IPSJ Transactions on Databases Vol.5 No.2 17–25 (June 2012)

Regular Paper

Error Control for High-density Monochrome Two-dimensional Barcodes

Ramon Francisco Mejia[1,a)] Yuichi Kaji[1] Hiroyuki Seki[1]

Received: December 20, 2011, Accepted: April 6, 2012

Abstract: Monochrome two-dimensional barcodes are rapidly becoming a de-facto standard for distributing digital data through printed medium because of their small cost and portability. Increasing the data density of these barcodes improves the flexibility and effectiveness of existing applications of barcodes, and has the potential to create novel means for data transmission and conservation. However, printing and scanning equipment introduce uncontrollable effects on image at very refined level, and the effects are beyond the scope of the error control mechanisms of existing barcode schemes. To realize high-density barcodes, it is essential to develop novel symbology and error control mechanisms which can manage these kinds of effects and provide practical reliability. To tackle this problem, this paper studies the communication channel defined by high-density barcodes, and proposes several error control techniques to increase the robustness of the barcode scheme. Some of these techniques convert the peculiar behavior of printing equipment to the well-studied model of additive white Gaussian (AWGN) channel. The use of low-density parity codes is also investigated, as they perform much better than conventional Reed-Solomon codes especially for AWGN channels. Through experimental evaluation, it is shown that the proposed error control techniques can be essential components in realizing high-density barcodes.

Keywords: two-dimensional barcodes, low-density parity check codes, error control, automatic identification and data capture

1. Introduction

In data management systems, storing data in *two-dimensional* (*2D*) *barcodes* [7] has some advantages over digital media. Applications which use 2D barcodes mainly rely on its high data capacity (compared to one-dimensional barcodes) and portability to provide better functionality. More than store indices to external databases (e.g., ID numbers, product codes, or website URLs), 2D barcodes can store a small database as well (e.g., contact information, registered mail or a flight ticket data). The ability of 2D barcodes to embed data on a printed symbol allows data to be distributed in many physical locations. Furthermore, a printed barcode survives for a long time (possibly decades), even if no additional costs are paid for maintenance and conservation. Because of these favorable properties, 2D barcodes are used in many applications where data sources are decentralized, such as health care [15], government [3], e-commerce [5] and other data management systems.

Past studies on barcode capacity mainly focused on improving the *symbology* of 2D barcodes, i.e., enhancing its structure and processing methods to increase data capacity while retaining its compact size and portability. Some technologies investigate colored data cell designs in order to increase the amount of data represented per cell [6], [13]. Using colored cells increases the density of barcodes, but high-quality color printers are expensive in general, and the degradation of colors over time can make it difficult to retrieve the original data. It will be more versatile and cost-efficient to develop *monochrome 2D barcodes*, because the equipment used to produce monochrome symbols are generally less expensive to acquire and operate, and symbols can be kept in storage medium that are cheaper and easier to maintain, such as paper products.

Another approach to improve the symbology of 2D barcodes is through increasing the *data density* of a barcode symbol. Data density is defined as the amount of data stored per unit area of a symbol; for instance, the number of cells printed per square millimeter (assuming that cells represent binary values). However, one inherent issue of storing digital data in a physical format is its decreased robustness against errors. Data stored in high-density monochrome barcodes are susceptible to errors caused by printing and scanning hardware, or physical damage inflicted on the symbol or its storage medium.

To enhance reading robustness, most 2D barcode symbologies use *Reed-Solomon* (RS) codes [11] to encode data prior to generating the symbol. In recent years, it has been shown that well-designed *low-density parity check* (LDPC) codes [4] perform well compared to RS codes in many communication channels. A remarkable aspect of LDPC codes is that we can perform *soft-decision decoding* for LDPC codes with almost linear-time complexity. Soft-decision decoding is an algorithm for error correction in which inputs to the algorithm can have continuous values. It is more powerful than *hard-decision decoding* algorithms in which inputs are quantized into two level, but requires very large computational complexity in general. Soft-decision decod-

[1] Nara Institute of Science and Technology, Ikoma, Nara 630–0192, Japan
[a)] ramon-m@is.naist.jp

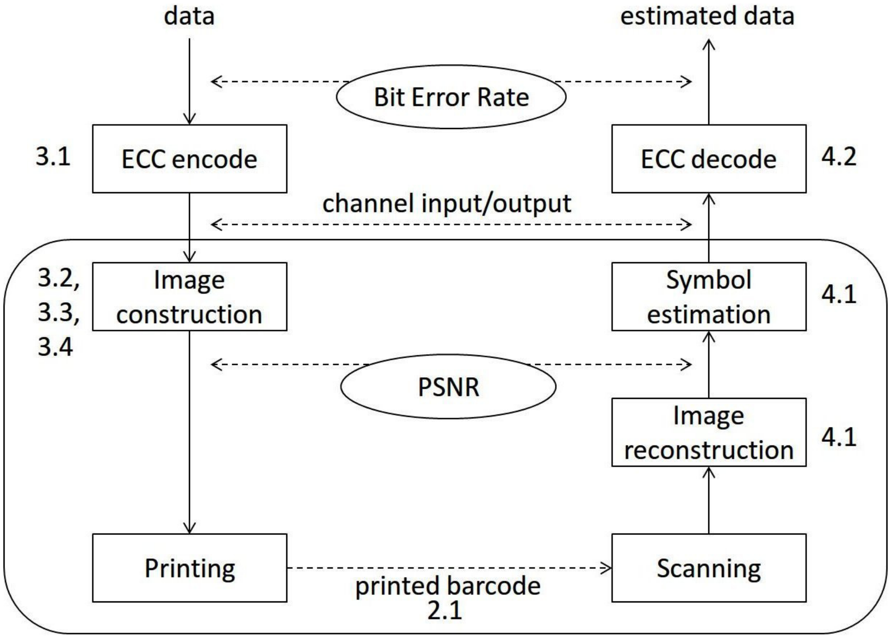


Fig. 1 System diagram of the barcode system.

ing was considered to be impractical for many years, but the sparse nature of LDPC codes allow efficient algorithms of soft-decision decoding to be implemented. Because of this, LDPC codes show a much better performance than conventional error-correcting codes, including RS codes [1], [2].

The goal of this research study is to examine error control techniques in order to improve the robustness of monochrome 2D barcodes with high data density. Particularly, this paper analyzes the effectiveness of LDPC codes compared to RS codes. Understanding these factors can deepen our knowledge of high-density barcodes and lead to design improvements for symbologies and their applications. Note that the symbology proposed in this paper is not meant to be an independent 2D barcode standard, but rather serve as a mechanism to develop and test fundamental techniques for reliable high-density barcodes.

The rest of this paper is structured as follows. Section 2 describes the channel model of a high-density 2D barcode, along with factors which cause errors in the channel. Then, the encoding and printing procedures of the experimental symbology are outlined in Section 3. Section 4 outlines the scanning and decoding procedures of the symbology. Finally, the methodology and results of evaluating the symbology are discussed in Section 5.

2. Channel Model of High-density 2D Barcodes

The "communication channel" defined by a barcode system is not a simple binary digital channel. In order to understand how to protect data in barcodes using LDPC codes, the *channel model* exhibited by the symbology must first be investigated.

The system diagram of our proposed high-density barcode system is shown in **Fig. 1**, though several components which have little relation to error control are not shown in the figure. Given the data to be stored in a barcode symbol, we first perform an encoding process of an error-correcting code. In this study, LDPC codes considered for this process are discussed in Section 3.1. The result of the encoding is then passed to the image construction process. Basically, data bits are represented as solid squares called *data cells*, where a white cell signifies the bit value '0' and a black cell signifies the bit value '1'. Additionally, several techniques discussed in Sections 3.2, 3.3 and 3.4 are employed to attain robustness in the printed image. The image construction process determines the barcode image which is given to the printing device such as a laser printer.

To access the stored data, the printed barcode image is scanned using a flatbed scanner. Because the printed image is expected to be disturbed by several factors, which are discussed in Section 2.1, we first need to reconstruct the original image, and then need to estimate the value of each symbol (data bit) from the reconstructed image. In this study, the estimated value of a symbol is simply computed as the *ratio* of black pixels over the total number of pixels in an (estimated) cell area. These issues are addressed in the Section 4.1. The estimated symbols are given to the LDPC decoder (Section 4.2), and we finally obtain the estimated data. From the viewpoint of encoder/decoder of LDPC codes, these estimated values can be regarded as the output of the communication channel, while the binary (0 or 1) result of ECC encoder should be regarded as the channel input.

As Fig. 1 shows, the error control mechanism consists of several components. To evaluate the effectiveness of each component, we focus on two empirical measures: *peak signal-to-noise ratio* (PSNR) for evaluating the image reconstruction process, and *bit error rate* (BER) [8] for evaluating the use of LDPC codes.

(1) Obviously, an LDPC code performs well if the input to the decoder is accurate. Because the input to the decoder is derived from the reconstructed image of the barcode, it is important to observe the quality of image reconstruction. To statistically measure quality of image reconstruction, we calculate the PSNR. First, the *mean squared error* (MSE), a

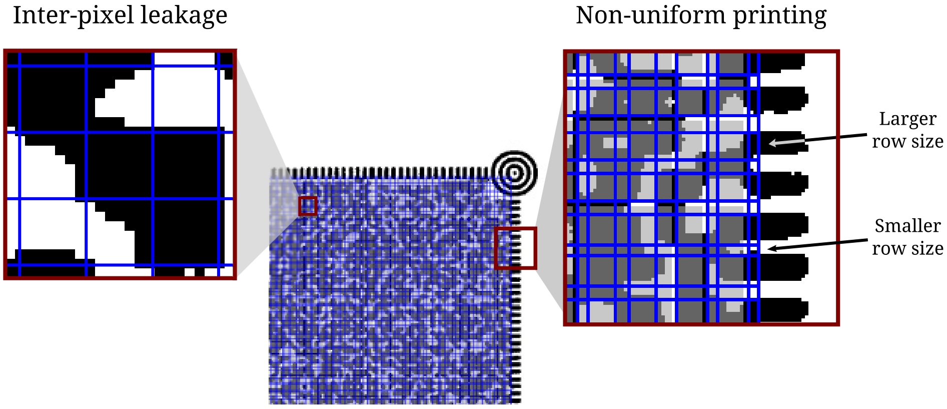


Fig. 2 Example of a symbol containing inter-pixel leakage and non-uniform printing of rows.

statistical measure of error used to compare two $m \times n$ images, is computed. Given the image of data cells (channel input) and their corresponding scanned images (channel output), the MSE of data cell $c \in C$ is computed as the squared difference of the expected value $val(c)$ of the cell and each pixel of the sampling cell img_c; specifically, MSE is computed as

$$MSE(c) = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [val(c) - img_c(i, j)]^2 \tag{1}$$

Given the MSE of all data cells in a symbol, the PSNR of the symbol is computed as

$$PSNR = 20 \cdot log_{10} \left(\frac{MAX_{img}}{\sqrt{\sum_{i=0}^{|C|} MSE(i)}} \right) \tag{2}$$

where $MAX_{img} = 1.0$, the maximum value of ratio r.

(2) The bit error rate is a direct measure of the quality of data transmission, but we need to remark that the decoding algorithm of LDPC code must be provided with the statistics of the communication channel, which is needed in the internal belief-propagation decoding. Through preliminary experiments, we found that the communication channel defined as above can be modeled as an additive white Gaussian noise (AWGN) channel, and the variance of the Gaussian distribution can be determined heuristically from the result of the preliminary experiments. In the preliminary experiments, we know the input and the output of the channel. We can collect samples O_{black} of ratios which corresponds to black cells and samples O_{white} of ratios which corresponds to white cells. The variance of each set is then calculated as

$$\sigma^2 = \int P(x)(x - \mu)^2 \, \mathrm{d}x \tag{3}$$

where x is either O_{black} or O_{white}, $P(x)$ is the probability distribution of x, and μ is the population mean of x.

2.1 Sources of Errors

We review the phenomena and factors which can cause problems in realizing a high-density barcode scheme. The factors can be classified into two types: *device-oriented factors* and *external factors*. Device-oriented factors are mainly caused by inaccuracies during printing and scanning barcodes using regular office equipment. These inaccuracies are not noticeable in regular use, but are significant if these devices are utilized to its performance limit. One such factor is called *inter-pixel leakage*. Data cells in high-density barcodes are printed in small sizes (less than $0.3 \times 0.3\,\mathrm{mm}^2$). As a result, the laser printer toner used to draw black cells splatter uncontrollably and blot neighboring white cells, thus becoming a major source of noise in the channel.

Another factor is called *non-uniform printing*, where cells along the same row were printed with similar heights, but cells in other rows were printed with different heights. This factor was also observed for the widths of columns. It is conjectured that this phenomenon is introduced by the mechanical constraints of the printer or scanner, and are not avoidable unless we use professional quality image setter. In the usual home and office environment, we need to expect that a printed barcode image is not as precise and uniform as stated in the specifications of the devices. **Figure 2** shows an example of both factors.

External factors are caused by the environment where 2D barcodes are placed. Harsh conditions can introduce physical defects on printed symbols. For example, an ink blot or stain on the symbol affects a continuous sequence of data cells in neighboring rows and columns. Thus, if data is also stored in a sequential manner (e.g., left-to-right, top-to-bottom), a defect can introduce a consecutive stream of erroneous bits (known as a *burst error*). External factors are a problem for regular-sized barcode schemes, and it becomes more problematic if a scheme relies on higher resolutions of a scanned barcode image.

To mitigate these factors, we design a symbology that employs several error control techniques. Device-oriented factors are addressed using a margin factor parameter and a tweaked design of the timing pattern which surrounds data area. To cope with external factors, we developed a strategy for interleaving which stores data in a non-contiguous way. This creates a more uniform distribution of errors across codewords when a symbol is partially damaged. Details of these techniques are discussed in the next section.

Despite these countermeasures, it is still difficult to accurately determine whether a scanned data cell is black or white. To be able to maximize the robustness of a symbol, we investigate the use of powerful LDPC codes with soft-decision decoding, instead of the conventional RS codes with hard-decision decoding.

3. Encoding and Printing

Conventional 2D barcodes employ symbologies which are good for fast reading. In this paper, however, we pursue a symbology which increases data density; that is, to safely accommodate as many data cells as possible in a given area. The following subsections describe the encoding and printing steps to create a symbol.

3.1 Encoding

First, the input data to be stored in a barcode symbol is encoded using error-correcting codes. Most 2D barcodes use RS codes for error correction, which is a conventional technique and thus details are not discussed in this paper. For an in-depth discussion of encoding using RS codes, see Wicker and Bhargava [14].

This study investigates the usage of LDPC codes, specifically the family of codes designed for the IEEE 802.16e standard (also known as mobile WiMAX). Using these codes is advantageous since they have smaller complexity for encoding, compared to other classes of LDPC codes. In general, the encoding operation of an LDPC code requires quadratic-order complexity in the code length; however, the IEEE codes defined in the standard are designed so that they have *quasi-cyclic* structure, which enables the realization a linear-order encoding algorithm. Another advantage of these codes is that the code parameters can be changed in a flexible manner. The standard defines several classes of LDPC codes with code rates 1/2, 2/3, 3/4 and 5/6, and code length ranges from 576 to 2,304 bits. These parameters have a strong relation to the efficiency and the error-correcting capability of the code [9]. We note however that we do not have to restrict ourselves to this family of LDPC codes. The techniques developed in this paper can be used for any LDPC codes.

3.2 Data Cells and Data Area

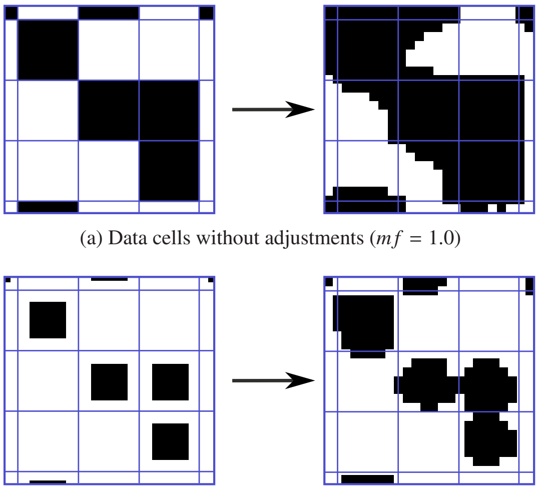
A monochrome [*1] 2D barcode stores codewords in data cells arranged inside a $25.4 \times 25.4\,\mathrm{mm}^2$ space or *data area*. We can consider larger data area spaces, though we fix the size in this study to make the following discussion simple and clear. Data cells inside the data area are arranged in a $dim{\times}dim$ matrix, where *dim* is an integer called the *dimension* parameter. Increasing the dimension parameter of a fixed-size data area has two effects on the symbol:

(1) **Data capacity and density**: The number of data cells available in the data area [*2] is increased. This is computed as slightly less than dim^2 because of the overhead introduced by the symbology. Because the size of the data area is fixed, data capacity can be regarded as the density of data: dim^2 cells per inch or $(dim^2/25.4)^2$ cells per mm.

(2) **Cell size** or printing size: The cell size of a data cell must be reduced in order to increase the data density of the symbol.

Table 1 Data capacity, density, and printing size of a cell for a symbol with dimension *dim*.

dim	Capacity (no. of cells)	Density (cells/ mm^2)	Cell size (mm^2)
57	3,053	127.9	0.446
67	4,293	127.9	0.379
77	5,733	176.3	0.330
87	7,373	233.4	0.292
97	9,213	304.9	0.262
107	11,253	378.1	0.237
117	13,493	450.7	0.217



(a) Data cells without adjustments ($mf = 1.0$)

(b) Data cells with margin factor ($mf = 0.6$)

Fig. 3 Effect of inter-pixel leakage on $0.237 \times 0.237\,\mathrm{mm}^2$ data cells.

Cell size is computed $(25.4/dim) \times (25.4/dim)\,\mathrm{mm}^2$. **Table 1** lists the relation of dimension, data capacity and density, and cell size.

To cope with inter-pixel leakage, the printing size of black cells are reduced by a certain *margin factor* relative to the size of the data cell. **Figure 3** shows data areas printed with a margin factor value of $mf = 1.0$ and $mf = 0.6$. Notice how black cells printed with $mf = 0.6$ contain a white border which acts as a buffer for inter-pixel leakage. The white cells in this case have less noise generated by adjacent black cells. Thus, the data area is more distinguishable than the data area printed with $mf = 1.0$. The optimum choice of margin factor depends on the size of the cells and the resolution of the printer. It is expected that the optimum margin factor is determined in advance for the environment where barcode images are produced.

3.3 Finder and Timing Patterns

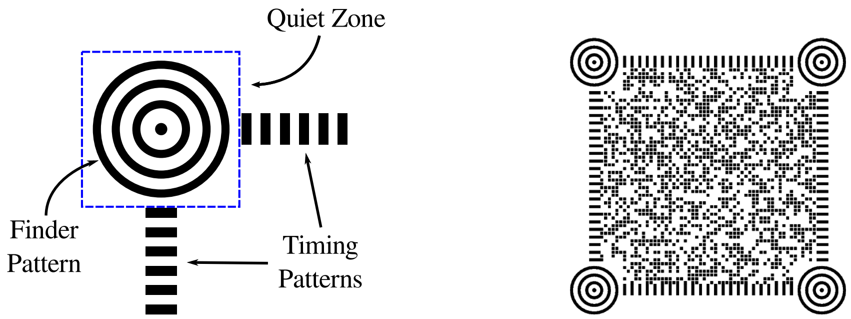
A set of *finder patterns* is used to isolate the barcode symbol from a scanned image, and a set of *timing patterns* is used to estimate the size and location of data cells. **Figure 4** (a) shows the individual design elements of the high-density symbology.

Finder patterns are placed on the four corners of the data area. It consists of three black rings co-centric to the corner of the data area, marking the corner with a single black cell. A *quiet zone* of white cells is placed underneath the finder pattern to improve detection accuracy.

The timing pattern is a consecutive series of modules located around the four edges of the data area. A timing pattern connects two finder patterns, with each module alternating between white

[*1] The term “monochrome” means that barcodes are printed and scanned as black-and-white images. Although grayscale images contain more information that may assist in barcode detection, producing small data cells in grayscale generate too much noise in the channel.

[*2] This definition does not take into account other techniques which would increase the total data capacity of a barcode symbol (such as data compression).



(a) Finder pattern, quiet zone and two timing patterns.

(b) Example of a symbol with a data area.

Fig. 4 Design elements and a complete symbol generated by the symbology.

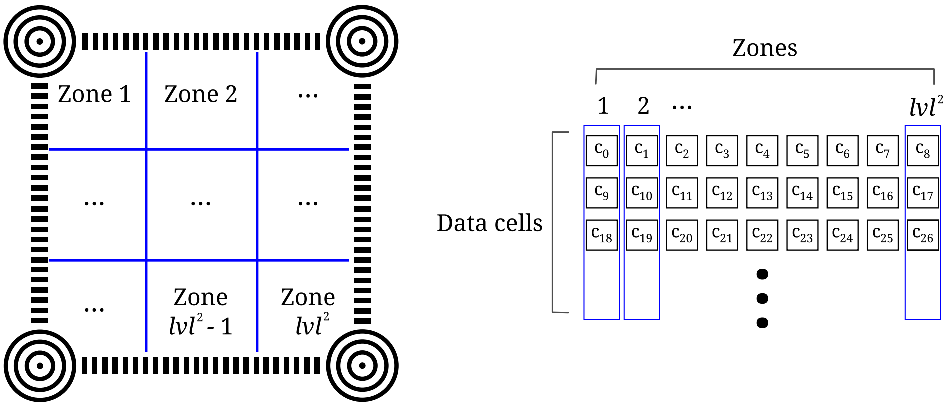


Fig. 5 Example of interleaving using interleave level $lvl = 3$.

and black cells. To increase the detection reliability, each timing pattern module is made up of three connected black or white cells. Timing patterns are also printed on the same row or column as data cells. Because of this, timing pattern modules are also subject to non-uniform printing. This is advantageous because the separation of data cells (indicated by the blue lines in Fig. 2) also adjusts to the non-uniform printing size of the rows and columns, and thus aiding the detection of data cells.

3.4 Interleaving

Finally, before a symbol is printed on a sheet of paper, the data cells inside the data area are interleaved. The interleaving technique designed for this symbology is illustrated in **Fig. 5**. Given the *interleave level* (denoted as lvl) of the symbol, the data area is first partitioned into lvl^2 sub-divisions or *zones*. These zones contain an equal number of data cells except for zones which overlap quiet zones. The cell c_i, which represents the i-th bit of data, is then written in the $(i \bmod lvl^2) + 1$ zone. In other words, data cells are placed into zones one by one, starting from zone 1 to lvl^2. This process repeats until all data cells have been stored.

When data cells are inserted into zones, the interleave level affects the distance between the cells. When there is no interleaving, the distance between two cells is 1 since they are stored sequentially. For symbols with interleaving, data cells are stored apart from each other with a computed distance

$$dist(dim, lvl) = \lfloor dim/lvl \rfloor \tag{4}$$

4. Scanning and Decoding

4.1 Symbol Processing

When data is needed from a barcode, the sheet of paper is scanned using a flatbed scanner and codewords are read from the symbol. We refer to this process as *symbol processing*, which involves the following steps:

(1) Barcode symbols (if any) are searched in the scanned image. The finder patterns of a symbol are located using a basic template matching algorithm [12]. The centers of all finder patterns are then computed.

(2) Lines connecting the centers of adjacent finder patterns are connected, and the resulting closed rectangle is masked. This enhances the detection of timing patterns in the following step.

(3) The lines connecting adjacent finder patterns are scanned through pixel by pixel. The path each line passes through is also the location of a timing pattern. When the value of the pixel changes from white to black, this pixel is marked as a *transition point*.

(4) Transition points from opposite timing patterns are connected, forming a grid of *sampling cells*.

(5) The ratio r of black pixels in each cell is determined. From the grid of sampling cells computed in (4), we can determine the set P of pixels which are used to represent one particular cell. The ratio r of this cell is simply the number of black pixels in P over the total number of pixels in P.

(6) The ratio r obtained from each sampling cell is mapped to a *soft value* using the function $f(r) = -(2r - 1)$. Soft values

are de-interleaved and grouped into codewords.

Steps 1 to 4 constitute the "image reconstruction" process in Fig. 1, while steps 5 and 6 constitute the "symbol estimation".

4.2 Decoding

Codewords obtained from symbol processing are then passed to a decoder program. If the data is encoded with RS codes, the soft values obtained from Step 6 of symbol processing are quantized into two level by straight-forward thresholding (e.g., '0' for soft values greater than 0, '1' otherwise). Afterwards, hard-decision decoding for RS codes [14] is performed on the discrete values.

If the data is encoded with LDPC codes, decoding is performed by using a belief-propagation algorithm [10]. In this algorithm, we consider representing the mathematical structure of the code with a bipartite graph whose incident matrix coincides with the check matrix of the code. The nodes of the bipartite graph are grouped to *variable nodes* and *check nodes*. A variable node receives information (the soft value) from neighbor check nodes, and it attempts to estimate which symbol ('0' or '1' bit) has been transmitted. During the estimation, the statistical information of the communication channel, such as the variance of the Gaussian channel, is considered to derive various probabilities. A check node receives the estimated symbols from neighbor variable nodes, monitors parity constraints, and gives check nodes suggestions for the transmitted symbol. The accuracy of the estimation improves as nodes exchange messages iteratively. Refer to literature [10] for the detailed description of the belief-propagation decoding algorithm.

5. Evaluation and Results

The following experiments were designed to determine the effects of data density on the channel, and evaluate the performance of the error control mechanisms in the symbology. All experiments used the same office equipment and symbol processing steps for testing. In both the printing and scanning process, monochrome color settings were used. The general steps for experiments are as follows:

(1) First, input data and parameters were passed to the ECC encoder followed by the image construction process, both written in the C programming language. The program then generated a sample set of symbols.

(2) The sample set was printed on a sheet of plain white bond paper using a Canon LBP3410 laser printer with the default settings.

(3) The sheet of paper was scanned using an Epson GT-F720 flatbed scanner at 720 dpi with monochrome settings.

(4) Finally, the scanned image was passed to the image reconstruction process and subsequent modules, which are also written in C.

In the image reconstruction, we used conventional algorithms for image processing. Refinement of the image processing algorithms can improve the results, but this is not investigated in this study currently.

Table 2 Input parameters used for Test 1.

Parameter	Value
Dimension *dim*	57, 67, 77, 87, 97, 107, 117
Margin factor mf	0.6, 0.7, 0.8, 0.9, 1.0
Encoding	None
Interleave level *lvl*	1

5.1 Test 1: Effect of Data Density on Channel

As noted earlier, the accuracy of estimating cell values is dependent on the data density of the barcode. To test this, 35 sets of symbols with different symbology parameters were processed. Each sample set consisted of five barcode symbols with random data cell values. The set was then printed with a combination of dimension and margin factor values listed in **Table 2**. After symbol processing, the PSNR of each combination of symbology parameters was graphed and analyzed to determine the overall accuracy of cell value estimations.

The PSNR results of the test are presented in **Fig. 6**. PSNR values for $mf = 0.8$, 0.9 and 1.0 showed a decreasing trend as data density increased. This is in line with the fact that as data cells are printed closer to each other, inter-pixel leakage of black cells affect neighboring white cells and leads to reduced channel quality.

For data cells with lower data densities and $mf = 0.6$ or 0.7, the printing size of black cells were small compared to the actual cell size (e.g., 0.268 mm^2 and 0.446 mm^2, respectively for $dim = 57$, $mf = 0.6$). Hence, the effects of inter-pixel leakage were not enough to compensate for the remaining space in the data cell. This led to lower soft values for black cells and lower PSNR values overall. The PSNR values improved when data density was increased, since sampling cells were more densely packed. In this case, the soft values stabilized and became proportional relative to the expected values. PSNR values for densities beyond 370.4 showed a decreasing trend, due to the increased leakage of black cells on neighboring white cells.

5.2 Test 2: Performance of RS Codes and LDPC Codes

The second experiment evaluates the error correction performance of RS and LDPC codes in the symbology. To measure error correction performance, the BER of each sample set was analyzed, where

$$BER = \frac{\text{Number of erroneous bits after decoding}}{\text{Total number of bits of input data}} \qquad (5)$$

BER results for each sample set were then gathered and analyzed. Note that RS and LDPC decoder programs were modified to return the final state of the data word, whether or not the the received codeword was decoded correctly. This modification allows for an accurate BER calculation.

To conduct the experiment, 10 sets of symbols were generated using parameters listed in **Table 3**. Each set contained 24 barcode symbols with data obtained from an input file. The RS encoder used a $(255, 211)$ code, while the LDPC encoder used a $(6, 5)$ code. In this test, symbols were not subjected to physical damage in order to assess performance against errors caused by device-oriented factors only.

The preliminary investigation also shows that symbol processing fails for parameter choices above $dim = 127$ and $mf = 0.6$,

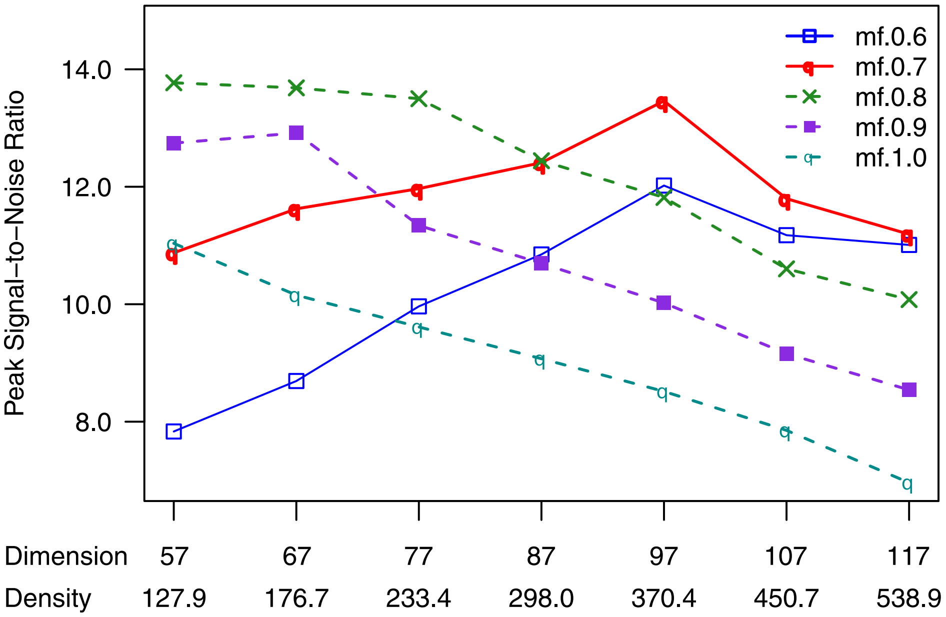


Fig. 6 Peak signal-to-noise ratio of symbols using parameters listed in Table 2.

Table 3 Input parameters used for Test 2.

Parameter	Value
Dimension *dim*	97, 107
Margin factor *mf*	0.6, 0.7, 0.8, 0.9, 1.0
Encoding	RS, LDPC
Interleave level *lvl*	1

Table 4 BER analysis of symbols using parameters listed in Table 3.

mf	*dim* = 97		*dim* = 107	
	RS	LDPC	RS	LDPC
0.6	0.013%	0.000%	0.094%	0.000%
0.7	0.017%	0.000%	0.069%	0.084%
0.8	0.750%	0.000%	2.447%	1.792%
0.9	4.086%	1.513%	5.021%	4.658%
1.0	6.280%	6.916%	10.549%	11.188%

Table 5 Input parameters used for Test 3.

Parameter	Value
Dimension *dim*	97
Margin factor *mf*	0.6, 0.7, 0.8, 0.9, 1.0
Encoding	LDPC
Interleave level *lvl*	1, 3, 5

and this seems to be the performance limit of the considered symbology. At these limit parameters, the noise of our channel is well approximated as the AWGN channel with the variance value $\sigma^2 = 0.05637$. The variance can become slightly smaller for smaller choices of dimension values, because the channel disturbance is mild if the dimension is small. It is ideal if we could estimate the variance from the scanned image in an adaptive manner, though, that kind of channel estimation is another challenging task. To get around this issue, we used the constant variance $\sigma^2 = 0.05637$ for all the evaluations in Test 2 and 3, because the barcode image can be subjected to external damages as investigated in Test 3. Thus, assuming a worse channel model (with higher variance) than expected seems to be a reasonable direction to mitigate unpredictable external factors.

Table 4 shows the results of BER analysis. Overall, the BER of symbols for *dim* = 97 were lower than for *dim* = 107. This was expected, since the PSNR values for *dim* = 107 were lower. Furthermore, the BER values exhibited the same trend as PSNR values with respect to the margin factor. This can be attributed to the decreased channel quality when margin factors are increased, as observed in Test 1.

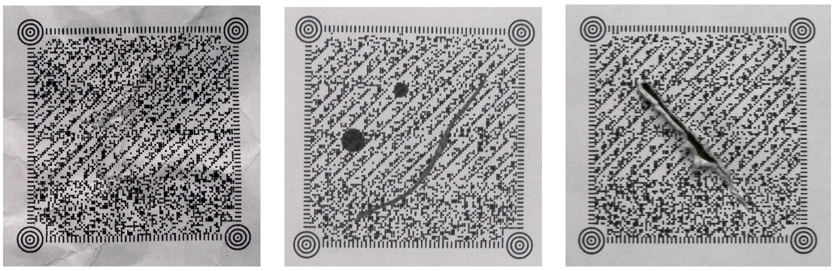
Even though values in Table 4 are not very accurate due to limited number of samples, LDPC codes generally yielded lower error rates than RS codes in both dimensions. This is because the accuracy of the estimated data cell value deteriorated when thresholded from a soft value into a discrete value; thus RS codes were not able to perform as well as LDPC codes. LDPC codes reported low BER (less than 0.05%) when PSNR values were greater than 11.176 dB. Both coding schemes performed poorly (BER > 5.0%) when symbols had a PSNR value less than 9.158 dB.

5.3 Test 3: Performance of Interleaving

To evaluate the symbology's robustness against burst errors, 15 sets of symbols with different symbology parameters listed in **Table 5** were printed. Interleave levels were varied to *lvl* = 1 (no interleaving), *lvl* = 3 and *lvl* = 5, while the dimension parameter was kept constant. Each symbol in a set was then subjected to one of three types of physical damage which normally occur to a sheet of paper in home and office environments, as shown in **Fig. 7**. After symbol processing, the BER results were obtained and conclusions were derived from the results.

The BER analysis for damaged symbols are shown in **Table 6**. Overall, the number of errors detected in this test were higher than in Test 2 because of the introduction of burst errors. Also, the trend of BER for all interleave levels performed similar to the results in Test 2, where increasing the margin factor (hence, decreasing PSNR) had a negative effect on the performance.

For comparison, consider symbols which did not use interleaving (i.e., *lvl* = 1) as a benchmark of performance. Based on this, the BER results improved when the interleave level was in-



(a) Folding/Crumpling (b) Markings (c) Tearing

Fig. 7 Types of damage to a symbol.

Table 6 BER analysis of damaged symbols using parameters listed in Table 5 with varying interleave levels.

mf	Interleave level lvl		
	1	3	5
0.6	1.070%	0.344%	3.736%
0.7	3.238%	1.673%	1.681%
0.8	4.382%	3.397%	4.093%
0.9	8.298%	6.147%	8.128%
1.0	11.845%	11.988%	14.594%

creased to $lvl = 3$. In these symbols, data was stored in a non-contiguous way and therefore the burst errors due to damage was decreased. Intuitively, the performance should further increase when the interleave level is also increased, but the BER results for $lvl = 5$ show that this is not the case. This is because the distance between cells was decreased from $dist(97, 3) = 32$ to $dist(97, 5) = 19$ (Eq. (4)). If the distance between two sequential data cells is lower, there is a higher chance that physical damage affects more sequential data cells.

Finally, note that this test focuses on the robustness of the high-density channel, hence physical damage was inflicted on the data area only and not to the finder pattern and timing patterns of the symbol. This is due to the fact that the current version of the symbology implements error correction mechanisms for the data cells only. Therefore, finder and timing patterns are sensitive to damage.

6. Conclusion and Future Work

In this paper, we investigated techniques for error control in a high-density 2D monochrome barcode. A high-density symbology was defined, and its communication channel was modelled. The performance of error correction mechanisms were then tested with different symbology parameters.

The study showed that some characteristics of laser printing technology introduce inaccuracies to high-density barcodes. The communication channel of the symbology is an AWGN channel, where the channel quality of a symbol can be evaluated based on its PSNR value. Some symbology elements such as margin factor and timing pattern design are effective for controlling errors. Also, LDPC codes show a better error correction performance compared to Reed-Solomon codes. Finally, the interleaving strategy of the symbology is a useful technique to mitigate burst errors caused by physical damage.

Future work for this study includes improving of the robustness of finder patterns and timing patterns, and the dynamic computation of AWGN variance for LDPC decoding; that is, the variance of a symbol is computed upon scanning the symbol - thereby taking into account the physical condition of the symbol - to improve decoding performance.

References

[1] Andreadou, N., Assimakopoulos, C. and Pavlidou, F.N.: Performance Evaluation of LDPC Codes on PLC Channel Compared to Other Coding Schemes, *IEEE International Symposium on Power Line Communications and Its Applications, ISPLC '07*, pp.296–301 (2007).

[2] Chen, J., Wang, L. and Li, Y.: Performance comparison between non-binary LDPC codes and Reed-Solomon codes over noise bursts channels, *Proc. 2005 International Conference on Communications, Circuits and Systems*, Vol.1, pp.1–4 (2005).

[3] Fu, J., Farn, C. and Chao, W.: Acceptance of electronic tax filing: A study of taxpayer intentions, *Information & Management*, Vol.43, No.1, pp.109–126 (2006).

[4] Gallager, R.: Low-density parity-check codes, *IEEE Trans. Inf. Theory*, Vol.8, No.1, pp.21–28 (1962).

[5] Gao, J., Kulkarni, V., Ranavat, H., Chang, L. and Mei, H.: A 2D Barcode-Based Mobile Payment System, *Proc. 2009 Third International Conference on Multimedia and Ubiquitous Engineering*, *MUE '09*, Washington, DC, USA, pp.320–329, IEEE Computer Society (2009).

[6] Jancke, G.: High Capacity Color Barcodes, Microsoft Research (online), available from ⟨http://research.microsoft.com/en-us/projects/hccb/⟩ (accessed 2011-11-27).

[7] Japanese Standards Association: JIS X 0500-2:2009 — Information technology — Automatic Identification and Capture *AIDC* Techniques — Harmonized Vocabulary — Part 2 Optically Readable Media *ORM* (2009).

[8] Lin, S. and Costello, D.J.: *Error Control Coding: Fundamentals and Applications*, Prentice Hall, Upper Saddle River, 2nd edition (2003).

[9] Luby, M.G., Mitzenmacher, M., Shokrollahi, M.A. and Spielman, D.A.: Improved low-density parity-check codes using irregular graphs, *IEEE Trans. Inf. Theory*, Vol.47, No.2, pp.585–598 (2001).

[10] MacKay, D.J.C.: Good Error Correcting Codes based on Very Sparse Matrices, *IEEE Trans. Inf. Theory*, Vol.45, No.2, pp.399–431 (1999).

[11] Reed, I. and Solomon, G.: Polynomial codes over certain finite fields, *Journal of the Society for Industrial and Applied Mathematics*, Vol.8, No.2, pp.300–304 (1960).

[12] Sonka, M., Hlavac, V. and Boyle, R.: *Image Processing, Analysis, and Machine Vision*, PWS Publishing, Pacific Grove, 2nd edition (1999).

[13] Tan, K., Chai, D. and Kato, H.: *Barcodes for Mobile Devices*, Cambridge University Press, New York, 1st edition (2010).

[14] Wicker, S. and Bhargava, V.: *Reed-Solomon codes and their applications*, Wiley-IEEE Press (1999).

[15] Yeh, Y.L., You, J.C. and Jong, G.J.: The 2D Bar-Code Technology Applications in Medical Information Management, *International Conference on Intelligent Systems Design and Applications*, Vol.3, pp.484–487 (2008).

Ramon Francisco Mejia received his B.S. and M.S. degrees in Computer Science from the Ateneo de Manila University in 2006 and 2008, respectively. He is currently a graduate student at Nara Institute of Science and Technology. His current research interests include role-based access control and error-correcting codes.

Yuichi Kaji is an Associate Professor at the Laboratory of Applied Algorithmics, Graduate School of Information Science, Nara Institute of Science and Technology. He received his Ph.D. degree in Information and Computer Sciences from Osaka University in 1994. He then joined the faculty of Nara Institute of Science and Technology in the same year. He was also a visiting researcher at the University of California, Davis in 2003, and the University of Hawaii at Manoa in 2004. His current research interests include theory of error-correcting codes, fundamental techniques for information security, and the theory of automata and rewriting systems.

Hiroyuki Seki is a Professor at the Laboratory of Applied Algorithmics, Graduate School of Information Science, Nara Institute of Science and Technology. He received his Ph.D. degree in Information and Computer Sciences from Osaka University in 1987. He became a faculty member of Osaka University as an Assistant Professor, and later, an Associate Professor from 1987 to 1994. In 1994, he joined the faculty of Nara Institute of Science and Technology, where he has been a Professor since 1996. His current research interests include formal language theory and formal approach to software development.

(Editor in Charge: *Takuya Kida*)