Circular Coding with Interleaving Phase

Robert Ulichney
Hewlett-Packard Laboratories
165 Dascomb Road
Andover, MA 01810
+1-978-897-6223
u@hp.com

Matthew Gaubatz
Hewlett-Packard Laboratories
701 Pike Street, Suite 900
Seattle, WA 98101
+1-206-245-5766
matthew.gaubatz@hp.com

Steven Simske
Hewlett-Packard Laboratories
3404 E Harmony Rd.
Ft. Collins, CO 80528-9544
+1-970-898-1359
steve.simske@hp.com

ABSTRACT
A general two-dimensional coding method is presented that allows recovery of data based on only a cropped portion of the code, and without knowledge of the carrier image. A description of both an encoding and recovery system is provided. Our solution involves repeating a payload with a fixed number of bits, assigning one bit to every symbol in the image –whether that symbol is data carrying or non-data carrying– with the goal of guaranteeing recovery of all the bits in the payload. Because the technique is applied to images, for aesthetic reasons we do not use fiducials, and do not employ any end-of-payload symbols. The beginning of the payload is determined by a phase code that is interleaved between groups of payload rows. The recovery system finds the phase row by evaluating candidate rows, and ranks confidence based on the sample variance. The target application is data-bearing clustered-dot halftones, so special consideration is given to the resulting checkerboard subsampling. This particular application is examined via exhaustive simulations to quantify the likelihood of unrecoverable bits and bit redundancy as a function of offset, crop window size, and phase code spacing.

Categories and Subject Descriptors
I.7.5 **[Document and Text Processing]**: Graphics recognition and interpretation; E.4 **[Coding and Information Theory]**; H.3.2 **[Information Storage and Retrieval]**: Information Storage.

General Terms
Algorithms, Measurement, Performance, Experimentation, Security.

Keywords
Data-bearing Hardcopy, Coding, Halftoning, Data Recovery.

1. INTRODUCTION
Embedding data in hardcopy is increasingly important for linking paper and electronic workflows, security and other applications. Data-bearing hardcopy is most often accomplished with various types of multidimensional barcodes, along with more aesthetically pleasing alternatives of encoding symbols in halftones [1][2]. We propose a new solution for recovering data when only a part of the data-bearing image can be captured, with no knowledge of the carrier image. The solution is designed to be robust in the presence of one or more of the many print-scan process degradations [3] that can hurt data integrity.

DocEng '14, September 16 - 19 2014, Fort Collins, CO, USA

http://dx.doi.org/10.1145/2644866.2644888

In this paper, we describe a coding scheme that allows complete recovery of a payload from an arbitrarily cropped version of a 2D array of symbols representing 1-bit values. The following notation is used to describe the approach:

- B Number of bits in the payload.
- P Payload, representing a value between 0 and 2^B-1.
- S Payload in Standard form.
- C Circular shift to get the payload from the standard form.
- c Number of bits needed to represent C
- U Phase code representation of C.
- V Phase code row interleave period.
- D Row-to-row Offset of the payload in the image.
- P' Candidate recovered payload.
- U' Candidate recovered Phase code.
- D' Candidate row-to-row offset.
- W Width of the crop window.

The coding scheme employs a circularly shifted payload. Before describing the encoder and decoder we define what we call the "standard form" of such a payload.

1.1 Circular Shifting and Standard Form
As a simple one-dimensional example, consider a B=5 bit payload P=11001_2 repeated continuously:

… 10011100111001110011100111 …

An arbitrary crop of 5 bits from that stream will yield one of 5 circularly shifted versions of that payload: 11100_2, 11001_2, 10011_2, 00111_2, or 01110_2. To specify the correct payload from such a crop, we need one more piece of information: the circular shift to the right, C, (= 0,1,2,3, or 4) relative to some reference representation. We define that reference representation the "standard form", S, as the minimum value of all possible circularly shifted versions of the payload. In our example, S=00111_2. The circular shift to get the payload P=11001_2 from the standard form is C=2.

When the number of bits in the payload, B, can be segmented into parts of equal length it is possible that certain payloads can have more than one circular shift that maps the standard form back to the payload. Consider B=6 and P=101101_2, which repeats a series of 3 bits "101". The standard form is S=011011_2, and the payload can be resolved with a circular shift of both C=1 and C=4. Similarly P=101010_2 has standard form S=010101_2 with three equivalent circular shifts C=1,3, and 5.

In an earlier work, a 2D circular coding scheme [4] was presented that repeated the payload on each row but offset successive rows by an amount equal to the circular shift C, so that the payload could be recovered from the standard form S. Using that method, cases of equivalent circular shifts did not present an issue since all such shifts would correctly resolve the payload. The method herein demonstrates other advantages over the earlier method, but requires that the circular shift, C, be unique. For that reason we restrict the number of payload bits, B, to have no multiplicative factors, and so B must be prime. The reason for the need for a unique value of C will be made clear in the Recovery System section. If the desired payload size is not prime the next larger prime number can be used, with the extra bits used for error correction.

2. ENCODING SYSTEM

The goal of the encoder is to represent a payload, P, consisting of B bits in a two-dimensional array of one-bit symbols. Encoding is accomplished by repeating the payload across each line where each successive row is circularly shifted relative to the row above it by a fixed amount, D. The circular phase C is embedded in a "phase code", U, also of length B and interleaved between groups of payload rows. Every row of phase code, U, is followed by (V-1) rows of payload, P.

For a payload of B bits, the circular shift can have values $C \in \{0,1,2,\ldots(B-1)\}$. The number of bits needed to represent C is denoted as $c=\lceil \log_2 B \rceil$, where $\lceil . \rceil$ is the ceiling function. As c is less than B, there are a number of ways to construct the phase code U. One approach is constructing an error correcting code to fill the B bits, but there are other strategies that can be better suited for the carrier of this data.

While the coding scheme can be used for any two-dimensional array that could include magnetic or optical media, the target use is for printed halftone images where individual halftone clusters carry a bit of data as indicated by single-pixel shifts [4]. We use monochrome classical 45^{o} screen clustered-dot halftones. One way to think of such a halftone screen is as a set of clusters in complementary checkerboard arrangements of highlight and shadow cells. Figure 1 illustrates an enlargement of a portion of an image with a transition from shadow to highlight area. The shadow cell positions are denoted by red dots. Half of the cells are incapable of carrying data: shadow cells in highlight areas are completely white, and highlight cells in shadow areas are completely black. This checkerboard subsampling of the code adds a challenge for preserving the data.

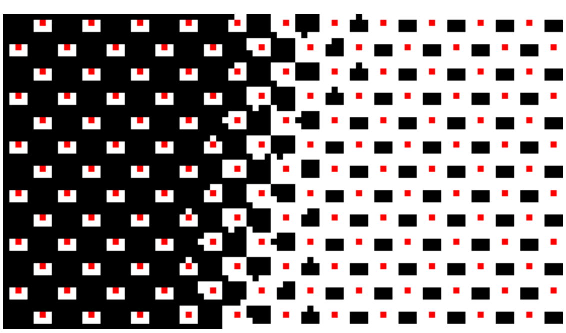

Figure 1. Zoomed halftone image with shadow cells indicated by red dots.

To protect the phase code, U, against the checkerboard decimation of halftone embedding, one approach is to duplicate every bit of C in succession, then repeat this "double" pattern until all B bits are filled. While there are many other possible representations, this is the method we use in this study. As an example, consider the case of B=17. The number of bits in C is c=5. The bit positions in the phase code U and C would be related as follows:

U bit position	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C bit position	4	4	3	3	2	2	1	1	0	0	4	4	3	3	2	2	1

Encoding the 2D array of symbols is carried out as follows:

(1) Set the payload bit length B, row-to-row offset D, phase code representation method, and interleave period V. B, D, and V must be known to the recovery system.

(2) Determine the standard form S of the payload P, and the circular phase C between S and P.

(3) Generate the Phase Code, U.

(4) Repeatedly fill (V-1) rows of S, followed by one row of U, with each row circularly shifted by D more bits than the row above.

Consider the example were B=17, D=2, and V=3. This code represents a very short payload but illustrates the approach. Figure 2 depicts the top left corner of a code generated by this method. Black numbers represent bit positions of the payload in standard form S and red numbers represent bit positions of the phase code U. Note that each row is circularly shifted by D=2 more bit positions than the row above it regardless of payload or phase code.

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	16	15	14	
1	0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	16	…
3	2	1	0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
5	4	3	2	1	0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	
7	6	5	4	3	2	1	0	16	15	14	13	12	11	10	9	8	7	6	5	
9	8	7	6	5	4	3	2	1	0	16	15	14	13	12	11	10	9	8	7	
11	10	9	8	7	6	5	4	3	2	1	0	16	15	14	13	12	11	10	9	
13	12	11	10	9	8	7	6	5	4	3	2	1	0	16	15	14	13	12	11	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	16	15	14	13	
0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	16	15	
2	1	0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
4	3	2	1	0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	…																			

Figure 2. The top right corner of an example 2D code for B=17, D=2, and V=3.

3. RECOVERY SYSTEM

The purpose of the recovery system is to find the payload P. The capture device only reads a cropped W by W subset of the array. The payload bit length B, row-to-row offset D, phase code representation method, and interleave period V are communicated from the encoder.

Recovery is achieved through the following steps:

(1) For each symbol in the 2D array, interpret its value as a 0, 1 or abstain. Abstains are assigned when the symbol is too ambiguous to decode and will not contribute to the averages. (At least half of the cells in an encoded halftone will be abstains as they are all black or all white.)

(2) Determine which rows contain the phase codes. For each of V possibilities, find an estimate of a shifted payload P', and the Confidence value associated with that estimate.

For each candidate phase code row position, eliminate the phase code rows and compute an average for each bit position from the remaining D-shifted rows. For each bit position $\{b_{B-1}, \dots, b_1, b_0\}$ find an average of all non-abstained values and assign a value of 0 if that average is less than 0.5 and a value of 1 if that value is greater than or equal to 0.5. The uncertainty, u_j, for each bit position is then the absolute value of the difference between bit estimate and the average. The uncertainty of a bit will range from 0 to 0.5.

The Confidence associated with a candidate is then

$$\text{Confidence} = 1 - (2/B)\Sigma u_j, \text{ for } j=\{0, 1, 2, \dots, (B-1)\}.$$

and ranges from 0 to 1.0.

(3) Select the candidate set where P' has the highest confidence.

(4) Find the corresponding phase code U'. For each bit position $\{b_{B-1}, \dots, b_1, b_0\}$ find an average for of all non-abstained values and assign a value of 0 if that average is less than 0.5 and a value of 1 if that value is greater than or equal to 0.5.

(5) Convert P' into the standard form S by finding the minimum binary number value of the binary strings representing all B circular shifts of P'.

(6) Convert U' to U by circularly shifting it by the same amount as in step (5) to convert P' to S.

(7) Extract the Circular Phase C from the Phase Code U.

(8) Circularly shift S by C to get the payload P.

Because the unambiguous recovery of U is tied to the uniqueness of the standard form, it is clear why B must be prime to avoid multiple values of C. Otherwise one will be unable to find the phase of the phase code.

3.1 Recovery Example

To continue the example using the code from Figure 2 with B=17, D=2, and V=3, a 10x10 capture is shown in Figure 3. The correct bit positions of the payload in standard form are indicated in black, and the correct bit positions of the phase code rows are indicated in red. This information will not be known to the recovery system but is shown for reference. This sample is from an encoded halftone in an area of all highlight or all shadow, indicated by the gray and white cells in the figure; this is shown to indicate that only half of the bits will survive checkerboard subsampling.

4	3	2	1	0	16	15	14	13	12
6	5	4	3	2	1	0	16	15	14
8	7	6	5	4	3	2	1	0	16
10	9	8	7	6	5	4	3	2	1
12	11	10	9	8	7	6	5	4	3
14	13	12	11	10	9	8	7	6	5
16	15	14	13	12	11	10	9	8	7
1	0	16	15	14	13	12	11	10	9
3	2	1	0	16	15	14	13	12	11
5	4	3	2	1	0	16	15	14	13

Figure 3. 10x10 capture of a portion of the code in Figure 2.

Following the steps for recovery above:

(1) While not shown, the value of each symbol is recovered and stored as a 0, 1 or abstain.

(2) Since V=3, there are a total of 3 candidate sets of rows where the phase code is present. These three sets of rows are indicated in Figure 4. In each case the rows in-between the candidate phase code rows are assumed to be payload rows and evaluated. The corresponding bit positions are averaged and the uncertainties and confidence scores are generated.

(3) In this case, the correct set (in Figure 4(b)), results in the highest confidence score. Sets 1 and 3 will have payload bits combined with phase code bits, which will be different from payload bits, and thus will generate more uncertainty.

(4) The Figure 4(b) phase code rows are averaged to form a shifted U' code. The codes are 3 rows apart so the relative shifts are D*V = 2*3 = 6 bit positions.

With the correct shifted payload P' and phase code U', steps (5) through (8) can then be carried out to deliver the payload P.

(a)

16	15	14	13	12	11	10	9	8	7
1	0	16	15	14	13	12	11	10	9
3	2	1	0	16	15	14	13	12	11
5	4	3	2	1	0	16	15	14	13
7	6	5	4	3	2	1	0	16	15
9	8	7	6	5	4	3	2	1	0
11	10	9	8	7	6	5	4	3	2
13	12	11	10	9	8	7	6	5	4
15	14	13	12	11	10	9	8	7	6
0	16	15	14	13	12	11	10	9	8

(b)

16	15	14	13	12	11	10	9	8	7
1	0	16	15	14	13	12	11	10	9
3	2	1	0	16	15	14	13	12	11
5	4	3	2	1	0	16	15	14	13
7	6	5	4	3	2	1	0	16	15
9	8	7	6	5	4	3	2	1	0
11	10	9	8	7	6	5	4	3	2
13	12	11	10	9	8	7	6	5	4
15	14	13	12	11	10	9	8	7	6
0	16	15	14	13	12	11	10	9	8

(c)

16	15	14	13	12	11	10	9	8	7
1	0	16	15	14	13	12	11	10	9
3	2	1	0	16	15	14	13	12	11
5	4	3	2	1	0	16	15	14	13
7	6	5	4	3	2	1	0	16	15
9	8	7	6	5	4	3	2	1	0
11	10	9	8	7	6	5	4	3	2
13	12	11	10	9	8	7	6	5	4
15	14	13	12	11	10	9	8	7	6
0	16	15	14	13	12	11	10	9	8

Figure 4. The 3 candidate assignments of the phase code rows (bit positions shown in red).

4. ANALYSIS

The circular coding method as described has the flexibility to select a number of parameters. Once a payload size B is chosen, there is some question as to what the optimal value of row-to-row offset D and phase code row interleave period V should be. An important consideration for robust recovery is first assuring that all bits for the payload P and circular shift C survive, then knowing the number of repeat instances of each bit – the higher the number the better to combat the uncertainties in the print-scan process. For a given square crop window of width W, the number of times a payload or phase code bit is repeated depends on the position of the crop window and the nature of the checkerboard subsampling. While sample tests can offer some insight, it is more useful to exhaustively simulate all possible crop positions within a halftone checkerboard subsampled array of codes.

The repeat count varies for each bit. For payload recovery, the average bit repeat count is not as important as a guaranteed repeat count of some high percentage of the all code bits. The value of 90% is used in this study. For the one 10x10 crop in our example in Figure 3, counting the highlight or shadow subsampled payload (black numbered) bits we find that 4 bits are repeated 3 times, 10 bits are repeated 2 times and 3 bits survive only once. The guaranteed repeat count for 90% of the bits is then only 1. The guaranteed repeat count for the bits of the phase code also be so analyzed.

Figure 5 plots the guaranteed 90% repeat counts for B=17 and V=3 for all row-to-row offsets D and a range of small crop window sizes for both the payload and phase code. Each point on the graph is the aggregate of an exhaustive set of all crop configurations for the specified values of D and W. The point circled indicates the example case in Figure 3, with a 10x10 crop window and D=2. This testing clearly reveals that some choices for D are better than others, and that D=0,1,6,11, and 16 in particular should be avoided.

The reason for this test is that in the presence of noise in the print and scan process, the accuracy of correctly recovering any single bit is not perfect; thus, redundancy is needed. It is not practical to rely on a guaranteed repeat count of 1, so a larger crop window is needed.

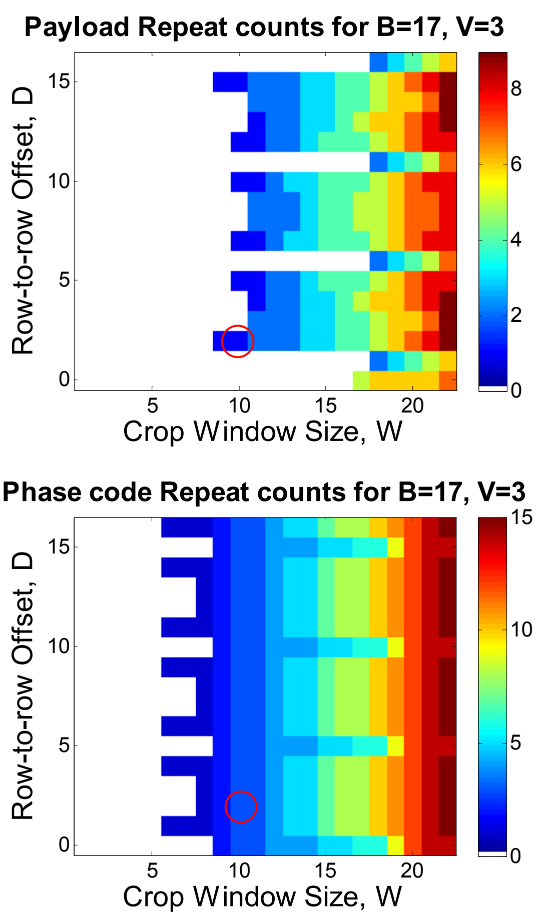

Figure 5. Minimum repeat counts for 90% of the bits in the payload and phase code as a function of crop window size and offset. Circled point corresponds to the example in Figure 3.

As phase code interleave period V decreases phase code repeat values increase but payload repeat counts decrease. For the larger payload size of B=67, examining the results for several choices of V showed that V=9 offers a good balance between these tendencies. Analysis of this case is shown in Figure 6. The plot shows that D=4 would be a good choice for row-to-row offset. Cropped captures as small as 30x30 will still have 90% of the bits repeated at least 4 times for both the payload and the phase.

The raw recovery rates for data encoded in clustered-dot halftone cells (stegatones) have been measured for a wide variety of printers and resolutions [5]. Combining this data with the minimum repeat count analysis allows for an efficient and robust design of a reference-free steganographic halftone generation and recovery system. The key advantage over the earlier method [4] is the ability to select a fixed row-to-row offset to optimize performance. The method allows complete data payload recovery from a cropped portion of the encoded image without the need for fiducial marks.

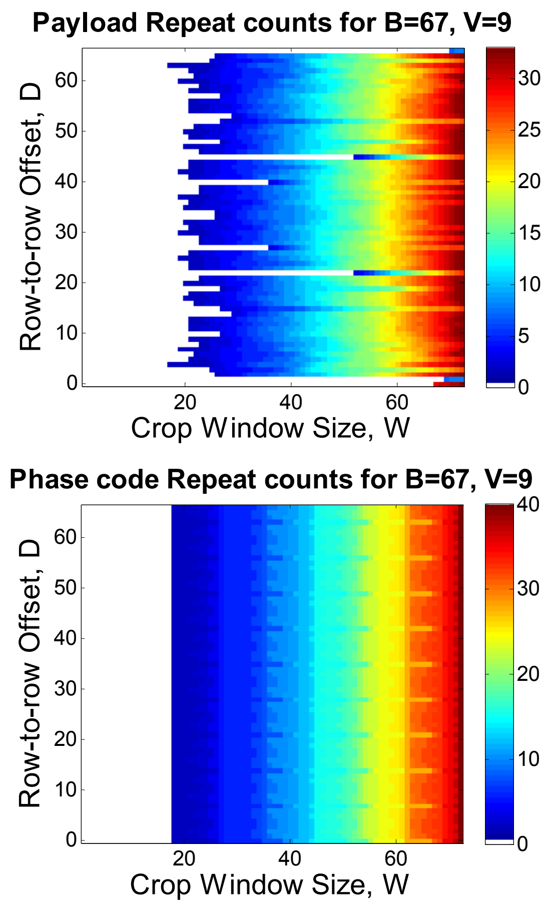

Figure 6. 90% repeat counts for B=67 and V=9.

5. REFERENCES

[1] Ulichney, R., Gaubatz, M., Simske, S., “Encoding Information in Clustered-Dot Halftones”, *IS&T NIP26 (26th Int. Conf. on Digital Printing Technologies)*, Austin, TX, 602-605, Sep 2010.

[2] Bulan, O., Sharma, G., and Monga, V., "Orientation Modulation for Data Hiding in Clustered-Dot Halftone Prints," *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2070-2084, Aug 2010.

[3] Solanki, K., Madhow, U., Manjanuath, B., Chandrasekaran,S., and El-Khalil, I., “’Print and Scan’ Resilient Data Hiding in Images,” *IEEE Transaction on Information Forensics and Security*, vol 1, pp 464-478, 2006.

[4] Ulichney, R., M. Gaubatz, S. Simske, “Circular Coding for Data Embedding”, *IS&T NIP29 (29th Int. Conf. on Digital Printing Technologies)*, Seattle, WA, Sep 2013.

[5] Chen, Y., Ulichney, R., Gaubatz, M., Pollard, S., “Stegatone Performance Characterization”, Media Watermarking, Security, and Forensics 2013, *IS&T/SPIE Electronic Imaging Symposium*, 8665-27, Feb 3-7, 2013.