard codes) have important applications in the reliable transmission of data across optical channels where data are referenced by a two-dimensional index set, (i.e. full-surface channels). This paper connects the calculation of the capacity of checkerboard codes with increased information density in full-surface data-storage systems.

Many authors have examined two-dimensional constrained codes. Recent work in two-dimensional runlength-limited codes, which arise in magnetic recording applications, includes multitrack recording [17], [16], work in cascading arrays [4], and calculation of capacity of two- and three-dimensional RLL codes [11], [15], [10]. The two-dimensional codes examined in this paper are related to two-dimensional RLL codes. Work related to computing capacity for general constraints for two-dimensional codes also exists [3], [2], [19], [6], [5]. Finally, work in constructing efficient encoders and decoders for constrained two-dimensional codes has also been completed [8].

Computing the capacity of a two-dimensional constrained code appears to be intrinsically harder than computing the capacity of a one-dimensional constrained code. In Section 2 the capacity of a constrained 2-D code (i.e. checkerboard code) is computed by constructing a matrix recursion, calculating a sequence of eigenvalues, and computing bounds that result in precise estimates of the capacity. These estimates of capacity are then used to

```
  0
0 1 0
  0
```

Diamond

```
0 0
0 1 0
  0 0
```

Hexagonal

```
0 0 0
0 1 0
0 0 0
```

Square

```
    0
  0 0 0
0 0 1 0 0
  0 0 0
    0
```

Diamond II

```
0 0 0
0 0 0 0
0 0 1 0 0
  0 0 0 0
    0 0 0
```

Hexagonal II

```
0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 0 0
0 0 0 0 0
```

Square II

```
      0
    0 0 0
  0 0 0 0 0
0 0 0 1 0 0 0
  0 0 0 0 0
    0 0 0
      0
```

Diamond III

```
0 0 0 0
0 0 0 0 0
0 0 0 0 0 0
0 0 0 1 0 0 0
  0 0 0 0 0 0
    0 0 0 0 0
      0 0 0 0
```

Hexagonal III

```
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

Square III

Figure 1: Various checkerboard constraints.

compute potential density improvements in full-surface data-storage systems that employ simple threshold demodulation. In Section 3 notation and system model is introduced for full-surface channels, and in Section 4 optimized thresholds are computed, efficient encoders and decoders for constrained codes and differential encoders and decoders are presumed to exist, and predicted results are verified for simple cases.

2 Capacity

A *checkerboard constraint* is a two-dimensional arrangement of *zeros* that must surround every *one* in a two-dimensional binary code. Various checkerboard constraints are shown in Figure 1. An $n \times m$ *checkerboard code* is the set of all binary arrays of size $n \times m$ that satisfy a given checkerboard constraint.

The *capacity*, C, for a given checkerboard code is

$$C = \lim_{n\to\infty} \lim_{m\to\infty} \frac{\log_2[M(n,m)]}{nm}$$

where $M(n,m)$ is the number of distinct codewords in the code. It is known [9] that the capacity exists for the checkerboard constraints in this paper.

The capacity C_n of a an $n \times \infty$ block is

$$C_n = \lim_{m \to \infty} \frac{\log_2[M(n,m)]}{nm}.$$

The sequence C_n will be used to estimate C.

For a general checkerboard constraint and fixed n, we may write

$$\mathbf{x}(m) = \mathbf{A}_n \mathbf{x}(m-1)$$

where $x_i(m)$ is the number of distinct $n \times m$ codewords that end with some pattern of d column vectors, and $\mathbf{A}_n$ is a matrix for elements of $\mathbf{x}$. The parameter d specifies the order of the constraint and is equal to one for constraints labeled with "I", two for constraints labeled with "II" and so forth. For $m \geq 1$, we have

$$\mathbf{x}(m) = \mathbf{A}_n^{m-1} \mathbf{x}(1)$$

The Perron-Frobenius theorem [13] can be applied to the adjacency matrices and it can be shown that as m approaches ∞,

$$C_n = \frac{\log_2[\lambda_n]}{n}$$

where λ_n is the largest eigenvalue of $\mathbf{A}_n$.

Finally, for arbitrary n and a dth-order constraint

$$C_n \frac{n}{n+d} \leq C \leq C_n.$$

The capacity, C, is given by

$$C = \lim_{n \to \infty} C_n$$

Computational limits prevent us from determining C to arbitrary accuracy by letting n increase. Rather, the sequence C_n results in a series of upper and lower bounds on C.

We have developed convenient recursions to construct $\mathbf{A}_n$, facilitating the efficient computation of λ_n. By carefully numbering the set of all legal $n \times 1$ column vectors, we can construct the recursion for the diamond constraint ([19] states the recursion without explanation).

$$\mathbf{A}_n = \begin{bmatrix} \mathbf{A}_{n-1} & \mathbf{B}_{n-1} \\ \mathbf{B}_{n-1}^T & \mathbf{0} \end{bmatrix} \mathbf{B}_{n-1} = \begin{bmatrix} \mathbf{A}_{n-2} \\ \mathbf{B}_{n-2}^T \end{bmatrix}$$

where $\mathbf{0}$ is the zero matrix.

This recursion is subject to the initial conditions that

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

This simple recursion for $\mathbf{A}_n$ allows us to construct successive $\mathbf{A}_n$ very quickly. Recursions for the other constraints in Figure 1 can be found in the paper by Weeks and Blahut [18].

If we assume that C_n has the form of a polynomial in inverse powers of n,

$$C_n = \sum_{k=0}^{\infty} \frac{a_k}{n^k}$$

then we may truncate this series at $k = N$ and use the data $C_n, C_{n+1}, \ldots, C_{n+N}$ to determine the a_k. As n approaches infinity, C_n approaches a_0; therefore, a_0 serves as the best estimate of C. This technique is known as Richardson extrapolation, and for a given N, it turns out that (see [1])

$$a_0 = \sum_{k=0}^{\infty} \frac{C_{n+k}(n+k)^N(-1)^{k+N}}{k!(N-k)!}$$

For all constraints, the best estimates of the capacity, along with the number of digits of precision in the estimate, have been listed in Table 1.

If $\mathbf{A}_n$ is real and symmetric (conditions met by the square and diamond constraints), we may use results from Calkin and Wilf [2] in conjunction with the recursive structure of $\mathbf{A}_n$ to produce tight upper and lower bounds on the capacity. We have the following as a lower bound on C:

$$C \geq \frac{\log_2 \left[\frac{\lambda_{p+2q+1}}{\lambda_{2q+1}}\right]}{p}$$

where p is a positive integer and q is a nonnegative integer. The following is an upper bound:

$$C \leq \frac{\log_2[\lambda'_p]}{p}$$

where λ' is the eigenvalue associated with the adjacency matrix of a checkerboard constraint defined on a cylinder of circumference p, where p is a positive even integer. These bounds are due to Calkin and Wilf [2] and usually improve capacity estimates by narrowing the bound window.

3 Full-Surface Data-Storage Model

In a full-surface data-storage system model the user data $\mathbf{u}$ is a one-dimensional signal over a two-dimensional index set, i.e., a matrix of data values. The data value in the ith row and jth column of $\mathbf{u}$ is denoted by $u_{i,j}$. The data $\mathbf{u}$ has N_x rows and N_y columns. The channel introduces ISI when the data is transmitted. The noise-free output of the channel, denoted $\mathbf{v}$, is modeled as the two-dimensional convolution of the user data $\mathbf{u}$ and the channel model $\mathbf{c}$. The noise-free signal is written $\mathbf{v} = \mathbf{c} * \mathbf{u}$. The channel $\mathbf{c}$ has γ_x rows and γ_y columns. To calculate any single value of $\mathbf{v}$, the following expression can be used:

$$v_{i_0,j_0} = \sum_i \sum_j c_{i_0-i,j_0-j} u_{i,j}$$

Table 1: Estimates of 2-D capacity

Constraint	Capacity	Precision
Diamond	.587891161775	10^{-12}
Hexagonal	.482644	10^{-6}
Square	.4250	10^{-4}
Diamond II	.350308	10^{-6}
Hexagonal II	.2775	10^{-4}
Square II	.236	10^{-3}
Diamond III	.241	10^{-3}
Hexagonal III	.181	10^{-3}
Square III	.148	10^{-3}

The channel model **c** characterizes the amount and extent of ISI in the full-surface channel. The values $c_{i,j}$ are computed according to [12]

$$c_{i,j} = \exp\left\{-\frac{((i-i_c)\delta_x)^2}{2\sigma_c^2}\right\}\exp\left\{-\frac{((j-j_c)\delta_y)^2}{2\sigma_c^2}\right\}, \tag{3.1}$$

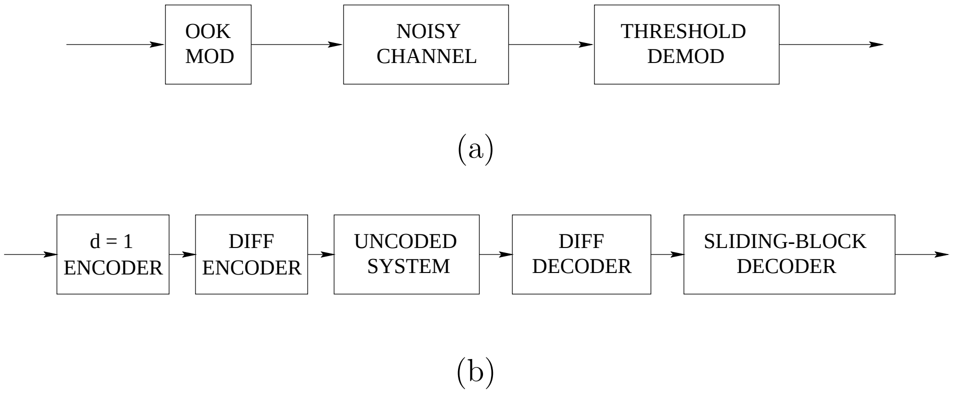
where $i\,\epsilon\,\{0,1,\ldots,\gamma_x-1\}$, $j\,\epsilon\,\{0,1,\ldots,\gamma_y-1\}$, $i_c=\frac{\gamma_x-1}{2}$, $j_c=\frac{\gamma_y-1}{2}$ (it is assumed that γ_x and γ_y are odd and that the center sample has an amplitude of 1), and δ_x and δ_y are the physical separation of bits in the vertical and horizontal directions, respectively. The channel model is the smallest rectangular grid of samples such that no sample outside the rectangular grid exceeds 1% of the center sample.

This definition of the channel model allows the definition of unitless data-separation parameters. Define

$$\Delta_x = \frac{\delta_x}{\sigma_c},\ \Delta_y = \frac{\delta_y}{\sigma_c}.$$

Data with separation parameters Δ_x and Δ_y has density given by $\frac{1}{\Delta_x\Delta_y}$. This "dimensionless" density has units of $\frac{\text{bits}}{\sigma_c^2}$. To compute physical density the unitless density is multiplied by σ_c^2.

Samples of a Gaussian random variable corrupt the noise-free output **v**, which results in the received signal $\mathbf{r}=\mathbf{v}+\mathbf{n}$. Elements of **n** are independent Gaussian variables with mean zero and variance $N_0=\sigma_n^2$. The *signal-to-noise ratio* (SNR) is given in decibels by $10\log_{10}\frac{E_b}{N_0}$, where $E_b=\frac{||\mathbf{c}||^2}{4}$ is the variance of the noise-free signal and $N_0=\sigma_n^2$, is the variance of the noise, and $||\mathbf{c}||$ is the Frobenius norm [7] (F-norm) of **c**, i.e., $||\mathbf{c}||^2=\sum_{i,j}c_{i,j}^2$.



(a)

(b)

Figure 2: Block diagram for the competing systems: (a) uncoded system and (b) coded system.

4 Threshold Demodulation

In the uncoded full-surface data-storage system in Figure 2, an ON-OFF keyed (OOK) modulator sends unconstrained data across a noisy, full-surface channel. A threshold device performs demodulation by comparing received values with an optimized threshold. Bit decisions are made from the received values on an individual basis. If a received bit is less than (greater than) the threshold value, that bit is decided to be a *zero* (*one*).

In the one-dimensional coded system in Figure 2, a finite-state machine encodes user data to satisfy a $d = 1$ runlength-limited (RLL) code along every row. The encoder operates at rate 2/3, and by definition every output *one* in the RLL code is surrounded by at least one *zero* on both the left and right. Then, a differential encoder translates *ones* into data transitions; i.e., denoting the output of the differential encoder as $\mathbf{y}$ and the input as $\mathbf{x}$, the differential encoder implements $y_k = y_{k-1} + x_k$, where addition is performed modulo-2, and it is understood that input and output symbols are either binary *zero* or binary *one*. The RLL encoder and differential encoder ensure that at any place in the channel sequence any *one* or *zero* is always accompanied by another *one* or *zero* before or after. This grouping of elements widens the threshold eye and hence improves the BER. Next, an OOK modulator transmits the coded data across a noisy channel, a threshold device performs demodulation, a decoder inverts the differential code, and a sliding-block device decodes the $d = 1$ RLL code. The $d = 1$ RLL encoder and sliding-block decoder are based on state-splitting algorithms [14].

The threshold is set midway between the lowest noise-free sample attributable to a *one* and the highest noise-free sample attributable to a *zero*. More precisely, the optimized threshold is given by

$$t_{opt} = \text{average}(\max_{\mathbf{u}|u_{i,j}=0}(v_{i+i_c,j+j_c}), \min_{\mathbf{u}|u_{i,j}=1}(v_{i+i_c,j+j_c})),$$

where (i_c, j_c) is the offset of the center peak from the upper-left corner of the channel model matrix, i.e., $(i_c, j_c) = \mathrm{argmax}_{(i,j)} c_{i,j}$. This optimized threshold is computed assuming a data set of infinite size so that edge effects may be neglected, and t_{opt} does not depend on the choice of i and j.

If $\max_{\mathbf{u}|u_{i,j}=0}(v_{i+i_c,j+j_c}) < \min_{\mathbf{u}|u_{i,j}=1}(v_{i+i_c,j+j_c})$, threshold demodulation is reliable in the sense that as SNR approaches infinity, the BER approaches zero. This criterion provides a simple test for reliable demodulation at a given data density.

Since data takes values in the set $\{0, 1\}$ and since channel coefficients are nonnegative, the maximum and minimum values for the uncoded threshold demodulation system can be calculated as

$$\min_{\mathbf{u}|u_{i,j}=1}(v_{i+i_c,j+j_c}) = c_{i_c,j_c} = 1$$

$$\max_{\mathbf{u}|u_{i,j}=0}(v_{i+i_c,j+j_c}) = \left(\sum_{i,j} c_{i,j}\right) - c_{i_c,j_c} = \left(\sum_{i,j} c_{i,j}\right) - 1.$$

For the coded threshold demodulation system, the minimum value is calculated as

$$\min_{\mathbf{u}|u_{i,j}=1}(v_{i+i_c,j+j_c}) = c_{i_c,j_c} + c_{i_c,j_c+1},$$

and the maximum value is calculated as

$$\max_{\mathbf{u}|u_{i,j}=0}(v_{i+i_c,j+j_c}) = \left(\sum_{i,j} c_{i,j}\right) - c_{i_c,j_c} - c_{i_c,j_c+1}.$$

Performing these calculations in the absence of noise shows that one-dimensional storage systems can achieve 16% linear data density improvement (at BER=0) by using $d = 1$ RLL coding. In the full-surface system, coding along rows improves areal data density by only 9%. These percentages for data density include the effects of data density-reducing codes and show that improvement in BER outweighs the rate cost of using codes (at high SNR). The intertrack interference accounts for the reduction in data density improvement between serial and full-surface systems.

Further density improvements are realized by using checkerboard constraints [18]. When used with a differential encoder, these constraints work to prevent ISI from causing erroneous demodulation. Comparing the coded systems with uncoded systems yields the density improvements in Table 2. These numbers reflect data density that has been adjusted by code rate and assume the existence of encoders and decoders that can come arbitrarily close to capacity values calculated by Weeks and Blahut [18]. Such encoders and decoders can be designed by using a multitrack approach [8], though issues of complexity and suboptimality loom. It is also assumed that a differential encoder exists that guarantees large blocks of *ones* and *zeros*. Designing capacity-achieving encoders, decoders, and appropriate differential encoders remains an open problem. Because such encoders and decoders have not been designed, the density improvement values in Table 2 provide only a theoretical estimate.

Table 2: Density Improvement for Various Coding Schemes

Constraint	% Improvement	Sampling
$d = 1$ RLL (serial)	16	center
$d = 1$ RLL (full-surface)	9	center
Diamond I	11	center
Hexagonal I	29	center
Square I	31	side
Diamond II	27	side
Hexagonal II	27	mixed
Square II	57	center
Diamond III	58	center
Hexagonal III	84	mixed
Square III	72	side

The numbers in Table 2 show that as the extent of the isolation of *ones* gets larger, greater density improvement is possible. Correspondingly, larger constraints require greater computational complexity to implement encoders and decoders. The column labeled "Sampling" denotes the type of sampling across the Gaussian PSF. "Center" sampling is the standard sampling in Equation (3.1). With "side" sampling the Gaussian function has been sampled on a grid that is displaced by $\frac{1}{2}\delta_x$ in the vertical direction and $\frac{1}{2}\delta_y$ in the horizontal direction from the center sampling. Finally, displacing either the horizontal or vertical axes from the center sampling results in "mixed" sampling.

Figure 3 displays density improvement for the modeled optical data storage system that stores a single row of data and demodulates at a fixed BER of 10^{-3}. These curves show linear density performance, and the coded system yields 19% improvement over the uncoded system. This result is close to the predicted values. The discrepancy may be due to the fact that the simulation results operated at a BER of 10^{-3} and the theory predicted the density improvement at a BER of zero. Furthermore, the decoders can occasionally correct errors. This effect was neglected in the theoretical computation. Note the crossover point at low SNR. This crossover is a common characteristic of coded systems. At high noise levels, errors have a greater impact on the coded systems that rely on block decoders, whereas in the uncoded systems, errors remain isolated.

Theoretical calculations for full-surface data show that the maximum possible density improvement of the coded system over the uncoded is 9.1%. Indeed, the simulation results in Figure 4 verify this theoretical result. The improvement at high SNR is 9.2%. Figure 4 shows only one curve for each system; however, many curves can be generated for each system by varying the ratio of δ_x to δ_y. For the coded system, a value of $r = 2.5$ achieves the best density performance where $r = \frac{\delta_x}{\delta_y}$. This result implies that in the coded system

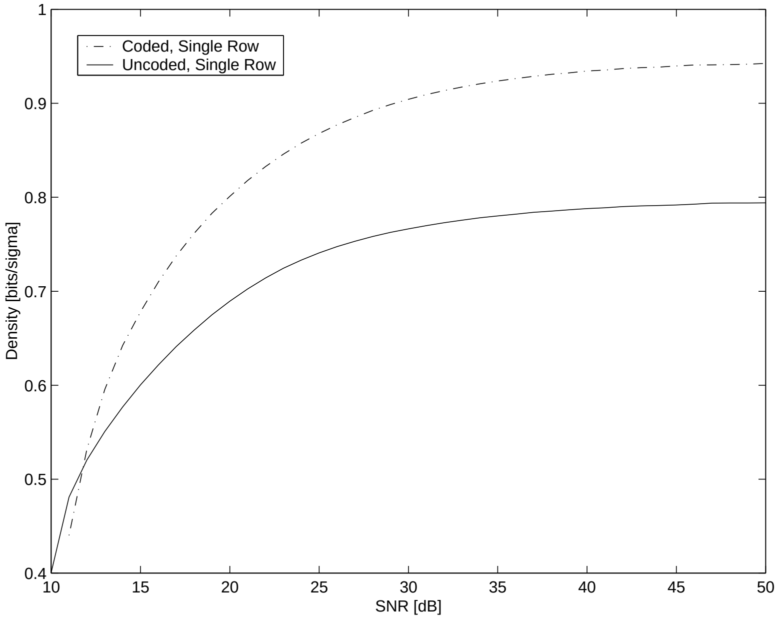


Figure 3: Simulation curves for a single row of data with coded and uncoded threshold demodulation, BER = .001.

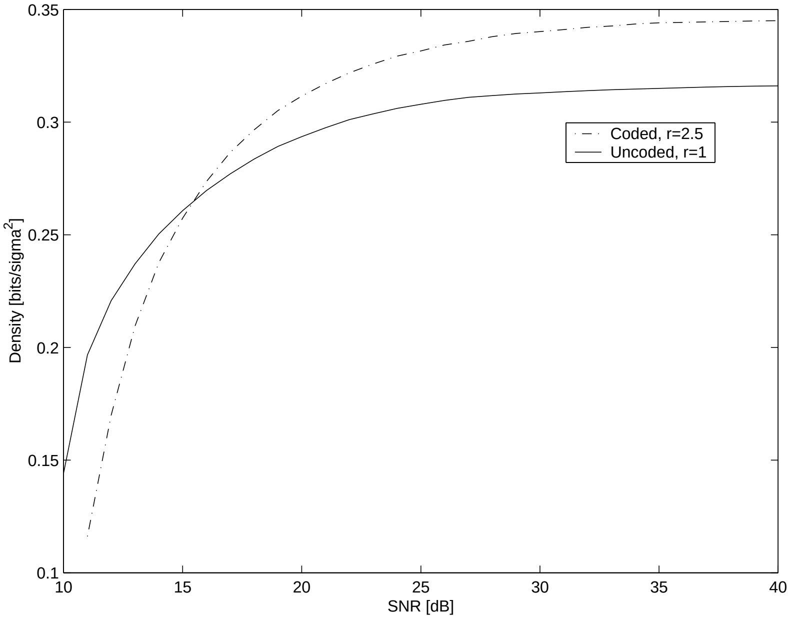


Figure 4: Simulation curves for full-surface data with coded and uncoded threshold demodulation, BER = .001.

where coding occurs along rows, the best performance is achieved when the data bits are closer along rows than along columns.

Therefore, $d = 1$ RLL coding on full-surface systems achieves modest density improvements. The density improvement is significantly less than for serial systems. The law of diminishing returns applies to this full-surface application, and the intertrack interference plays a significant role in this reduction. Nevertheless, the full-surface system can gain 9% in data density at minimal complexity cost under the Gaussian blur model, and the overall data density of the full-surface system is better than a serial system where tracks are stored in a two-dimensional media. Finally, checkerboard constraints show much promise for improving data density under simple threshold demodulation if some of the remaining open problems in codec design can be solved.

5 Conclusion

The capacity of various checkerboard codes has been estimated using matrix recursions, eigenvalue computations, and upper and lower bounds. These capacity values lead to predicted theoretical density improvements in full-surface data-storage systems that have been verified for simple cases. While the existence of encoders and decoders for constrained two-dimensional codes has been established, work remains to develop differential encoders and decoders.

References

[1] C. M. Bender and S. A. Orszag. *Advanced Mathematical Methods For Scientists and Engineers*. McGraw-Hill, Inc., St. Louis, 1978.

[2] N. Calkin and H. Wilf. The number of independent sets in a grid graph. *SIAM Journal of Discrete Mathematics*, 11(1):54–60, 1998.

[3] K. Engel. On the fibonacci number of an m x n lattice. *Fibonacci Quarterly*, 28(1):72–78, 1990.

[4] T. Etzion. Cascading methods for runlength-limited arrays. *IEEE Transactions on Information Theory*, 43(1):319–323, January 1997.

[5] S. Forchhammer and J. Justesen. Entropy bounds for constrained two-dimensional random fields. *IEEE Transactions on Information Theory*, 45(1):118–27, January 1999.

[6] S. Forchhammer and J. Justesen. Bounds on the capacity of constrained two-dimensional codes. *IEEE Transactions on Information Theory*, 46(7):2659–66, November 2000.

[7] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996.

[8] S. Halevy and R. M. Roth. Parallel constrained coding with application to two-dimensional constraints. *IEEE Transactions on Information Theory*, 48(5):1009–1020, May 2002.

[9] J. Justesen and Y. M. Shtarkov. Combinatorial entropy of images. *Problemy Peredachi Informatsii*, 33(1):3–11, Jan-Mar 1997.

[10] A. Kato and K. Zeger. On the capacity of two-dimensional run-length constrained channels. *IEEE Transactions on Information Theory*, 45(5):1527–40, July 1999.

[11] A. Kato and K. Zeger. Partial characterization of the positive capacity region of two-dimensional asymmetric run length constrained channels. *IEEE Transmactions on Information Theory*, 46(7):2666–70, November 2000.

[12] B. M. King and M. A. Neifeld. Parallel detection algorithm for page-oriented optical memories. *Applied Optics*, 37(26):6275–6298, 10 September 1998.

[13] D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, New York, 1995.

[14] B. H. Marcus, P. H. Siegel, and J. K. Wolf. Finite-state modulation codes for data storage. *IEEE Journal on Selected Areas in Communication*, 10(1):5–37, January 1992.

[15] Z. Nagy and K. Zeger. Capacity bounds for the three-dimensional (0,1) run length limited channel. *IEEE Transactions on Information Theory*, 46(3):1030–3, May 2000.

[16] R. E. Swanson and J. K. Wolf. A new class of two-dimensional runlength-limited recording codes. *IEEE Transactions on Magnetics*, 28(6):3407–16, November 1992.

[17] B. V. Vasic, S. W. McLaughlin, and O. Milenkovic. Shannon capacity of m-ary redundant multitrack runlength limited codes. *IEEE Transactions on Information Theory*, 44(2):766–74, March 1998.

[18] W. Weeks, IV and R. E. Blahut. The capacity and coding gain of certain checkerboard codes. *IEEE Transactions on Information Theory*, 44(3):1193–1203, May 1998.

[19] Z. Ye and Z. Zhang. On the capacity rates of two-dimensional runlength limited codes. In *Proceedings of the 1995 IEEE International Symposium on Information Theory*, page 247, New York, NY, 1995.