Analysis of a visually significant bar code system based on circular coding*

Yufang Sun[1], Robert Ulichney[2], Matthew Gaubatz[3], Stephen Pollard[4], Steven Simske[5], and Jan P. Allebach[1];
[1] **Electronic Imaging Systems Laboratory, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, U.S.A. {sun361,allebach}@purdue.edu;**
[2] **Hewlett-Packard Laboratories, Stow, MA 01775, U.S.A. u@hp.com**
[3] **Hewlett-Packard Laboratories, Seattle, WA 98136, U.S.A. matthew.gaubatz@hp.com**
[4] **Hewlett-Packard Laboratories, Bristol, UK, stephen.pollard@hp.com**
[5] **Hewlett-Packard Laboratories, Ft. Collins, CO 80528, U.S.A. steve.simske@hp.com**

Abstract

Circular Coding [19] is a 2D coding method that allows data recovery with merely a cropped portion (size $W \times H$) of the code and no carrier image knowledge involved. The B-bit payload is repeated from row-to-row, with each row being circularly shifted by D positions relative to the previous row. Every V rows, a phase row is interleaved between the payload rows. It is also shifted in the same manner as the payload rows. The encoded data array is embedded in a halftone image by shifting the dot-clusters within the halftone cells. The resulting image can be printed, and then captured with a mobile phone camera. The encoded data array is extracted from the captured halftone image by detecting the shifts in the dot-clusters.

In this paper, we introduce the encoding and decoding system and investigate the performance of the method for noisy and distorted images. For a given required decoding rate, we modele the transmission error and compute the minimum requirement for the number of bit repeats. Also, we develope a closed form solution to find the the corresponding cropped-window size that will be used for the encoding and decoding system design.

Introduction

Information embedding techniques in hardcopy prints are useful in many applications. One category of the data embedding techniques embed data in a region that is solely dedicate to contain the message, but the visual appearance of the coded image may be unsatisfactory. 1D and 2D barcodes [**?**] and DataGlyphs [10] [9] are the predominant techniques in this class. In the other category, information is embedded while the original image is retained. Many methods of hiding information in printing [1–3, 8, 12, 15, 16, 19–22] have been proposed. Among them, Bulan [3] used a orientation modulation for data hiding in clustered-dot halftone prints, where the message is represented by the different orientations of the clustered-dot halftoning.

Once the image is encoded, printed, and captured by some device, there are different errors that might be present, such as the local distortion in printing, rotation and distortion in capturing the image. These errors and data erosion requires a robust channel coding method to ensure the decoding success. Ulichney [20] proposed a circular coding with an interleaving method for channel encoding. Circular Coding [19] is a 2D coding method that allows data recovery with merely a cropped portion (size $W \times H$) of the code, and no carrier image knowledge involved. The B-bit payload is repeated from row-to-row, with each row being circularly shifted by D positions relatively to the previous row. There is a phase row interleaved between the payload rows in every V rows. It is also shifted similarly as the payload rows. The encoded data array is embedded in a halftone image by shifting the dot-clusters within the halftone cells. The resulting image can be printed and captured with a cell phone camera. The encoded data array is extracted from the captured halftone image by detecting the shifts in the dot-clusters. To decode the payload from data array given B, D, and V, the relative location of the phase rows is determined by finding the best fit to the assumed data structure. Then the phase rows are decoded to get the payload starting position in each line of the data array that contains payload information.

In this paper we will analyze the performance of the encoding scheme and use a statistical model to predict the decoding performance with simulated data transmission errors and erosion.

Data Embedding Framework

The data embedding framework is shown in Fig. 1. The message $\mathbf{m}$ is embedded into the continuous-tone (carrying) image $I[m,n]$, which is halftoned and used to embed message $\mathbf{m}$ into the cluster dot/hole. The data embedded halftone image is denoted as $I^h[m,n]$, which is printed and captured by some device, such as a scanner or a camera. The decoded message $\hat{\mathbf{m}}$ is then decoded from the captured image, which is denoted as $I^c[m,n]$. To overcome the data erosion and error in the printing-capture channel, we used the channel encoder with data redundancy.

Channel Encoder With Circular Coding

The goal of encoding is to represent a payload using a 2D binary symbol. The data carrying unit is the image cluster of 4×4 pixels. There are 3 steps for image decoding:

Halftoning

Digital image halftoning quantizes a gray scale image to 1 bit/pixel. It could be classified to amplitude modulation (AM), frequency modulation (FM), or AM-FM hybrid. Block-error diffusion [7] is the method of producing FM halftones for printing and display. Halftone images are typically binary. Each pixel of halftone image is either on or off, indicating whether link/toner is deposited on this pixel or not. There are various methods that can

*Research sponsored by Hewlett-Packard Laboratories, U.S.A.

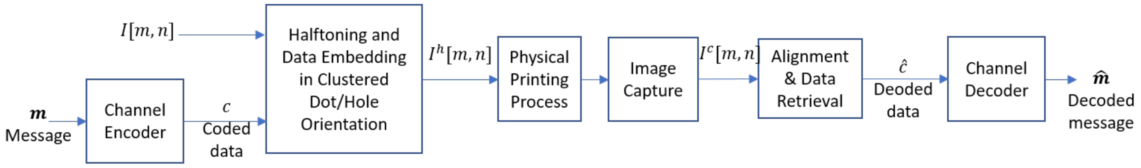

Figure 1: *Halftone data embedding framework.*

be used to embed data into halftone images [4–6, 11, 13, 17] .

The gray scale image is block-thresholded with a screening array $T(k,l)$:

$$T(k,l) = \frac{1}{64} \cdot \begin{bmatrix} 50 & 52 & 48 & 44 & 15 & 13 & 17 & 21 \\ 54 & 64 & 62 & 46 & 11 & 1 & 3 & 19 \\ 56 & 58 & 60 & 42 & 9 & 7 & 5 & 23 \\ 36 & 38 & 40 & 34 & 31 & 27 & 25 & 29 \\ 16 & 14 & 18 & 22 & 49 & 51 & 47 & 43 \\ 12 & 2 & 4 & 20 & 53 & 63 & 61 & 45 \\ 10 & 8 & 6 & 24 & 55 & 57 & 59 & 41 \\ 32 & 28 & 26 & 30 & 35 & 37 & 39 & 33 \end{bmatrix} \qquad (1)$$

Let $I(m,n)$ denote the gray scale image, then the halftone image $I^h(m,n)$ is obtained using Eq. (2).

$$I^h(m,n) = \begin{cases} 1, & \text{if } I(m,n) \geq T(m,n) \\ 0, & \text{if } I(m,n) < T(m,n) \end{cases} \qquad (2)$$

The halftone cell is 8 × 8, and each halftone cell contains 4 sub-cells, each of which is 4 × 4. For a constant gray scale image at level $0 \sim 1$, there are only 33 halftone patterns as shown in Fig. 2. To make the halftone patterns limited within these patterns, we will first average the gray scale value with each 4 × 4 sub-cell.

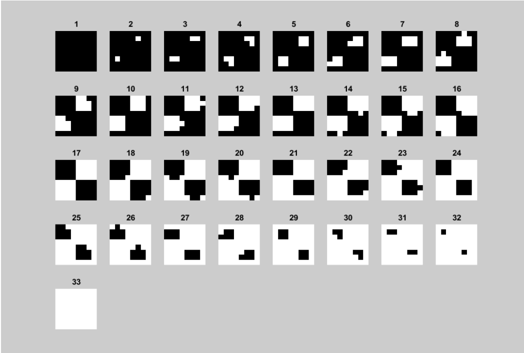

Figure 2: *Halftone images using the screening array $T(m,n)$ with diffrent gray scale value.*

For each of the sub-cells, if it is all black or write, we call it an "abstention" sub-cell; if it is white holes surrounded by black ones, or vice versa, we call these potential "carrying" sub-cell. Examples of the sub-cells are shown in Fig. 3.

Creating the data array using circular coding method

The payload contains B binary symbols. It is then repeated in the first row of the data array, until the end of the row. For each row below, every symbol is circularly shifted by a given bit value D. However, for every V-th row, the payload is replaced by a phase row, which has the same length as the payload, and which is used to represent some important information. As we circularly shift the payload and transfer each version of the payload to its decimal value, there will be some versions that have the smallest decimal value, which we define as the standard version of the payload P, denoted as S. Then the payload P is represented as the standard version S, and the circular shifting bits C. C is encoded in the phase line with some method. We select the payload with the unique standard version S, to avoid confusion in further decoding.

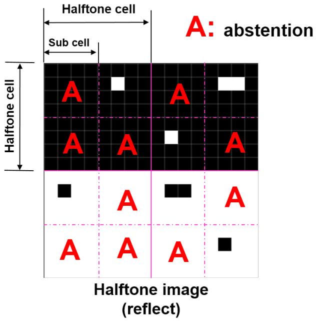

Figure 3: *Example of halftone cell and sub-cell. Halftone cell has a size of 8 × 8, and sub-cell has a size of* 4 × 4.

Embedding the data array into the halftone image

The data array symbol is used to hide data into each of the halftone sub-cells, row by row, and sub-cell by sub-cell. If the sub-cell is an abstention, we can not embed any symbol in it, but it still takes one position in the data array.

The symbol can be embedded into the halftone cluster by changing the orientation of the cluster [3]. Or it can be embedded into the clustered-dot halftone image by shifting the clustered-dot within the sub-cell [18]. For example, let the un-shifted halftone sub-cell represent 0s, and shift the cluster-dot right and down one pixel within the halftone sub-cell to represent 1s. In order to have a homogeneous shift of the clustered-dots for the whole encoded halftone image, we use the 'alanced shift' rule. That is, for each gray level, we push the clustered-dot either to the north-west and south-east direction to represent 0s, and push the clustered-dot either to the north-east or south-west to represent 1s. We alternately select the direction for each time we need to encode a bit.

In Fig. 4, the gray-scale image is halftoned, then the cluster of the black dots or white holes within each halftone sub-cell is shifted to embed the data array.

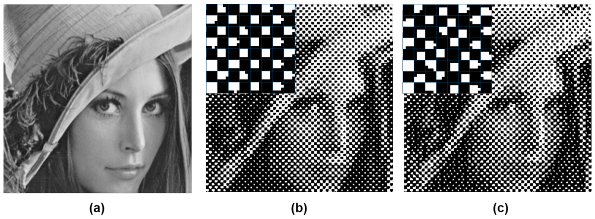

Figure 4: *Example of embedding a data array into a halftone image.* $\mathbf{P} = [11111111111111111], \mathbf{U} = [00000000000000000], B = 17, D = 2, V = 3$.*(a) gray-scale, (b) halftoned, (c) halftoned with embedded data array.*

Channel Decoder With Circular Coding

The payload length B, the row-to-row shift D, the interleaving phase period V, and a cropped portion of the data array are given to the decoder, while the phase row first appears in the cropped data array is unknown. The decoder then exhaustively searches the highest possible position of the first phase row start. To do this, we remove the assumed phase rows and examine the remaining payload rows. If there is no error in the data array transmission, then every bit will be repeated in its repeating position and will yield a perfect consistency. However, if the assumption ofthe position of the phase rows is incorrect, then the remaining payload rows will contain both payload rows and phase rows. For each bit and its repeating positions, it will contain the value of the payload and phase, which will have a lower consistency. With a certain bit error rate in transmission, it is usually the case that the pure payload set will have higher consistentcy than the mixture set. So we will try to find the set with highest consistency as the pure payload set.

Once we find where are the payload rows, the remaining phase rows are determined as well. By checking the major bit value of the repeating bit positions of the payload rows, we can find a shifted version of the payload P'. Similarly, by checking the major bit value of the repeating bit positions of the phase rows, we can find a shifted version U' of the phase. For every payload, as we discussed before, we guarantee that the standard version is unique, then there is a unique circular shift C that will shift from the standard version S to the original payload P. We will find out the standard version from P', and figure out the circular shift C' from P' to S. It will be the same C' that will shift the phase U' to U.

Analyze the Bit Position Shift Structure

Find the bit position index of shifted locations

In the circular coding algorithm, the bit position index is circularly shifted row by row with a given number D. Given the bit position index $\mathscr{P}(m,n)$ at location (m,n) in the circularly shifted data array, we would like to know the shifted bit position index $\mathscr{P}(m+\Delta m, n+\Delta n)$ at a shifted location $(m+\Delta m, n+\Delta n)$.

Row shift

First, only the row shift is considered. If there are Δm rows shift down, and the row-to-row offset is D, then the total bit shift from location $\mathscr{P}(m,n)$ to $(m+\Delta m, n)$ can be denoted as $f(\Delta m, 0)$.

$$f(\Delta m, 0) = \Delta m \cdot D \tag{3}$$

If we set the bit position at original position (m,n) to be 0, then the bit position at the shifted position $(m+\Delta m, n+\Delta n)$ is a function of Δm, the row-to-row shift D, and the payload bit length B.

$$\mathscr{P}(m+\Delta m, n) = \mathrm{mod}(f(\Delta m, 0)), B) = \mathrm{mod}(\Delta m \cdot D, B) \tag{4}$$

Now let us consider the bit position at original position (m,n) as an arbitrary number from 0 to $B-1$, then the bit position at the shifted position $(m+\Delta m, n)$ is a function of Δm, the row-to-row shift D, payload bit length B, and the bit position at the original location (m,n) can be expressed as

$$\mathscr{P}(m+\Delta m, n) = \mathscr{P}(m,n) - \Delta m \cdot D + \mathscr{Q}, \tag{5}$$

where $\mathscr{Q} = \left\lceil \frac{\Delta m \cdot B - \mathscr{P}(m,n)}{B} \right\rceil \cdot B$

Column shift

Then, the column shift is considered. Similarly, let us consider the bit position index at original position (m,n) as an arbitrary number from 0 to $B-1$, then the bit position at the shifted position $(m, n+\Delta n)$ is a function of Δn, the row-to-row shift D, the payload bit length B, and the bit position at the original location (m,n). We found that

$$\mathscr{P}(m, n+\Delta n) = \mathrm{mod}(\mathscr{P}(m,n) + \Delta n, B) \tag{6}$$

Combine row shift and column shift

Now combining the column shift with row shift, we get

$$\begin{aligned} &\mathscr{P}(m+\Delta m, n+\Delta n) \\ &= \mathrm{mod}(\mathscr{P}(m+\Delta m, n) + \Delta n, B) \\ &= \mathrm{mod}\left(\mathscr{P}(m,n) - \Delta m \cdot D + \left\lceil \frac{\Delta m \cdot B - \mathscr{P}(m,n)}{B} \right\rceil \cdot B, B\right) \end{aligned} \tag{7}$$

This equation shows that for given B, D and two locations (m_1, n_1) and (m_2, n_2), if their bit position indices are the same, i.e. $\mathscr{P}(m_1, n_1) = \mathscr{P}(m_2, n_2)$, then for any shift $(\Delta m, \Delta n)$ for these two original locations, respectively, the shifted bit position indices will also be the same; and vise versa.

This relationship can be written as

$$\begin{gathered} \mathscr{P}(m_1, n_1) = \mathscr{P}(m_2, n_2) \\ \Leftrightarrow \mathscr{P}(m_1+\Delta m, n_1+\Delta n) = \mathscr{P}(m_2+\Delta m, n_2+\Delta n) \end{gathered} \tag{8}$$

Canonical Crop Window Location Set (CCWLS)

From Eq. (8) we can see that for a given crop window of data cropped from a data array (with payload length B, row-to-row shift D, and interleaving phase period V), the entire set of bit position indices of this cropped data is determined by the starting bit position index at the upper left corner of the crop window.

In order to evaluate every possible cropped data arrangement, we need to consider every unique starting bit position index of the crop window. There exists a minimum size rectangular region of data that includes every unique starting position index for the crop window, which is called the Canonical Crop Window Location Set (CCWLS).

How to find the CCWLS?

First, assuming there is no impact from the interleaving phase period V. We have the parameters of payload length B and row-to-row shift D. To cover every possible starting index, we note that a rectangular region with size $1 \times B$ covers every possible starting position index from 0 to $B-1$. So we have the CCWLS rectangular region size of $1 \times B$.

Second, with the consideration of the interleaving phase period V, there are parameters B, V, and D. In this case, the crop window may consist of a combination of payload rows and phase rows. These two sets of rows comprise difference data sets. So every possible cropped data arrangement in these pairs of sets can be evaluated.

In a rectangular region of data with starting row index m', the first phase row may be any one of the V rows with index $0 \leq m' \leq V-1$. In any phase row, the phase code repeats with period B. Thus, the CCWLS only need to contain a length B segment of one phase row. This means that there are V different pairs of payload and phase data sets. So we need to expand the CCWLS rectangular region to size $V \times B$.

Note that during the halftone process, the screening array $T(m,n)$ that we used will produce the clusters where around half of them will be all black or all white, without carrying any useful information. So in a local region, we can approximate this texture using a sub sampling mask $\mathscr{M}(m,n)$. It is a checkerboard pattern defined by Eq. (9).

$$\mathscr{M}(m,n) = \text{mod}(m+n, 2) \tag{9}$$

To avoid the case that all the repeating positions will fall into the abstention region, in our system design, we always choose an odd number for the length of the payload. So the sub-sampling mask value at one position in any given row and the sub-sampling mask value at the position B bits later in that row will be different. Therefore, any consecutive set of $2B$ rows will include each of the B unique bit position indices in an unmasked position.

The same argument applies when we consider the interleaving phase period V. So we conclude that the CCWLS rectangular region has size $V \times 2B$.

An example of the CCWLS

Let the payload bit length $B = 7$, the bit index $j = 0, 1, \ldots, 6$. The row-to-row shift $D = 1$, and the interleaving phase period $V = 3$. Let index numbers written in red indicate a phase, and index numbers written in black indicate a payload. Note that half of the data is been masked to simulate the halftone process that will cause half of the sub-cells to be abstentions.

In the example showb in Fig. 5, we can see that for a randomly selected crop window $\mathbf{W}_1$, we will be able to find the related crop window $\mathbf{W}_1^*$ that starts within the CCWLS, and contains the same bit indies for the entire crop window.

Requirement of Input Data for Recovery in a Noise Free Channel

In a noise free channel, the minimum requirement of data recovery is that every bit has at least one repeat. So we will find a general formula to calculate the bit repeat count in a given crop window size of the data array.

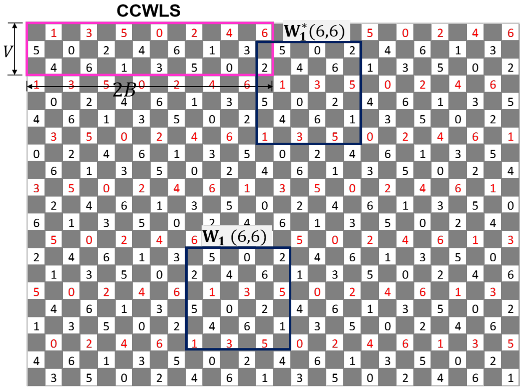

Figure 5: *A example canonical crop window location set (CCWSL). payload bit length $B = 7$, bit index $j = 0, 1, \ldots, 6$. The row to row shift $D = 1$, and the interleaving phase period $V = 3$. The CCWLS has size of $V \times 2B$. Any crop window $\mathbf{W}$ has a related crop window $\mathbf{W}^*$ that contains exactly the same bit repeating positions, but starting within the CCWLS.*

Let us denote by $\mathbf{W}$ a crop window with row index $0 \leq h \leq H-1$ and column index $0 \leq w \leq W-1$. Without loss of generality, we assume that the bit position index $j = 0$ at the starting position in $\mathbf{W}$, where $(h, w) = (0, 0)$.

The calculation of the bit repeat count contains the following steps:

1. Find out the column index $q(h, j)$ in row h of the crop window $\mathbf{W}$, where bit position index j first appears. Note that $q(h, j)$ will be inside the crop window, if $q(h, j) \leq W-1$ and $h \leq H-1$; otherwise, $q(h, j)$ will be outside of $\mathbf{W}$.
   By observing how the bit position index repeats with period B, and is shifted by D from row to row, we find that
   $$q(h, j) = \text{mod}(j + hD, B) \tag{10}$$

2. For each row in the crop window, find the number of positions starting from the column index $q_h(j)$ to the end of row h for bit position index j. Here $h < H$ since only rows within the crop window $\mathbf{W}$ are considered. This can be split into two cases, either $q(h, j) \leq W-1$ (inside of $\mathbf{W}$), or $q(h, j) \geq W$ (outside of $\mathbf{W}$), and can be formulated as
   $$N_{n,j} = \begin{cases} W - q(h, j), & \text{if } q(h, j) < W \\ 0, & \text{if } q(h, j) \geq W \end{cases} \tag{11}$$
   The above two cases can be combined in the Eq. (12)
   $$N_{h,j} = W - \min(q(h, j), W) \tag{12}$$

3. Find the number of times that the bit position index j appears in row h, denoted as $\mathscr{B}(h, j; B, D, W)$, in terms of $N_{h,j}$. This simple relationship can be written as
   $$\mathscr{B}_{\text{row}}(h, j; B, D, W) = \left\lceil \tilde{\mathscr{B}}_{\text{row}}(h, j; B, D, W) \right\rceil \tag{13}$$
   where
   $$\tilde{\mathscr{B}}_{\text{row}}(h, j; B, D, W) = \frac{N_{h,j}}{B} \tag{14}$$

4. To obtain the bit position repeat count for all the rows within the crop window, we sum over the rows.

$$\mathscr{B}(j;B,D,W) = \sum_{h=0}^{H-1} \mathscr{B}_{\text{row}}(h,j,q;B,D,W) \qquad (15)$$

5. Let us consider the sub-sampling mask $\mathscr{M}[m,n]$ defined in Eq. (9). If we shift the position from (m,n) a number of bits Δm and Δn, respectively, then the new mask value at the shifted location can be shown to be

$$\mathscr{M}[m+\Delta m, n+\Delta n] = \begin{cases} \mathscr{M}(m,n), & \text{if } \operatorname{mod}(\Delta m+\Delta n,2)=0 \\ 1-\mathscr{M}(m,n), & \text{if } \operatorname{mod}(\Delta m+\Delta n,2)=1 \end{cases} \qquad (16)$$

Let $k \in \{0,1\}$ be the mask value at the upper left corner of the crop window, then, to count the bit position repeat count, we first need to apply the mask to every bit occurrence within the crop window.

There could be more than one occurrence of bit position j in a given row of the crop window. Since B is always odd, we know that if bit position j is masked in the first positions of any row, then it will not be masked in its next occurrence in that row, if there is one. This pattern of alternating appearances of bit position j, either masked or unmasked, will repeat with period B until the end of the crop window row.

To account for this alternating pattern, we need to separate the occurrences of bit position j in each row, according to whether they occur in an even-numbered or an odd-numbered B-length period in that row. Here we assume that the B-length periods are numbered starting from zero.

In a given row, if the occurrences of bit position j occur in even-numbered B-length periods, the first occurrence of bit position j will be at bit location $q(h,j)$. On the other hand, if the occurrences of bit position j occur in odd-numbered B-length periods, the first occurrence of bit position j will be at bit location $q(h,j)+B$.

Then the total bit position occurrence count for bit position j is summed over all the rows in the crop window:

$$\mathscr{B}(j,k;B,D,W,H) = \sum_{h=0}^{H-1} \mathscr{B}_{\text{row}}(h,j,k;B,D,W,H) \quad (17)$$

where now the bit repeat count for each row can be represented as:

$$\mathscr{B}_{\text{row}}(h,j,q;B,D,W) = \left\lceil \frac{\tilde{\mathscr{B}}_{\text{row}}(h,j,q;B,D,W)\cdot g(k,h,q)}{2} \right\rceil + \left\lceil \frac{\tilde{\mathscr{B}}_{\text{row}}(h,j,q+B;B,D,W)\cdot g(k,h,q+B)}{2} \right\rceil \qquad (18)$$

where $\tilde{\mathscr{B}}_{\text{row}}(h,j,q;B,D,W) = \frac{W-\min(q(h,j),W)}{B}$, $g(k,h,q) = \operatorname{mod}(k+h+q,2)$, and $q(h,j) = \operatorname{mod}(j+hD,B)$.

6. We consider the interleaving phase with period V. In this case, the crop window $\mathbf{W}$, which has size $W \times H$, may consist of a combination of payload rows and phase rows.

We define the set of all the row indices in $\mathbf{W}$ as

$$\mathfrak{H} = \{k : 0 \le k \le K-1\} \qquad (19)$$

These rows will be divided into phase rows and payload rows. There will be V possible partitions of $\mathfrak{H}$ into the sets of payload row indices denoted as $\mathfrak{H}_{\text{pay}}$ and the set of phase row indices denoted as $\mathfrak{H}_{\text{pha}}$, where

$$\mathfrak{H} = \mathfrak{H}_{\text{pay}} \cup \mathfrak{H}_{\text{pha}} \qquad (20)$$

Let v denote the first index of the first phase row in the crop window $\mathbf{W}$. Then we can define the set of phase row indices as:

$$\mathfrak{H}_{\text{pha}} = \left\{ v+lV : l = 0,1,\cdots,\left\lfloor \frac{H}{V} \right\rfloor, v = 0,1,\cdots,V-1 \right\}$$
$$\mathfrak{H}_{\text{pay}} = \mathfrak{H} - \mathfrak{H}_{\text{pha}} \qquad (21)$$

To account for the role of the interleaving phase in our analysis of the bit repeating count, we separate the summation over the rows and columns in our previous expression for the bit repeating count according to this partition.

Note that the above illustrations are all based on the bit position index $j=0$. However, the formulas we derived are applicable to any bit position $j \in [0, B-1]$ within a single row h.

Result of bit repeat counts for given parameters

Using the formula developed in Eq. (17), and examining every possible crop window starting position in the CCWLS, we can calculate the bit repeat count for the payload and phase for every bit position. Fig. 6 shows the calculation result.

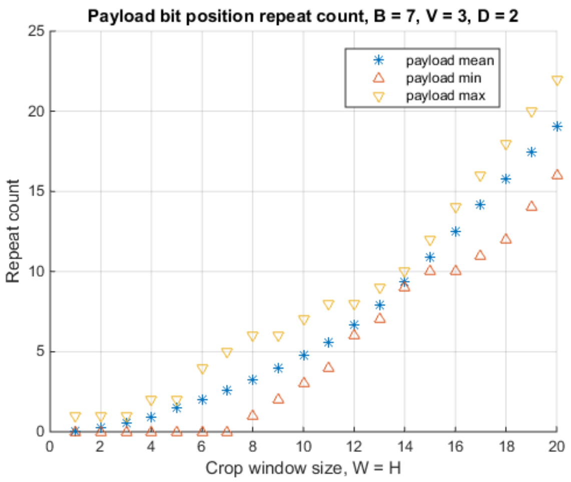

Figure 6: *Bit repeat count for payload. The min, max and mean values are determined based on the population of all bits that start within the CCWLS window.*

Requirement of Input Data for Recovery from a Noisy Channel Based on Statistic Modeling

To simplify the calculation, we assume that the crop window height H is an integer multiple of the interleaving phase period V. Thus, among those H rows of data in the crop window, there are H/V rows of phase, and $H \cdot (V-1)/V$ rows of payload. In addition, we assume that the number of columns W is also an integer multiple of the payload length B. So for each row in the crop window, there will be the same number of occurrences for each value of the bit position index j. Note that for a given bit index j, the number of occurrences of this bit position can be represented as $\mathscr{B}(j;H,W,B,D)$, without considering the difference between phase rows and payload rows.

The bit value for each repeating bit position, if it is belongs to a payload row, can be denoted as $P(j)$; and if this bit is in a phase row, then the bit value can be denoted as $U(j)$. We assume that the bit values of payload and phase are statistically independent. That is, the probability that the payload bit value is the same as phase bit value for a particular bit position index j is 0.5, which is also the probability that payload bit value is different from the phase bit value.

Model for the communication channel

There are different models for communication channels. One simple model is the memory-less Binary Symmetric Channel (BSC). The Binary Symmetric Channel has binary input and output, with a probability of transmission error p, i.e. the probability of switching values between 1 and 0. The probability of success of transmission of one bit is $P(S) = 1-p$, and the probability of failure of transmission of one bit is $P(F) = p$. We assume that the errors at each position are independent and identically distributed.

Prediction of the decoding rate for pure payload rows

For the subset of the cropped data that includes all the actual payload, we can calculate the probability of the output value Y_i, given the value of input X_i. The probability that the detected bit value is the same as the original bit value in the pure payload set is the probability that fewer than half of the bits have a transmission error. Thus, we can have up to $\mathscr{K}$ transmission errors, where

$$\mathscr{K} = \left\lceil \frac{\mathscr{B}(j)-1}{2} \right\rceil \tag{22}$$

The probability that the decoded bit value for bit repeat position j is the same as its original value is thus given by

$$\begin{aligned} P(Y_j=0|X_j=0) &= P(Y_j=1|X_j=1) \\ &= \sum_{k=0}^{\mathscr{K}} \begin{pmatrix} \mathscr{B}(j) \\ k \end{pmatrix} (p)^k (1-p)^{\mathscr{B}(j)-k} \end{aligned} \tag{23}$$

The probability that the decoded bit value for bit repeating position j is the different from as its original value is the probability that at least half of the bits changed their value.

$$\begin{aligned} P(Y_j=1|X_j=0) &= P(Y_j=0|X_j=1) \\ &= \sum_{k=\mathscr{K}+1}^{\mathscr{B}(j)} \begin{pmatrix} \mathscr{B}(j) \\ k \end{pmatrix} (p)^k (1-p)^{\mathscr{B}(j)-k} \end{aligned} \tag{24}$$

We also have that $P(Y_j=1|X_j=0) = 1-P(Y_j=0|X_j=0)$ and $P(Y_j=0|X_j=1) = 1-P(Y_j=1|X_j=1)$. Thus, the probability that the decoded bit value Y_j is the same as the original bit value, and can be calculated as follows:

$$\begin{aligned} P(Y_j=X_j) =& P(Y_j=0|X_j=0)P(X_j=0) \\ &+ P(Y_j=1|X_j=1)P(X_j=1) \end{aligned} \tag{25}$$

We assume that the bit value at each bit position index does not depend on the bit position index, and the symbols 0 and 1 are equally likely. Thus, we have $P(X_j=0)=0.5$, and $P(X_j=1)=0.5$, for all j. So the probability that every bit in the payload is correctly decoded can be expressed as:

$$P(Y_j=X_j) = \sum_{k=0}^{\mathscr{K}} \begin{pmatrix} \mathscr{B}(j) \\ k \end{pmatrix} (p)^k (1-p)^{\mathscr{B}(j)-k} \tag{26}$$

The probability that the entire payload is correctly decoded is the joint probability that every bit in the payload is correctly decoded. Recall that we model the transmission errors as being identically and independently distributed at each bit position. So the joint probability of successfully decoding the entire payload is just the product of the probabilities of successfully decoding each bit position, formulated as:

$$P(Y=X) = \prod_{j=0}^{B-1} P(Y_j=X_j) \tag{27}$$

Validation of the analytical results with experimental results

By calculating the bit decoding rate with different probability of transmission error p from 0 to 0.3, and the bit repeating count $\mathscr{B}(j)$ from 1 to 40, the result is plotted in Fig. 7.

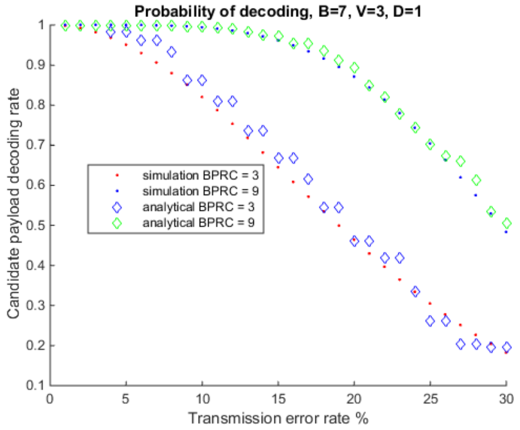

Figure 7: *Validation of the analytical results with simulation results. The simulation result is based on 1000 random noise sets.*

Conclusion and Future Work

In this paper, we validated that the experimental payload decoding rates are consistent with their theoretical results, given particular parameters and with various cropped-window sizes (Fig.

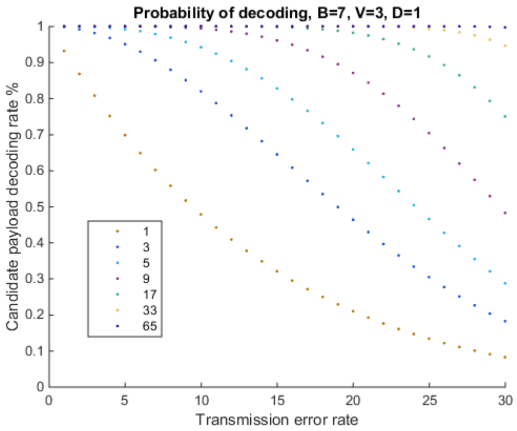

Figure 8: *Analytical decoding rate prediction as a function of different transmission error rate for different values of the minimum bit repeat count.*

7). Therefore, given required decoding rate and anticipated transmission error, we can compute the minimum requirement for the number of repeats (Fig. 8) or the corresponding cropped-window size (Fig. 6). The work has assumed that the phase information is known. In the future, we will repeat a similar simulation and analysis accounting for the impact of phase.

References

[1] Zachi Baharav and Doron Shaked. Watermarking of dither halftoned images. In *Security and Watermarking of Multimedia Contents*, volume 3657, pages 307–316, 1999.

[2] Jack T Brassil, Steven Low, Nicholas F. Maxemchuk, and Lawrence O'Gorman. Electronic marking and identification techniques to discourage document copying. *IEEE Journal on Selected Areas in Communications*, 13(8):1495–1504, 1995.

[3] Orhan Bulan, Gaurav Sharma, and Vishal Monga. Orientation modulation for data hiding in clustered-dot halftone prints. *IEEE Transactions on Image Processing*, 19(8):2070–2084, 2010.

[4] Pei-Ju Chiang, Jan P Allebach, and George T-C Chiu. Extrinsic signature embedding and detection in electrophotographic halftoned images through exposure modulation. *IEEE Transactions on Information Forensics and Security*, 6(3):946–959, 2011.

[5] Pei-Ju Chiang, Aravind K Mikkilineni, Edward J Delp, Jan P Allebach, and George T-C Chiu. Development of an electrophotographic laser intensity modulation model for extrinsic signature embedding. In *NIP & Digital Fabrication Conference*, volume 2007.

[6] Pei-Ju Chiang, Aravind K Mikkilineni, Edward J Delp, Jan P Allebach, and George T-C Chiu. Extrinsic signatures embedding and detection in electrophotographic halftone images through laser intensity modulation. In *NIP & Digital Fabrication Conference*, volume 2006, pages 432–435. Society for Imaging Science and Technology, 2006.

[7] Niranjan Damera-Venkata, Jonathan Yen, Vishal Monga, and Brian L Evans. Hardcopy image barcodes via block-error diffusion. *IEEE transactions on image processing*, 14(12):1977–1989, 2005.

[8] Ricardo L de Queiroz, Karen M Braun, and Robert P Loce. Detecting spatially varying gray component replacement with application in watermarking printed images. *Journal of Electronic Imaging*, 14(3):033016–033016, 2005.

[9] David L Hecht. Embedded data glyph technology for hardcopy digital documents. *SPIE-Color Hard Copy and Graphics Arts III*, 2171:341–352, 1994.

[10] David L Hecht. Printed embedded data graphical user interfaces. *Computer*, 34(3):47–55, 2001.

[11] Dhiraj Kacker, Tom Camis, and Jan P Allebach. Electrophotographic process embedded in direct binary search. *IEEE Transactions on Image Processing*, 11(3):243–257, 2002.

[12] Aravind K Mikkilineni, Gazi N Ali, Pei-Ju Chiang, George T-C Chiu, Jan P Allebach, and Edward J Delp. Signature-embedding in printed documents for security and forensic applications. In *Security, Steganography, and Watermarking of Multimedia Contents*, pages 455–466, 2004.

[13] Aravind K Mikkilineni, Pei-Ju Chiang, George T-C Chiu, Jan P Allebach, and Edward J Delp. Data hiding capacity and embedding techniques for printed text documents. In *NIP & Digital Fabrication Conference*, volume 2006, pages 444–447. Society for Imaging Science and Technology, 2006.

[14] Villan Sebastian, Renato Fisher, Svyatoslav Voloshynovskyy, Oleksiy Koval, and Thierry Pun. Multilevel 2d bar codes: towards high capacity storage modules for multimedia security and management. 1(4):405–420, 2005.

[15] Maria V Ortiz Segovia, George T-C Chiu, and Jan P Allebach. Using forms for information hiding and coding in electrophotographic documents. In *Information Forensics and Security, 2009. WIFS 2009. First IEEE International Workshop on*, pages 136–140. IEEE, 2009.

[16] Gaurav Sharma and Shen-ge Wang. Show-through watermarking of duplex printed documents. In *Security, Steganography, and Watermarking of Multimedia Contents*, pages 670–684. SPIE, Jan 2004.

[17] Sungjoo Suh, Jan P Allebach, George T-C Chiu, and Edward J Delp. Printer mechanism-level data hiding for halftone documents. In *NIP & Digital Fabrication Conference*, volume 2006, pages 436–440. Society for Imaging Science and Technology, 2006.

[18] Robert Ulichney, Matthew Gaubatz, and Steven Simske. Encoding information in clustered-dot halftones. In *NIP & Digital Fabrication Conference*, volume 2010, pages 602–605. Society for Imaging Science and Technology, 2010.

[19] Robert Ulichney, Matthew Gaubatz, and Steven Simske. Circular coding for data embedding. In *NIP & Digital Fabrication Conference*, volume 2013, pages 142–147. Society for Imaging Science and Technology, 2013.

[20] Robert Ulichney, Matthew Gaubatz, and Steven Simske. Circular coding with interleaving phase. In *Proceedings of the 2014 ACM symposium on Document engineering*, pages 21–24. ACM, 2014.

[21] Fuping Wang and Jan P Allebach. Printed image watermarking using direct binary search halftoning. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages

2727–2731. IEEE, 2016.

[22] Shen-ge Wang. Digital watermarking using phase-shifted stoclustic screens, June 26 2001. US Patent 6,252,971.

Author Biography

Yufang Sun received her BS in Electrical Engineering from the University of Jilin from China (2004). She is currently a PhD student, working as image processing and data analysis research assistant with Prof. Jan Allebach, in the School of Electrical and Computer Engineering at Purdue University. Her research interests are in image information embedding, decoding error analysis, etc. She has been working on the projects of circular coding and stegaframe detection, both sponsored by HP Labs.