Stegatone Performance Characterization*

Yung-Yao Chen[a], Robert Ulichney[b], Matthew Gaubatz[c], Stephen Pollard[d], Chun-Jung Tai[a], and Jan P. Allebach[a]

[a]School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-2035, U.S.A.
[b]Hewlett-Packard Laboratories, Stow, MA 01775, U.S.A
[c]Hewlett-Packard Laboratories, Seattle, WA 98109, U.S.A
[d]Hewlett-Packard Laboratories, Stoke Gifford, Bristol, U.K BS34 8QZ

ABSTRACT

Embedding data in hard copy is in widespread use for applications that include pointing the reader to on-line content by means of a URL, tracing the source of a document, labeling, and packaging. Most solutions involve placing overt marks on the page. The most common are 1D, 2D, and 3D (color) barcodes. However, while barcodes are a popular means for encoding information for printed matter, they add unsightly overt content.

In order to avoid such overt content, Stegatones are clustered-dot halftones that encode a data payload by single-pixel shifts of selected dot-clusters. In a Stegatone, we can embed information in images or graphics – not in the image file, as is done in traditional watermarking, but in the halftone on the printed page. However, the recovery performance of stegatones is not well understood across a wide variety of printing technologies, models, and resolutions, along with variations of scanning resolution. It would thus be very useful to have a tool to quantify stegatone performance under these variables. The results would then be used to better calibrate the encoding system. We develop and conduct a test procedure to characterize Stegatone performance. The experimental results characterize Stegatone performance for a number of printers, scanners, and resolutions.

Keywords: Stegatone, periodic clustered-dot halftone, calibration, print-scan systems

1. INTRODUCTION

Technology for enabling data-bearing hard copy is in widespread use for applications. These include security and forensic applications for tracing the source of a document, labeling, and packaging. Most solutions involve placing overt marks on the page. The most common are 1D, 2D, and 3D (color) barcodes.[1,2] While barcodes are a popular means for encoding information for printed matter, they add unsightly overt content. For the sake of aesthetics, it is more attractive than using barcodes if we can embed information in images or graphics – not in the image file as is done in traditional watermarking, but in the halftone on the printed page.

Hiding information in a continuous-tone image is often referred to as *watermarking* and has a long history of research.[3] However, through the extreme bi-level quantization inherent in halftoning, the nature of such encoded images changes considerably. Unavoidable distortions during the print-scan process affect data detection accuracy. Embedding data in halftoned hard copy therefore encounters new challenges other than those faced by traditional watermarking and has been studied by many researchers.[4–8]

For electrophotographic printers, periodic, clustered-dot screening is preferred to other halftoning approaches due to its low computational load, stable dot formation, and robustness to printer artifacts, such as dot gain

* This work was supported by the Hewlett-Packard Company.
Further author information: (Send all correspondence to Yung-Yao Chen.) 1 765 491 7160
Yung-Yao Chen: E-mail: chen298@purdue.edu, Telephone: 1 765 491 7160
Chun-Jung Tai: E-mail: tai1@purdue.edu, Telephone: 1 765 337 6575
Robert Ulichney: E-mail: u@hp.com, Telephone: 1 978 897 6223
Matthew Gaubatz: E-mail: Matthew.Gaubatz@hp.com, Telephone: 1 607 220 0035
Stephen Pollard: E-mail: stephen.pollard@hp.com, Telephone: +44 117 3162569
Jan P. Allebach: E-mail: allebach@purdue.edu, Telephone: 1 765 494 3535

and banding. By forming clusters, the possibility of dot development failure is significantly reduced, i.e., the stability of dot development is increased. In order to encode information in a monochrome, periodic, clustered-dot halftone, there are three methods: 1) encode by adding single-pixel shifts of dot-clusters;[9,10] 2) encode by changing orientation of dot-clusters;[11] and 3) encode by modulating dot size through laser intensity modulation.[12]

Stegatones[10] are clustered-dot halftones that encode a data payload by single-pixel shifts of selected dot-clusters. The selected dot-clusters are referred to as the carriers. In order to add data carrying capacity, variable-bit-length codes (1-bit, 2-bit, and 3-bit) are embedded in different carriers, which presents new challenges compared to simple 1-bit carriers. There is always a trade-off between data carrying capacity and recovery rate (bit error rate). It may be that the recovery rate for some carriers is very low, which would affect how those carriers would be used to encode data for the encoding system. In this case, even though the data carrying capacity will decrease, we might consider not using these carriers for the purpose of robust recovery accuracy (and to minimize the need for error correction codes). In addition, for different types of hardcopy prints, inherently different print qualities (print resolutions) are used. However, the recovery performance of Stegatones is not well understood across a wide variety of printing technologies, models, and resolutions, along with variations in scanning resolution. It would thus be very useful to have a tool to quantify Stegatone performance under these variables. The results would then be used to better calibrate the encoding system.

We design a test procedure to characterize Stegatone performance. To remove errors due to alignment and minimize the effect of print-scan distortions and neighboring carriers, a test pattern is designed where each carrier is surrounded by four fiducial marks. The distance between the fiducial marks is set to be large enough to ensure that the fiducial marks do not influence the dot development of the carrier dot-clusters. This test pattern will provide us the absolute best case result for decoding of each type of carrier.

The remainder of this paper is organized as follows. In Sec. 2, we briefly describe the carrier cell types and the Carrier Shift Rule used in our Stegatone system. In Sec. 3, we describe the test pattern, along with the analysis procedure that is used to characterize Stegatone performance with a wide variety of test variables. In Sec. 4, experimental results are provided. Finally, we draw our conclusions in Sec. 5.

2. PRELIMINARIES

In this paper, we represent a halftone image by a discrete brightness level between 0 (black) and 1 (white). We use boldface to indicate vector or matrix quantities. We use boldface lower case letters to indicate vector quantities, and boldface upper case letters to indicate matrix quantities. We use $(\mathbf{x}) = (x, y)^T$ and $[\mathbf{m}] = [m, n]^T$ to represent continuous and discrete spatial coordinates, respectively.

2.1 Periodic, Clustered-Dot Halftoning

Screening determines the halftone output by thresholding the discrete-space continuous-tone image $g[\mathbf{m}]$ with a threshold array $t[\mathbf{m}]$ pixel by pixel.[13,14] The threshold array of a periodic screen is spatially periodic according to a pair of linearly independent vectors, called the screen or tile vectors $\mathbf{n}_1$ and $\mathbf{n}_2$. The screen matrix $\mathbf{N}$ is defined by $\mathbf{N} = [\mathbf{n}_1|\mathbf{n}_2]$. Hence, the threshold array $t[\mathbf{m}]$ is periodic according to $\mathbf{N}$, i.e., $t[\mathbf{m} + \mathbf{N}\mathbf{q}] = t[\mathbf{m}], \forall \mathbf{q} \in Z^2$.

The nature of any halftone pattern generated by screening is dictated by a deterministic, periodic array of threshold values. Once the screen tile vectors are defined, the output halftone textures that screens generate are dependent on the spatial arrangement of the thresholds in the threshold array, and can be classified by whether they generate dispersed dots or clustered dots. The spatial arrangement of the thresholds within the threshold array is referred to as the dither template or index matrix $d[\mathbf{m}]$.

Figure 1(a) illustrates the dither template for a classical 45-degree clustered-dot screen with the two screen tile vectors $\mathbf{n}_1 = [8,\ 0]^T$ and $\mathbf{n}_2 = [0,\ 8]^T$. In this case, the values from 0 to 63 define the order in which each printer-addressable pixel will be turned *on* or *white*. With a 600 dot-per-inch (dpi) printer, the screen frequency will be 106 lines-per-inch (lpi).

The actual threshold values that will be compared against 8-bit input pixel values have to be normalized. To be mean-preserving, the following relationship is used to scale the dither template $d[\mathbf{m}]$ to the normalized threshold array $t[\mathbf{m}]$

$$t[\mathbf{m}] = 255 - \text{int}\left\{\frac{255}{|\mathbf{N}|}\left(d[\mathbf{m}] + \frac{1}{2}\right)\right\}, \tag{1}$$

where for notational simplicity, we let $|\mathbf{N}| \equiv |\det(\mathbf{N})|$. Therefore, for a input 8-bit continuous-tone image $g[\mathbf{m}]$, the resulting halftone image $h[\mathbf{m}]$ can be expressed by

$$h[\mathbf{m}] = \begin{cases} 1, & \text{if } g[\mathbf{m}] \geq t[\mathbf{m}] \\ 0, & \text{else} \end{cases}, \tag{2}$$

where the value 1 means turned *on* or *white*.

14	12	16	20	49	51	47	43
10	0	2	18	53	63	61	45
8	6	4	22	55	57	59	41
28	26	24	30	35	37	39	33
48	50	46	42	15	13	17	21
52	62	60	44	11	1	3	19
54	56	58	40	9	7	5	23
34	36	38	32	29	27	25	31

(a)

58	50	66	82	198	206	190	174
42	2	10	74	214	254	246	182
34	26	18	90	222	230	238	166
114	106	98	122	142	150	158	134
194	202	186	170	62	54	70	86
210	250	242	178	46	6	14	78
218	226	234	162	38	30	22	94
138	146	154	130	118	110	102	126

(b)

Figure 1. Illustration of halftoning framework for periodic, clustered-dot screen with the two screen tile vectors $\mathbf{n}_1 = [8,\ 0]^T$ and $\mathbf{n}_2 = [0,\ 8]^T$: (a) dither template $d[\mathbf{m}]$; and (b) normalized threshold array $t[\mathbf{m}]$. The 4×4 shaded regions depict shadow cells, and the 4×4 unshaded regions depict highlight cells.

2.2 Carrier Cells and Carrier Shift Rule

Equation (2) represents a standard clustered-dot halftone image $h[\mathbf{m}]$, which is referred to as the *reference halftone*. In the Stegatone system, the information is embedded by single-pixel shifts of selected dot-clusters of the reference halftone. The selected dot-clusters are referred to as the *carrier cells*.

Using the screen specified in Fig. 1, the 4×4 shaded regions depict shadow cells, and the 4×4 unshaded regions depict highlight cells. Shadow cells will be white hole-clusters surrounded by black background, and highlight cells will be black dot-clusters surrounded by white background. Figure 2(a) lists all 16 shadow cells, where the number of white hole-cluster pixels (carrier size of shadow cells) that comprise the center cluster is indicated. Likewise Figure 2(b) lists all 16 highlight cells with the number of black dot-cluster pixels indicated. Note that this list of cell shapes is, of course, dictated by the specification of the screen shown in Fig. 1.

Even though the information is embedded by shifting the carrier cells, the dot-clusters are constrained to be within the 4×4 region after being shifted, in order to maintain the image quality of a stegatone. As a result,

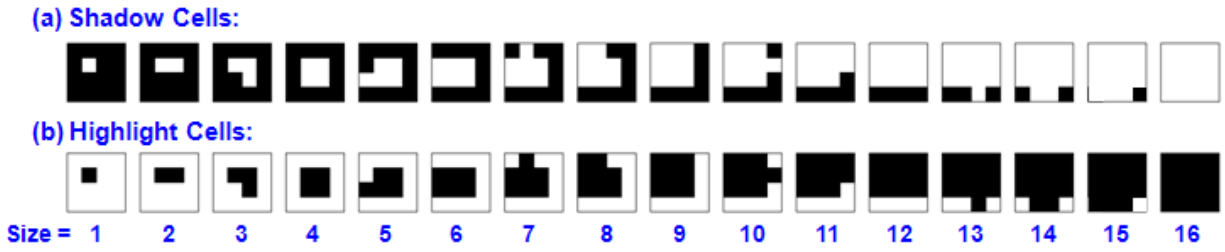


Figure 2. Illustration of carrier types if the dither template in Fig. 1 is used. White and black clusters for (a) shadow cells, and (b) highlight cells.

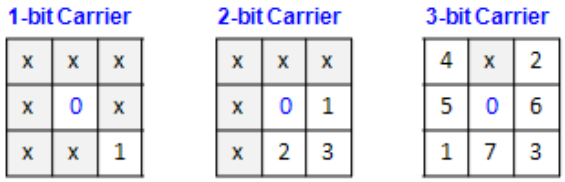


Figure 3. Carrier Shift Rule which defines how codes are represented.

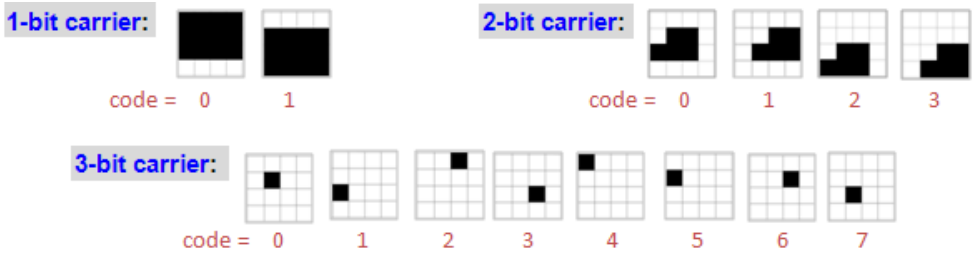


Figure 4. Shift positions for 1-, 2-, and 3-bit codes as defined by the Carrier Shift Rule.

not all clusters have the same degree of freedom to move; and it depends on the carrier size. The carrier cells that have room to move to at least 8 single pixel shift positions, and can carry 3 bits are defined as the 3-bit carriers. In both Figs. 2(a) and 2(b), carrier sizes 1 through 4 are the 3-bit carriers. Carrier sizes 5 through 9 can move to 4 positions, and are defined as the 2-bit carriers. Carrier sizes 10 through 15 can move to only 1 positon and are defined as the 1-bit carriers.

The mapping of 1-, 2-, and 3-bit codes to shift positions are defined by a *Carrier Shift Rule*, as shown in Fig. 3. The unshifted position is shown in the center of each 3×3 array; and it signals a code 0. An entry of "$\times$" is used to indicate shift positions that are not used, and non-zero entries indicate the shift position for the code of that particular value. For example, for the 2-bit carrier shift rule, embedding a code of "1" will cause a shift right by 1 pixel; embedding a code of "2" will cause a shift down by 1 pixel; embedding a code of "3" will cause a shift right by 1 pixel and a shift down by 1 pixel. The shift positions for example 1-bit, 2-bit, and 3-bit carriers are illustrated in Fig. 4. Note that our shift rule for 1-bit carriers (carrier sizes 10 through 15) shifts down and to the right with wrap-around. So, unlike sizes 1 through 9, the shapes of the shifted versions are different.

3. STEGATONE PERFORMANCE CHARACTERIZATION

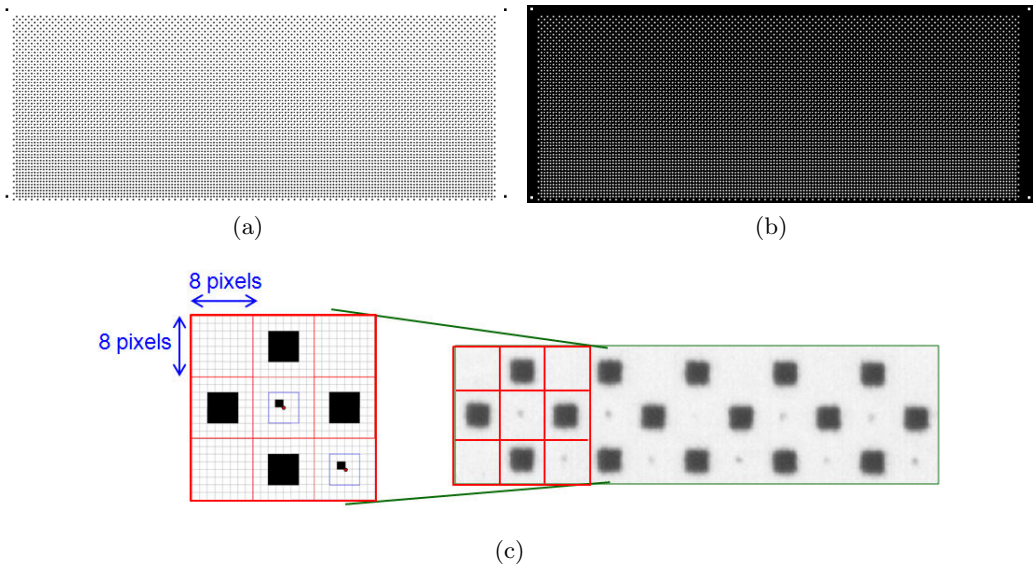
In Sec. 2.1, a classical 45-degree clustered-dot screen was introduced, which defines the carrier types for carrier sizes 1 through 15, as shown in Fig. 2. In Sec. 2.2, a Carrier Shift Rule was specified, which defines the shifted positions for 1-, 2-, and 3-bit codes. Some examples of the resulting shifted positions are shown in Fig. 4. For a Stegatone system, however, note that both the screen and the Carrier Shift Rule mentioned above are not unique. Different combinations of screens and Carrier Shift Rules have to be tested to better calibrate the Stegatone encoding system. It may be that the recovery rate for some carriers is very low, which would affect how those carriers would be used to encode data for the encoding system. The objective of this paper is to design a test procedure that enables us to effectively characterize Stegatone performance.

3.1 Design of Test Pattern: Fiducial-Based Pattern

Like other hardcopy embedding methods, the fundamental challenges of Stegatones are unavoidable printing and scanning distortions, which include non-affine warping. The alignment results might be different in different positions on the printed page or on the scanner plane. We found that every printer or scanner has slightly different horizontal and vertical pixel periods that must be accounted for. This means that the printer or scanner pixel lattices are not perfectly square and vary from position to position.

There are some factors which impact the accuracy of the printed and scanned page: first, the artifacts due to limitations of the printer mechanism, such as process direction velocity fluctuations, dynamic skew during paper feed, and paper warping; second, the artifacts due to limitations of the scanner mechanism, such as non-ideal scanner optics, non-constant scan-head velocity, and dynamic scan-head skew during the scanning process. To characterize Stegatone performance, it is useful to deal with printer issues and scanner issues as separately as possible. The scanner calibration can be done by using a mechanically-ruled resolution target. Hence, we can precisely record the horizontal and vertical scanner pixel pitch as a function of different positions. However, it takes time to calibrate every scanner that we want to use; and the problem of displacement between printer pixel lattice and scanner pixel lattice is not solved.

To minimize the impact of artifacts due to the printer and scanner mechanisms and the print-scan distortion, a test pattern is designed where each carrier is surrounded by four fiducial marks, as shown in Fig. 5. In this fiducial-based pattern, we print identical 80 carriers in each row. For 2-bit and 3-bit carriers (carrier sizes 1 through 9), the unshifted carrier type (i.e., code 0 position) is replicated 320 times or 4 rows to account for the overall printer instability. For 1-bit carriers (carrier sizes 10 through 14), since we want to test the Carrier Shift Rule with wrap-around, carrier types of both code 0 position and code 1 position are replicated 160 times or 2 rows, separately. For carrier size 15, we want to compare 2 different shifted pairs of code 0 position and code 1 position, which come from 2 different dither templates, as illustrated in Fig. 9. Hence, each carrier type is replicated 80 times in the same row.



(a) (b) (c)

Figure 5. Illustration of the fiducial-based pattern: (a) highlight carrier cells (carrier sizes 1 through 15) where black dot-clusters are surrounded by white background; (b) shadow carrier cells (carrier sizes 1 through 15) where white hole-clusters are surrounded by black background; and (c) enlarged version of the 4×4 highlight carrier (the blue square) in the printer pixel lattice, which is surrounded by four identical carrier fiducial marks (four 4×4 dot-clusters). The red dot inside the blue square represents the nominal carrier center; where left is digital image, and right is the scanned image that is printed at 600 dpi and scanned at 600 dpi. In (a) and (b), the larger global fiducial marks that lie in four corners are used to define the global skew vectors.

3.2 Locate Fiducial Marks and Carrier Cell Center

Although there may be large-scale geometric distortion in the scanned image due to the printer and scanner limitations discussed earlier, we assume that these distortions are not significant on the local scale corresponding to the region containing each carrier cell and its four surrounding fiducial marks, thus allowing us to focus on characterizing the Stegatone performance via the rendering ability of different printing technologies/options. In other words, as long as we can locate the centroids of the four surrounding fiducial marks precisely, the location

of the carrier cell center can be estimated precisely, as well. Unlike different carrier types, the fiducial marks are designed identically as 4×4 dot-clusters, and can easily be located in the scanned image.

As shown in Fig. 5(a), the larger fiducial marks which lie in four corners of highlight part of the test pattern are called *global fiducial marks*. The smaller fiducial marks that surround carrier cells are called *carrier fiducial marks*. We locate the top left global fiducial mark first, and locate the following carrier fiducial marks in a raster order. As shown in Fig. 5(c), the distance between two adjacent carrier fiducial marks in each row/column is equal; they are perfectly aligned in the horizontal/vertical direction. However, those properties are changed after print-scan distortion. The four global fiducial marks also provide us the two global skew vectors, the horizontal skew vector $\mathbf{v}_1$ and the vertical skew vector $\mathbf{v}_2$, for the entire scanned fiducial-based pattern. That is, once a carrier fiducial centroid is estimated to be at position $\mathbf{P}$, the position $\mathbf{P}+\mathbf{v}_1$ will be our preliminary estimation of the location of next carrier fiducial centroid in the same row, and the position $\mathbf{P}+\mathbf{v}_2$ will be our preliminary estimation of the location of next carrier fiducial centroid in the same column.

To locate fiducial marks, the scanned image is first transformed to the absorptance domain (0: white and 1: black), and is represented as $a[\mathbf{m}]$. The binary version $B[\mathbf{m}]$ of the scanned image $a[\mathbf{m}]$ can be expressed as

$$B[\mathbf{m}] = \begin{cases} 1\,(\text{black}), & \text{if } a[\mathbf{m}] \geq a_{\text{binary}} \\ 0\,(\text{white}), & \text{else} \end{cases}, \tag{3}$$

where the threshold a_{binary} is obtained by Ostu's method.[15] We use this binary image as a binary mask for the fiducial marks. The segmented mask regions are obtained by applying connected-component labeling (8-connected rule) to $B[\mathbf{m}]$, and the preliminary estimation that comes from the global skew vectors.

The centroid of each fiducial mark is calculated based on the spatial distribution of colorant absorptance throughout its corresponding mask region. Let D_i be the segmented region of the i^{th} fiducial mark in the binary mask image. Then, the horizontal and vertical centroids of the i^{th} segmented fiducial mark are given by

$$C_{x,i} = \frac{\sum_{[m,n]\in D_i} m \cdot a[m,n]}{\sum_{[m,n]\in D_i} a[m,n]},\ C_{y,i} = \frac{\sum_{[m,n]\in D_i} n \cdot a[m,n]}{\sum_{[m,n]\in D_i} a[m,n]}, \tag{4}$$

where $a[m,n]$ is the absorptance value of the scanned image at the pixel with coordinates $[m,n]$.

The location of each carrier center is estimated by the intersection of two line segments which come from its four surrounding fiducial marks: the line segment from the bottom fiducial centroid $\mathbf{C}_1 = (x_1, y_1)$ to the top fiducial centroid $\mathbf{C}_2 = (x_2, y_2)$; and the line segment from the left fiducial centroid $\mathbf{C}_3 = (x_3, y_3)$ to the right fiducial centroid $\mathbf{C}_4 = (x_4, y_4)$. The location of the carrier center can be expressed as

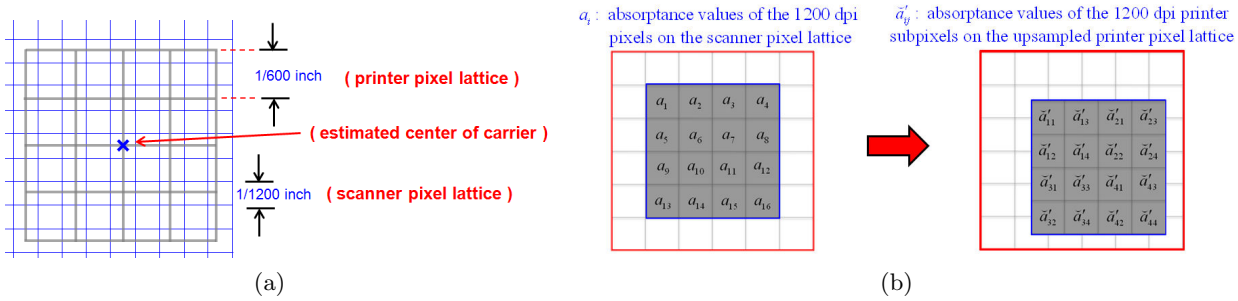
$$C_{carrier} = \mathbf{C}_1 + u \cdot (\mathbf{C}_2 - \mathbf{C}_1), \tag{5}$$

where the constant u is obtained by solving a linear equation and can be expressed as

$$u = \frac{(x_4 - x_3)(y_1 - y_3) - (x_1 - x_3)(y_4 - y_3)}{(x_2 - x_1)(y_4 - y_3) - (x_4 - x_3)(y_2 - y_1)}. \tag{6}$$

3.3 Deal with the Displacement Between the Printer Pixel Lattice and the Scanner Pixel Lattice

As shown in Fig. 4, in the decoding part of the Stegatone system, different shift positions within a 4×4 cell region represent different codes and need to be determined correctly. One thing that we want to know is the rendering ability of the target printers, i.e. can a given target printer render precise carrier shapes at the precise locations? The scanned image is represented in the discrete spatial-domain $\mathbf{m} = [m,n]$. However, there may be a displacement between the printer pixel lattice and the scanner pixel lattice, as shown in Fig. 6(a). The displacement will affect the system performance since we are using single-pixel shift encoding in the Stegatone system. In Eqs.(4) and (5), the estimated locations of the fiducial centroid and the carrier centers are in the continuous spatial-domain (not integers), which result from the displacement mentioned above and the deviations



(a) (b)

Figure 6. Illustration of the displacement between the printer pixel lattice and the scanner pixel lattice, and the process of upsampling the printer lattice to make estimation of printer pixel values easier. (a) in this example, the printer pixel lattice (gray lattice) has resolution 600 dpi, and the scanner pixel lattice (blue lattice) has resolution 1200 dpi. To understand more precisely the printer rendering ability, we want to estimate the absorptance values $a'_i, i = 1, ..., 16$ of a 4×4 carrier cell region of the printer pixel lattice from the scanned image pixel values $a_i, i = 1, ..., 64$, which are defined on the scanner pixel lattice. (b) To simplify this process, we upsample the printer lattice to match the resolution of the scanner lattice. Here, we show only a 2×2 portion of the 4×4 600-dpi printer carrier cell in (a). This part of the printer carrier cell corresponds to the 4×4 array of scanner pixels and the 4×4 array of upsampled printer pixels shown.

due to print-scan distortion. In order to deal with the non-integer estimated locations, the scanned image $a[\mathbf{m}]$ needs to be converted into the continuous spatial-domain.

The objective in this step is to estimate the absorptance values of each pixel on the printer pixel lattice from the scanned image which is defined on the scanner pixel lattice, as shown in Fig. 6(b). In other words, we want to know the corresponding absorptance values of the printer pixels. To simplify this task, we assume that the scanner resolution is an integer multiple of the printer resolution. For example, in Fig. 6(a), the printer has resolution 600 dpi; and the scanner has resolution 1200 dpi. Next, we upsample the printer lattice to match the resolution of the scanner lattice, as shown in Fig. 6(b). Then our task becomes one of estimating the absorptance values of the upsampled printer subpixels $\breve{a}'_{ij}$ from the underlying scanner pixels a_k. Here, the *breve* symbol is used to distinguish between printer pixels on the original printer lattice and printer subpixels on the upsampled printer lattice. The index i refers to the original printer lattice; and the index j refers to the subpixels $\breve{a}'_{ij}$ corresponding to a'_i. For the example in Fig. 6(a), there are four such subpixels corresponding to $j = 1, 2, 3, 4$. Once we have estimates for the absorptance of the printer subpixels $\breve{a}'_{ij}$, we will sum each group of four such estimated absorptances to obtain the estimated absorptance of the corresponding pixel a'_i on the original printer lattice. For example, as shown in Fig. 6(b), we will use the 1200 dpi scanner pixels a_k to estimate the absorptances of the 1200 dpi printer subpixels $\breve{a}'_{ij}$. The advantage of this approach is that the absorptance of each printer subpixel only depends on the absorptances of at most four underlying scanner pixels, as shown in Fig. 7.

The absorptance value of a certain scanner pixel in the discrete spatial-domain can be considered as the average absorptance value of that pixel area in the continuous spatial-domain. When the coordinates are both integers, this means exactly the top left corner of the pixel lattice. In other words, $a_i = a[m, n]$ implies that the average absorptance value of the area $\{(x, y) | \, m \leq x \leq m + 1, n \leq y \leq n + 1\}$ equals a_i. Each coordinate can be separated into the greatest integer less than or equal to that coordinate plus a fractional value, i.e. $x = \lfloor x \rfloor + \Delta x$ and $y = \lfloor y \rfloor + \Delta y$. As a result, given an estimated continuous-parameter location (x, y) that lies in the scanner pixel lattice, the corresponding absorptance value $\breve{a}'$ of the printer subpixel can be expressed as

$$\breve{a}' = (1 - \Delta x)(1 - \Delta y)a_1 + (1 - \Delta x)(\Delta y)a_2 + (\Delta x)(1 - \Delta y)a_3 + (\Delta x)(\Delta y)a_4 \, , \tag{7}$$

where a_1, a_2, a_3, and a_4 are the absorptance values of the four neighboring scanner pixels, as shown in Fig. 7. Currently, we only account for a displacement between the printer pixel lattice and the scanner pixel lattice. We assume there is no shear/skew within a carrier cell. The high decoding accuracies reported in Sec. 4 provide support for the validity of this assumption. Finally, as mentioned earlier, we simply sum each group of four printer subpixel values to obtain the corresponding printer pixel absorptance on the original printer lattice.

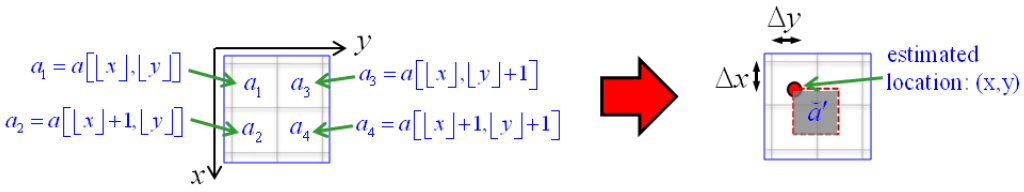


Figure 7. Illustration of how the absorptance value $\breve{a}'$ of each printer subpixel is estimated from the four absorptance values a_1, a_2, a_3, and a_4 of the neighboring scanner pixels.

Assuming a 2× difference between the scanner and printer resolution, we have

$$a'_i = \sum_{j=1}^{4} \breve{a}'_{ij} \,. \tag{8}$$

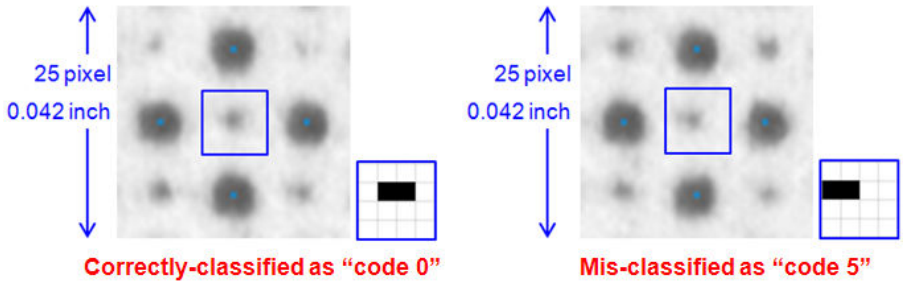
4. EXPERIMENTAL RESULTS

The objective of this paper is to design a test procedure which enables us to effectively characterize Stegatone performance under a wide variety of test variables (i.e., printing technologies, models, resolutions, along with variations of print mode and paper type). Each set of such test variables is referred to as one *virtual printer*; and we quantify the recovery rate of different carrier types that are printed by different virtual printers.

As discussed in the previous section, the absorptance values of each 4×4 carrier cell region of the printer pixel lattice are estimated by Eqs. (7) and (8). In order to quantify the recovery rate, we compare the sum of the absorptance values of the dot-clusters (or the white hole-clusters) in different shift positions. That is, s_i is defined as the sum of the absorptance values of the dot-clusters in the shift position i, i.e. $i = 0, 1$ for 1-bit carriers, $i = 0, 1, 2, 3$ for 2-bit carriers, and $i = 0, 1, ..., 7$ for 3-bit carriers. For the highlight cells (black dot-clusters surrounded by white background), the final decision is made by simply selecting the one with the largest sum, and for the shadow cells (white hole-clusters surrounded by black background), the final decision is made by simply selecting the one with the smallest sum, i.e.,

$$\text{code } i = \begin{cases} \arg\max\limits_i s_i, & \text{highlight cells,} \\ \arg\min\limits_i s_i, & \text{shadow cells.} \end{cases} \tag{9}$$

For this project, we have analyzed 846 combinations of printer model, printer units of the same model, media, print mode, scanner model, and scanner mode. For each case, we cropped from the printed and scanned test page a correctly-classified case and all mis-classified cases in order to provide a visual basis for understanding the recovery performance of that particular carrier cell when printed and scanned under that particular condition. Figures 8(a) and 8(b) illustrate these images for a highlight carrier with size 2 and code 0. In this case, there was only one mis-classification. In addition, for each printer/scanner condition and carrier type, we recorded two 6×6 matrices of statistics (the mean value and the standard deviation) over all printed replications of it. These are illustrated in Figs. 8(c) and 8(d), again for a highlight carrier with size 2 and code 0. Here, the blue box shows the carrier cell region; and the yellow-shaded region indicates the correct dot positions for this particular carrier. These statistics can help to explain the observed recovery rates for a given printer/scanner condition and carrier type. For example, Fig. 8(e) shows the mean and standard deviation data contained, respectively, in the third row of the matrices in Figs. 8(c) and 8(d), plotted as a function of the column index in these matrices. We can see that the mean absorptance values of the middle two pixels exceeds the mean absorptance values of the other four pixels by more than three standard deviation units. Since the middle two pixels are the correct dot locations for this particular carrier, we would expect a very high recovery rate; and indeed, the recovery rate for this printer/scanner condition and carrier is 99.84%.



(a)

(b)

Mean value (over 320 marks)

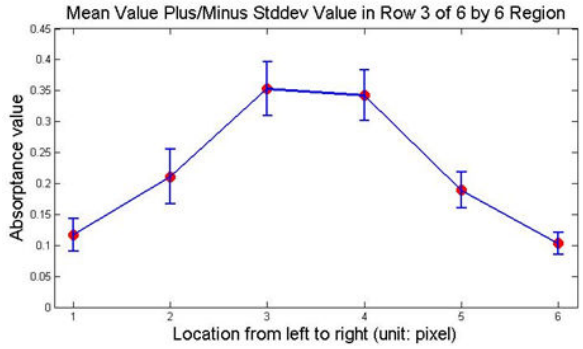
0.086	0.108	0.129	0.126	0.101	0.081
0.107	0.167	0.246	0.237	0.148	0.094
0.117	0.211	0.353	0.342	0.189	0.103
0.100	0.161	0.254	0.248	0.153	0.095
0.081	0.101	0.129	0.129	0.102	0.081
0.072	0.078	0.086	0.087	0.080	0.073

(c)

Standard deviation (over 320 marks)

0.017	0.020	0.023	0.022	0.018	0.015
0.024	0.035	0.038	0.034	0.025	0.017
0.026	0.044	0.043	0.041	0.029	0.018
0.020	0.030	0.034	0.030	0.025	0.018
0.015	0.018	0.021	0.021	0.018	0.016
0.015	0.014	0.015	0.016	0.015	0.013

(d)



(e)

Figure 8. Example illustrating analysis of highlight carrier cell size 2, code 0, for which 320 identical samples are printed in the test pattern. Blue squares are overlaid to show the 4×4 carrier cells, where the yellow shaded region represents the correct position of the dot-cluster for code 0. The scanned image was printed by virtual printer no. 1 at 600 dpi (color Laser EP, color mode, plain paper) and scanned by the Epson Expression 10000XL at 600 dpi: (a) the case which is correctly-classified as code 0. The cyan dots in the four surrounding fiducial marks represent the estimated centroid locations; (b) the case which is mis-classified as code 5; (c) mean values in absorptance domain within a 6×6 region, where the inner 4×4 region corresponds to the carrier cell; (d) standard deviation values within the same 6×6 region; and (e) plot of mean value plus/minus standard deviation in row 3 of the 6×6 region. In this example, the recovery rate is 99.84%.

These statistics can also help us to quickly identify a suitable selection of carrier type and Carrier Shift Rule for a target printer/scanner condition. For example, Fig. 9 shows part of the scanned image of a test page that was generated using two different dither templates. It was printed by virtual printer no. 2 (color Laser EP, color mode, plain paper) at 600 dpi and scanned with an Epson Expression 10000XL at 600 dpi. Note that code 0 of each carrier cell is the unshifted position, which is directly determined by its dither template. Here, the 1-bit Carrier Shift Rule shown in Fig. 3 is used (embedding a code of '0" will be no shift; embedding a code of "1" will cause shifts down and to the right with wrap-around). Let us compare the statistics which come from the two different dither templates. Figures 9(b) and 9(c), respectively, show the statistics of the carrier type code 0 from the first dither template and the resulting code 1 after applying a 1-bit Carrier Shift to the carrier cell shown in Fig. 9(b). Figures 9(e) and 9(f), respectively, show the statistics of the carrier type code 0 from the second dither template and the resulting code 1 after applying a 1-bit Carrier Shift to the carrier cell shown in Fig. 9(e). As we can see in Fig. 9(c), due to the dot-gain effect, the mean absorptance value of the only white hole within the 4×4 carrier region is even higher than that of the pixel to its left. This means that it is very hard to detect this carrier type. In this example, using the second dither template would be a better choice. These statistics are consistent with the scanned image of the test page shown in Fig. 9(g), where the four indicated rows are each printed with identical carrier types of Figs 9(b), 9(c), 9(e), and 9(f). If the carrier type contains a white hole that is not in the corner, it is very hard to detect, as shown in the second indicated row in Fig. 9(g).

In this paper, we test 21 virtual printers. In order to obtain a general understanding of Stegatone performance, we further divide these virtual printers into five subsets according to their settings, as shown in Table 1. For all virtual printers, we print the test patterns at two print resolutions (400 dpi and 600 dpi) and scan them at 600 dpi with an Epson Expression 10000XL. Figures 10 and 11 show the average recovery rate and the standard deviation of each subset. Figures 10(a) and 10(b) show the recovery rates of highlight carriers and shadow carriers, respectively, when the test page is printed at 600 dpi. For both cases, the recovery rate of carrier size 1 is the lowest. This is due to the difficulty of rendering the isolated dot for highlight carriers and due to the dot-gain effect in shadow carriers. However, the recovery rate increases to more than 90%, as the carrier cell size increases to size 3 for highlight and size 5 for shadow. Over a wide middle range of carrier cell sizes, the recovery rate for both highlight and shadow is better than 90%. The recovery rate decreases slightly beyond carrier cell size 13 or 14 for both highlight and shadow carriers. We can observe the same general trends in Fig. 11 when the test page is printed at 400 dpi. However, comparing these results to those shown in Fig. 10, we see that printing the test page at a lower resolution increases the recovery rate dramatically across the entire range of carrier cell sizes for both highlight and shadow carriers.

Table 1. Settings of Each Printer Subset

	Printer Type	**Paper Type**	**Print Mode**	**Print/Scan Resolution**	**Number of Units**
Subset 1	Color Inkjet	Glossy	Color	400, 600 / 600 (dpi)	1
Subset 2	Color Inkjet	Glossy	Black	400, 600 / 600 (dpi)	2
Subset 3	Color Laser EP	Plain	Color	400, 600 / 600 (dpi)	8
Subset 4	B/W Laser EP	Plain	Black	400, 600 / 600 (dpi)	5
Subset 5	B/W Laser EP	Glossy	Black	400, 600 / 600 (dpi)	5

5. CONCLUSION

In this paper, we proposed a test procedure that enables us to effectively characterize Stegatone performance subject to a wide variety of test variables. We not only designed the fiducial-based test pattern, but also described how to deal with the print-scan distortion by local fiducial marks and how to deal with the displacement between the printer pixel lattice and the scanner pixel lattice. The resulting data provides the basis for optimizing the free parameters of the Stegatone encoding process. For example, even though each size 1 carrier can encode 3-bit data, we might not use it in the Stegatone system due to its low recovery rate. Or given a fixed Shift Rule, our data enables us to select a dither template that generates carrier cells with a higher recovery rate. From Figs. 10 and 11, over a wide range of carrier cell sizes, we have demonstrated better than 90% recovery for both highlight and shadow carriers. The quantified recovery rates can be used to better tailor error correction coding schemes used as part of the Stegatone process.

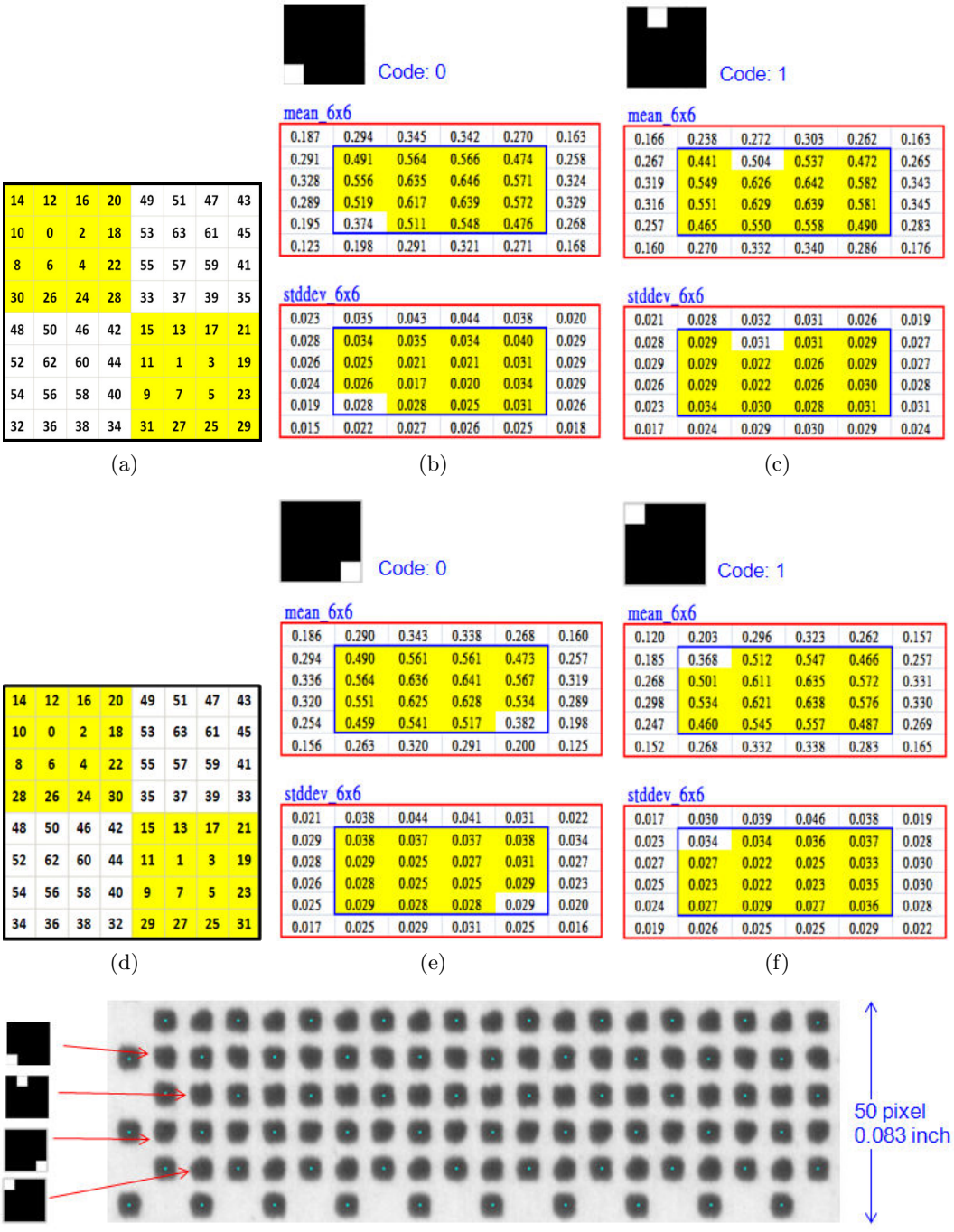
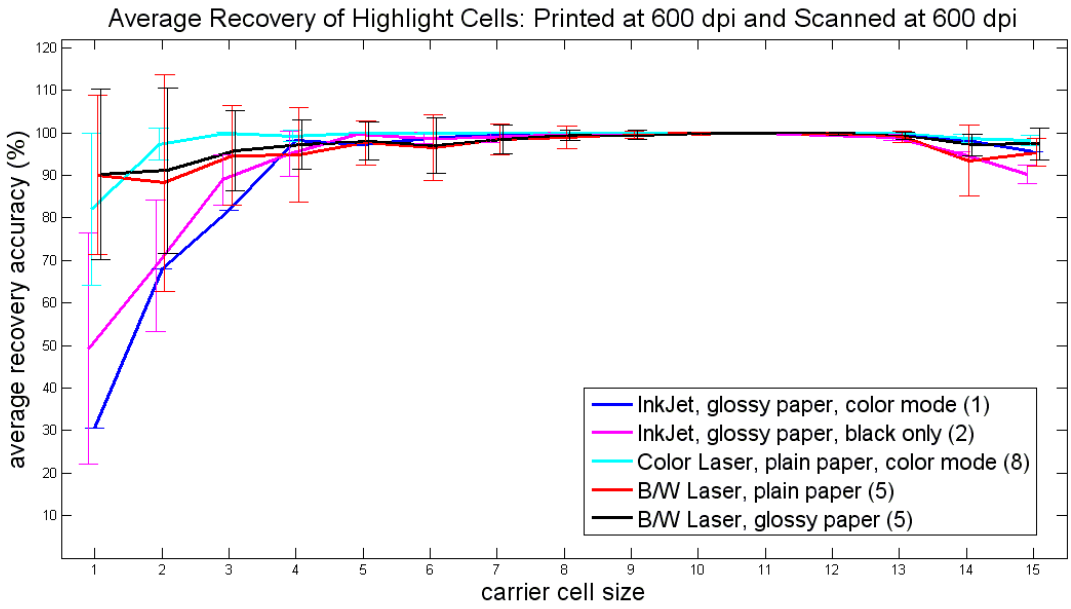
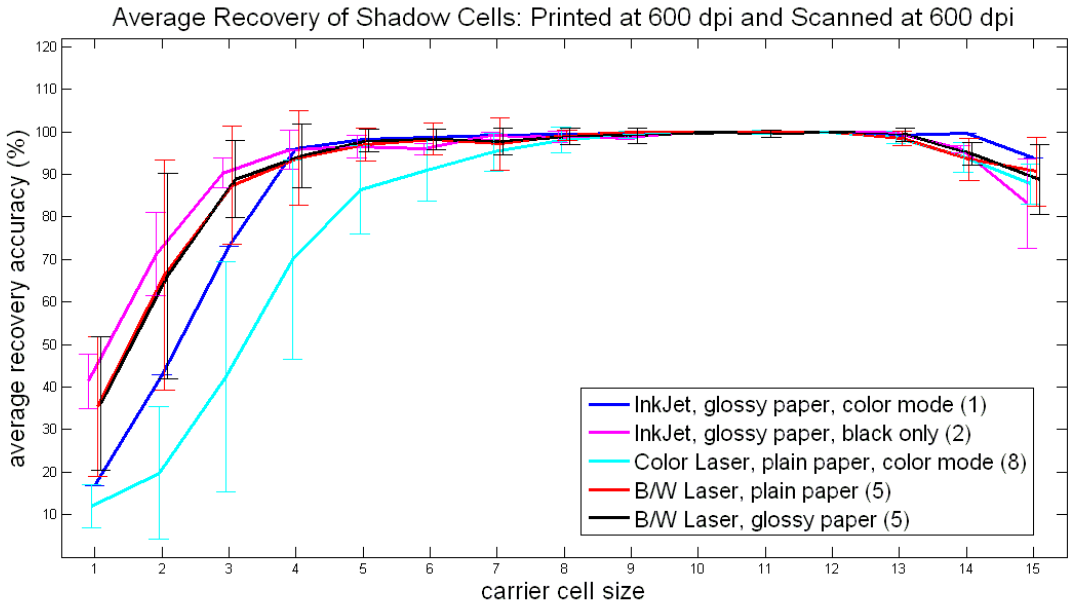


Figure 9. Illustration of how statistics (mean and standard deviation) from fiducial-based patterns can be used to select suitable carrier types for a target printer. Here, we assume that the 1-bit Carrier Shift Rule is determined according to Fig. 3: (a) the first dither template; (b) carrier type of size 15, code 0, and the statistics from printed and scanned replications of the fiducial-based pattern, where the inner blue square represents the 4×4 carrier cell; (c) carrier type of size 15, code 1, which comes from applying the Shift Rule to (b), and its statistics; (d) the second dither template; (e) carrier type of size 15, code 0 and its statistics; (f) carrier type of size 15, code 1, which comes from applying the Shift Rule to (e), and its statistics; and (g) scan of test pattern that is printed by virtual printer no. 2 (color Laser EP, color mode, plain paper) at 600 dpi and scanned with an Epson Expression 10000XL at 600 dpi. The four indicated rows are each printed with identical carrier types of (b), (c), (e), and (f), respectively. It can be seen that for the code 1 from the first dither template (in second row), the white hole is not in the corner and is very hard to detect, which is consistent with the statistics in (c), i.e. the absorptance value of that white hole position is very close to those of other printed positions. In this example, the recovery rates for these four carrier types are 100%, 16.28%, 100%, and 98.75%, respectively.

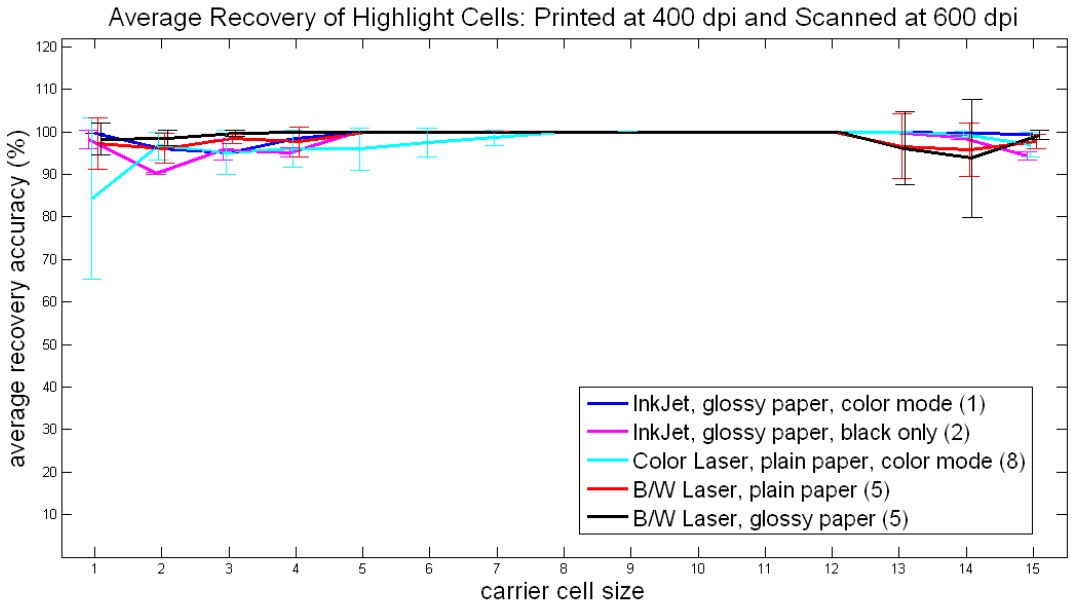


(a)

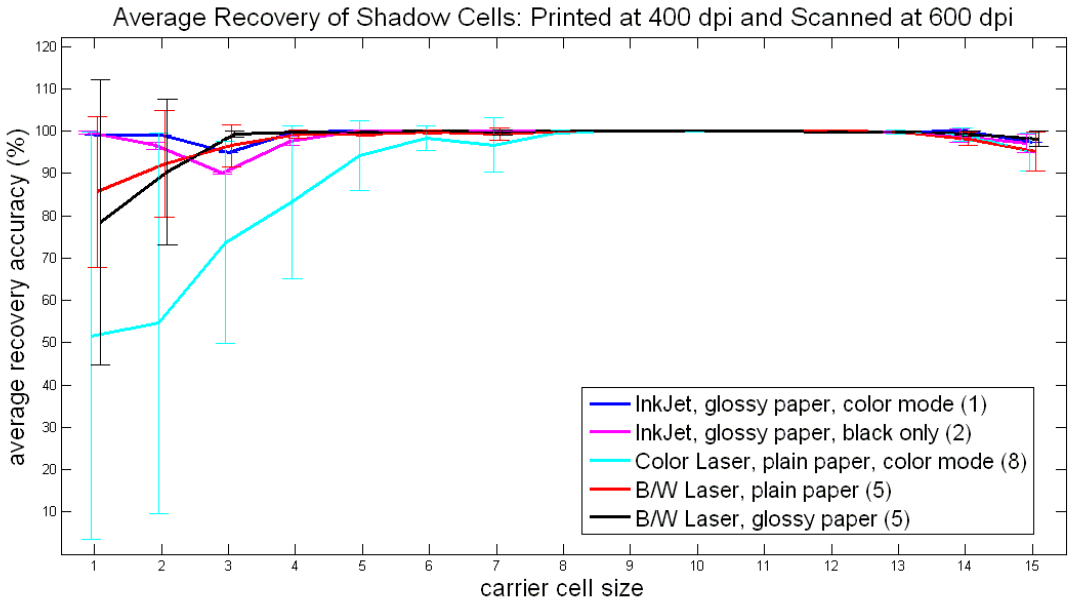


(b)

Figure 10. Recovery rate for both highlight and shadow carriers when the test page is printed and scanned at 600 dpi. The virtual printers are divided into 5 subsets according to their settings; and the number in the parenthesis represents the number of units in each subset. The average recovery rate of each subset is shown by the solid line, and the standard deviation of each subset is added as the error bar. The error bars for different subsets are shifted slightly in the horizontal direction, in order to avoid overlap. (a) recovery rate of highlight carriers, and (b) recovery rate of shadow carriers.



(a)



(b)

Figure 11. Recovery rate for both highlight and shadow carriers when the test page is printed at 400 dpi and scanned at 600 dpi. The virtual printers are divided into 5 subsets according to their settings; and the number in the parenthesis represents the number of units in each subset. The average recovery rate of each subset is shown by the solid line, and the standard deviation of each subset is added as the error bar. The error bars for different subsets are shifted slightly in the horizontal direction, in order to avoid overlap. (a) recovery rate of highlight carriers, and (b) recovery rate of shadow carriers.

REFERENCES

[1] Damera-Venkata, N., Yen, J., Monga, V., and Evans, B., "Hardcopy image barcodes via block-error diffusion," *IEEE Trans. Image Processing* **14**, 1977–1989 (Dec. 2005).

[2] Villán, R., Voloshynovskiy, S., Koval, O., and Pun, T., "Multilevel 2d bar codes: towards high capacity storage modules for multimedia security and management," *IEEE Trans. Information Forensics and Security* **1**, 405–420 (Dec. 2006).

[3] Podilchuk, C. and Delp, E., "Digital watermarking: algorithms and applications," *IEEE Signal Processing Magazine* **18**, 33–46 (Jul. 2001).

[4] Knox, K. T. and Wang, S., "Digital watermarks using stochastic screens," *Proc. SPIE, Color Imaging: Device Independent Color, Color Hard Copy, and Graphic Arts II* **3018**, 316–322 (Feb. 1997).

[5] Fu, M. S. and Au, O. C., "Data hiding watermarking for halftone images," *IEEE Trans. Image Processing* **2**, 477–484 (Apr. 2002).

[6] Kacker, D. and Allebach, J. P., "Joint halftoning and watermarking," *IEEE Trans. Signal Processing* **51**, 1054–1068 (Apr. 2003).

[7] Segovia, M. V. O., Chiu, G. T. C., and Allebach, J. P., "Using forms for information hiding and coding in electrophotographic documents," *IEEE International Workshop on Information Forensics and Security* , 136–140 (Dec. 2009).

[8] Chiang, P. J., Khanna, N., Mikkilineni, A. K., Segovia, M. V. O., Allebach, J. P., Chiu, G. T. C., and Delp, E. J., [*Intelligent Multimedia Analysis for Security Applications*], ch. Printer and Scanner Forensics: Models and Methods, 145–187, Springer-Verlag, Barrington, N.J., USA (2010).

[9] Shah, V., *Embedding and Detecting Machine Readable Data in Images Printed by Electrophotographic Printer*, Master's thesis, Purdue University, West Lafayette, IN 47907 (May 2008).

[10] Ulichney, R., Gaubatz, M., and Simske, S., "Encoding information in clustered-dot halftones," *IS&T NIP 26 (26th Int. Conf. Digital Printing Technologies)* , 602–605 (Aug. 2010).

[11] Bulan, O., Sharma, G., and Monga, V., "Orientation modulation for data hiding in clustered-dot halftone prints," *IEEE Trans. Image Processing* **19**, 2070–2084 (Aug. 2010).

[12] Chiang, P. J., Allebach, J. P., and Chiu, G. T. C., "Extrinsic signature embedding and detection in electrophotographic halftoned images through exposure modulation," *IEEE Trans. Information Forensics and Security* **6**, 946–959 (Sep. 2011).

[13] Ulichney, R., [*Digital Halftoning*], MIT Press, Cambridge, MA, USA (1987).

[14] Baqai, F. A. and Allebach, J. P., "Computer-aided design of clustered-dot color screens based on a human visual system model," *Proc. IEEE* **90**, 104–122 (Jan. 2002).

[15] Sezgin, M. and Sankur, B., "Survey over image thresholding techniques and quantitative performance evaluation," *J. Electronc Imaging* **13(1)**, 146 – 165 (2004).