

Hardcopy Image Barcodes Via Block-Error Diffusion

Niranjan Damera-Venkata, *Member, IEEE*, Jonathan Yen, Vishal Monga, *Member, IEEE*, and Brian L. Evans, *Senior Member, IEEE*

Abstract—Error diffusion halftoning is a popular method of producing frequency modulated (FM) halftones for printing and display. FM halftoning fixes the dot size (e.g., to one pixel in conventional error diffusion) and varies the dot frequency according to the intensity of the original grayscale image. We generalize error diffusion to produce FM halftones with user-controlled dot size and shape by using block quantization and block filtering. As a key application, we show how block-error diffusion may be applied to embed information in hardcopy using dot shape modulation. We enable the encoding and subsequent decoding of information embedded in the hardcopy version of continuous-tone base images. The encoding–decoding process is modeled by robust data transmission through a noisy print-scan channel that is explicitly modeled. We refer to the encoded printed version as an image barcode due to its high information capacity that differentiates it from common hardcopy watermarks. The encoding/halftoning strategy is based on a modified version of block-error diffusion. Encoder stability, image quality versus information capacity tradeoffs, and decoding issues with and without explicit knowledge of the base image are discussed.

Index Terms—Barcodes, halftoning, hardcopy security, information embedding.

I. INTRODUCTION

DIGITAL image halftoning quantizes a grayscale image to one bit per pixel for display and printing on binary devices. The goal of digital halftoning is to produce, via a clever distribution of binary dots, the illusion of continuous tone. Digital halftoning may be classified into three categories: amplitude modulated (AM), frequency modulated (FM), and AM-FM hybrid. In AM halftoning, the dot size is varied depending on the graylevel value of the underlying grayscale image while the dot frequency is held constant, e.g., conventional clustered-dot ordered dither screening. FM halftones have a fixed dot size, but the frequency of the dots varies with the graylevel of the underlying grayscale image. Conventional digital FM halftones have a fixed dot size of one pixel, e.g., those produced by dispersed-dot ordered dither and error diffusion [1]. AM-FM halftones [2]–[7] are hybrids that allow both dot size and dot frequency to vary in order to represent the underlying grayscale image.

Manuscript received July 21, 2003; revised November 23, 2004. N. Damera-Venkata and B. L. Evans were supported by the U.S. National Science Foundation CAREER Award under Grant MIP-9702707. N. Damera-Venkata conducted some of this research while at The University of Texas, Austin. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Reiner Eschbach.

N. Damera-Venkata and J. Yen are with the Imaging Systems Laboratory, HP Laboratories, Palo Alto, CA 94304-1126 USA (e-mail: damera@exch.hpl.hp.com; jyen@exch.hpl.hp.com).

V. Monga and B. L. Evans are with the Embedded Signal Processing Laboratory, The University of Texas, Austin, TX 78712 USA (e-mail: vishal@ece.utexas.edu; bevans@ece.utexas.edu).

Digital Object Identifier 10.1109/TIP.2005.859776

In this paper, we generalize conventional error diffusion halftoning to produce FM halftones with user-controlled dot size and shape [8]. We replace a pixel in conventional error diffusion with a pixel block. In a pixel block, the quantization error at each pixel is diffused to pixels in neighboring blocks in selected proportions. Hence, an entire block of quantization error is diffused at a time. The generated FM halftones can be designed to have very low dot size/shape variation, and the dot spacing is modulated depending on the underlying grayscale image. Unlike the aforementioned FM and AM-FM halftoning methods, the proposed block-error diffusion framework provides *explicit control* over the dot shape.

The idea of using block structures in error diffusion to generate clustered dot halftones is not new. Fan [9] describes a block-based error diffusion algorithm that combines traditional clustered-dot dithering and block-error diffusion to reduce the contouring observed when using traditional ordered dither screens with very few levels. While we also quantize pixel blocks, the key difference between Fan's work and ours is that Fan produces ordered dither halftones while we produce error diffused halftones with user-defined minority dot shapes.

Fan's method [9] is targeted at reducing contouring in traditional ordered dither screens. In contrast, our method modulates inter-dot distances between user-defined dot shapes to achieve tone reproduction (by determining minority/majority pixel blocks and quantizing them appropriately). Hence, our method is a block-FM halftoning method whose output resembles error diffusion. We also formalize the block-based mathematics in terms of linear algebra, which Fan's paper did not address.

Eschbach [10] proposes an error diffusion technique to generate clustered dot halftones using block threshold modulation. However, unlike our approach, Eschbach's approach does not quantize a pixel block at a time, nor does it diffuse a block of error. Instead, it is a scalar error diffusion system using threshold modulation that is block-periodic.

We use the dot shape control built into our proposed block-error diffusion framework to modulate dot shapes within an image according to an *information* signal. Thus, information may be encoded into the hardcopy version of an original image [11]. This is a key application of the block-error diffusion framework in which high information capacity is required.

Significant attention has been devoted to hardcopy watermarking by injecting watermarks into the halftoned image [12]–[18]. The methods in [12]–[14] are not practical for typical print-scan channels. The method in [15] employs a search over several halftone patterns, in a way similar to direct binary search [19] and is, therefore, very slow for real-time printing applications. While the methods in [16]–[18] are practical, their

aim is hardcopy authentication and not high-rate information embedding.

Data hiding for high-quality watermarking using scalar error diffusion has been explored in [20]. The authors encode information in the halftone by setting the binary state of preselected pixels of the halftone image. The quantization error due to these pixels is diffused as in conventional error diffusion. With this approach data may not be hidden in close proximity to other data carrying locations as this would create stability problems for the encoder. This significantly limits information embedding capacity.

Our method relies on dot shape modulation to embed information in a 2 bits/2 × 2 block or 1 bit/pixel. Information may be embedded in any pixel block. We aim for very high information capacity. Threshold image quality that enables the image to be recognized by a human is sufficient for our application. For this reason, we refer to the hardcopy image with information embedded in it as an image barcode (IBC).

In this paper, we show how to encode information into a continuous-tone base image, which is then printed using a conventional inkjet or laserjet printer. The printed image is scanned via a scanner and the resulting digital image is processed by a decoder to recover the transmitted message. The decoder may or may not have access to the original base image. The information capacity of an IBC lies between conventional two-dimensional (2-D) barcode technologies such as PDF-117 and hardcopy watermarks. In a typical application for example, we could embed the entire biography of an individual onto the photograph on his/her business card. Further, small mp3 clips and executables may also be embedded into hardcopy images providing a rich media experience.

Section II introduces the notation used in the paper. Section III formulates the framework of block-error diffusion. Section IV demonstrates the effect of designing the block-error filter from well known scalar error filter prototypes. We also discuss how FM halftones with user-controlled dot shape and size may be produced. Section V comprehensively describes a practical image communication system that encodes information into IBCs using the block-error diffusion framework. We show how the information embedded in the hardcopy images may be robustly decoded by processing the scanned IBC. Information capacity and encoder stability are also analyzed. Section VI summarizes the contributions of this paper and presents future research directions.

II. NOTATION

In this paper, we process vector-valued sequences using multifilters [21]. A multifilter is a filter with matrix-valued coefficients. Vector-valued sequences arise from grouping pixels in the input grayscale image to be halftoned into blocks of $N \times M$ pixels (rectangular blocks). We may order each block of MN samples into an $MN \times 1$ vector to form an image of vectors.

Letting \mathbf{x} be an image of vectors, with each vector having $MN \times 1$ elements (pixel values), the z -transform of \mathbf{x} is

$$\mathbf{X}(z_1, z_2) = \sum_{m_1} \sum_{m_2} \mathbf{x}(m_1, m_2) z_1^{-m_1} z_2^{-m_2}. \quad (1)$$

We may filter the vector-valued image using a multifilter. A multifilter with $K \times K$ support can be represented by a $K \times K$ sequence in which each sample is a $MN \times MN$ matrix. The z -transform of the matrix-valued filter (a.k.a. multifilter) sequence $\tilde{\mathbf{h}}$ is

$$\tilde{\mathbf{H}}(z_1, z_2) = \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} \tilde{\mathbf{h}}(k_1, k_2) z_1^{-k_1} z_2^{-k_2}. \quad (2)$$

An alternate representation of a multifilter with $K \times K$ support and $MN \times MN$ matrix-valued coefficients uses the $MN \times K^2 MN$ matrix

$$\tilde{\mathbf{T}} = [\tilde{\mathbf{h}}'(0) | \tilde{\mathbf{h}}'(1) | \dots | \tilde{\mathbf{h}}'(K^2 - 1)] \quad (3)$$

where $\tilde{\mathbf{h}}'(0), \tilde{\mathbf{h}}'(1), \dots, \tilde{\mathbf{h}}'(K^2 - 1)$ are the coefficients of the matrix-valued filter $\tilde{\mathbf{h}}$ ordered by rows. The filtering operation of a multifilter $\tilde{\mathbf{h}}$ with input \mathbf{x} is given by the matrix-vector convolution

$$\mathbf{y}(m_1, m_2) = \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} \tilde{\mathbf{h}}(k_1, k_2) \mathbf{x}(m_1 - k_1, m_2 - k_2) \quad (4)$$

where \mathbf{y} represents the output image of $MN \times 1$ vectors. In the z -domain, the matrix-vector convolution becomes a linear transformation by an $MN \times MN$ transformation matrix given by

$$\mathbf{Y}(z_1, z_2) = \tilde{\mathbf{H}}(z_1, z_2) \mathbf{X}(z_1, z_2). \quad (5)$$

For a scalar signal $x(m)$, we denote its z -transform by $X(z)$. We use \mathbf{m} to denote the 2-D index (m_1, m_2) , and \mathbf{k} to denote the 2-D index (k_1, k_2) .

III. BLOCK ERROR DIFFUSION

Fig. 1 shows a block diagram for block-error diffusion. Although the block diagram resembles conventional error diffusion halftoning, there are several key differences. The input is an $N \times M$ block of pixels (called a pixel block) as opposed to a single pixel in conventional error diffusion. We consider each block to be ordered into an MN -element vector as discussed in Section II.

The quantizer output for each pixel in a pixel block is exactly one element from the discrete set $\mathcal{O} = \{-1, 1\}$. Here, -1 represents black and $+1$ represents white. We quantize each pixel block using a scalar quantizer. The quantizer is defined by

$$\mathbf{Q}(\mathbf{u}) = \begin{pmatrix} Q(u_1) \\ Q(u_2) \\ \vdots \\ Q(u_{MN}) \end{pmatrix} \quad (6)$$

where

$$Q(u_i) = \begin{cases} 1, & u_i \geq 0 \\ -1, & u_i < 0 \end{cases} \quad (7)$$

Here, u_i refers to the i th pixel value in a $N \times M$ pixel block and, hence, i varies from 1 to MN . The output (quantization) levels are chosen to be -1 and $+1$ for the convenience of having midgray at 0. In Fig. 1, the filter in the feedback loop has matrix-

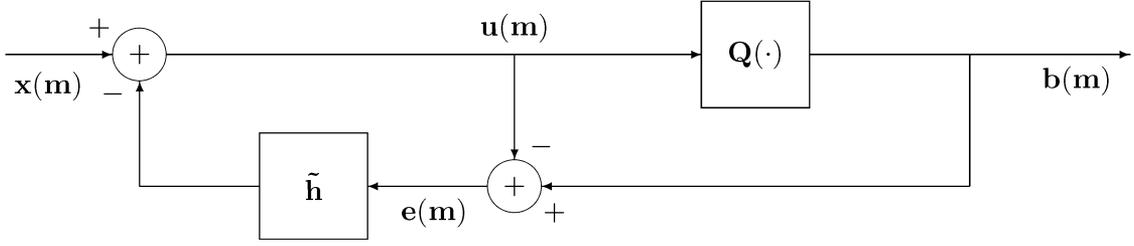


Fig. 1. System block diagrams for block-error diffusion halftoning where \tilde{h} represents the impulse response of a fixed 2-D nonseparable FIR error filter with matrix-valued coefficients. The vector \mathbf{m} represents the 2-D index (m_1, m_2) .

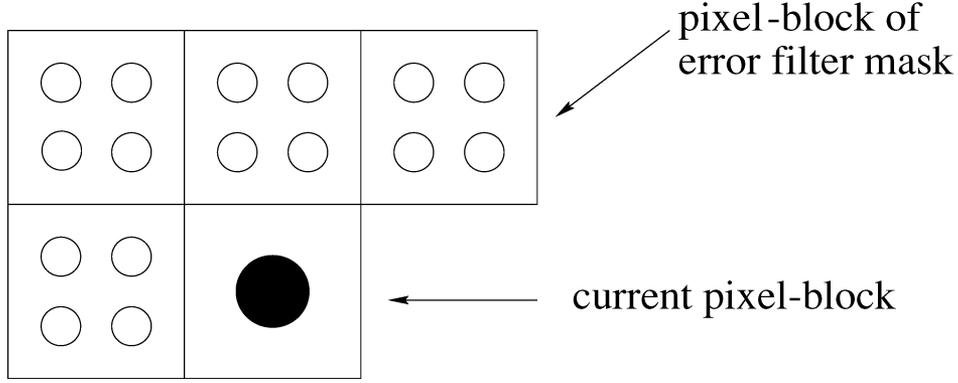


Fig. 2. Block-error filter operating on pixel blocks of 2×2 pixels. The shaded circle indicates the current pixel block. The unfilled circles indicate the error image pixels underlying the block filter mask. The pixels in the output pixel block are computed using four linear combinations of all 16 error pixels within the error filter mask.

valued coefficients. The filter operates on the quantization error sequence $\mathbf{e}(\mathbf{m})$ to produce the feedback signal

$$\mathbf{f}(\mathbf{m}) = \sum_{\mathbf{k} \in \mathcal{S}} \tilde{\mathbf{h}}(\mathbf{k}) \mathbf{e}(\mathbf{m} - \mathbf{k}) \quad (8)$$

where \mathbf{m} and \mathbf{k} are 2-D index vectors, $\tilde{\mathbf{h}}(\cdot)$ is an $MN \times MN$ matrix-valued sequence, and \mathcal{S} is the filter support. In this paper, we assume four-tap filter support defined by (horizontal, vertical) offsets to the current pixel being processed as $\mathcal{S} = \{(0, 1), (1, 0), (1, 1), (1, -1)\}$ unless otherwise specified. The input to the quantizer is given by

$$\mathbf{u}(\mathbf{m}) = \mathbf{x}(\mathbf{m}) - \mathbf{f}(\mathbf{m}). \quad (9)$$

In terms of block filtering the operation defined by (8) can be described with the help of Fig. 2, which illustrates a block-error filter operating on pixel blocks of 2×2 pixels. The output pixel block is computed by forming four different linear combinations of all 16 pixels in the pixel block mask. Each linear combination produces a single output pixel of the output pixel block. The next section shows how the matrix-valued coefficients of an error filter may be designed to promote user-defined minority pixel clustering in the generated halftone image.

IV. FM HALFTONING VIA BLOCK ERROR DIFFUSION

The block-error filter in the feedback loop governs how quantization error is diffused to the neighboring pixel blocks. For conventional error diffusion, one only needs to decide how much of the quantization error is to be diffused to each neighboring

pixel under the constraint that all the quantization error be diffused. The same applies in block-error diffusion, for which the constraints become

$$\tilde{\mathbf{\Gamma}} \mathbf{1}_{MNK^2 \times 1} = \mathbf{1}_{MN \times 1} \quad (10)$$

$$\tilde{\mathbf{\Gamma}} \geq 0 \quad (11)$$

where $\mathbf{1}_{r \times c}$ represents an $r \times c$ matrix of all one entries. These conditions correspond to the assertion that the elements of the matrix-valued error filter coefficients be nonnegative and that their rows sum to unity. Section IV-A designs error filters for block-error diffusion. Section IV-B generates FM halftones using block-error diffusion.

A. Error Filter Design

In designing the error filter coefficients $\tilde{\gamma}$ in (3), we map the coefficients of a conventional error filter into the corresponding block filters. If we start with a scalar filter with the same support as the multifilter or block filter, and represent its coefficients by the row vector $\tilde{\mathbf{\Gamma}}$, where

$$\tilde{\gamma} = [g'(0)|g'(1)|\dots|g'(K^2 - 1)] \quad (12)$$

then a multifilter $\tilde{\mathbf{\Gamma}}$ may be derived from it as

$$\tilde{\mathbf{\Gamma}} = \tilde{\gamma} \otimes \tilde{\mathbf{D}} \quad (13)$$

where \otimes denotes the Kronecker product operation and $\tilde{\mathbf{D}}$ is an $MN \times MN$ diffusion matrix. Since the elements of $\tilde{\gamma}$ are the coefficients of a conventional error filter, they are nonnegative and sum to one. Thus, to satisfy the constraints imposed by (10), the diffusion matrix must satisfy the constraints

$$\tilde{\mathbf{D}} \mathbf{1}_{MN \times 1} = \mathbf{1}_{MN \times 1} \quad (14)$$

$$\tilde{\mathbf{D}} \geq 0. \quad (15)$$

Thus, by imposing structure on $\tilde{\mathbf{F}}$, we only need to design the $MN \times MN$ diffusion matrix $\tilde{\mathbf{D}}$.

The decomposition of (13) is a natural and intuitive way of designing suitable error filters to generate FM halftones via block-error diffusion. The physical meaning of deriving the block filter from a given conventional error filter via (13) is that the quantization error incurred at the current pixel block is diffused to the neighboring pixel blocks in the same proportions that a conventional error filter diffuses error to its neighboring pixels. The diffusion matrix $\tilde{\mathbf{D}}$ governs the proportions to which errors are distributed within the pixels of a block. According to our proposed structure, these proportions are constant, and are independent of the relative position of the pixel blocks to which errors are diffused. This enforces a local isotropy constraint, by which we mean that no pixel within a pixel block is given preference over other pixels within the same block. The constraints on the diffusion matrix simply indicate that *all* of the quantization error that is diffused to a pixel block must be diffused among pixels that compose the block. Thus, the pixel blocks in the block-error diffusion framework are made to behave like pixels in conventional error diffusion and the block errors are diffused in much the same way as pixel errors in conventional error diffusion.

B. FM Halftoning

By using block-error diffusion, we show how to produce FM-halftones with dot clusters that are greater than one pixel in size. One method of achieving dot-clustering would be to halftone a downsampled version of the grayscale image and then replicate pixels to obtain a halftone of the same size as the grayscale image. For example, if we want 2×2 minority pixel dot clusters, then we could filter the original grayscale image with a halfband filter, downsample the original grayscale image by retaining every other sample in the horizontal and vertical directions, halftone the downsampled grayscale image, and interpolate the halftone to the resolution of the original grayscale image by pixel replication. This process is identical to halftoning the filtered image after replicating the upperleftmost sample in each block to all samples and using the identity diffusion matrix $\tilde{\mathbf{D}} = \tilde{\mathbf{I}}_{4 \times 4}$. Fig. 3 shows an example halftone generated by this method. The method first filters the original image (to prevent aliasing) and then performs downsampling. The downsampled image is then halftoned using conventional error diffusion. Pixel clustering is then induced by replicating each pixel to form a pixel block. The spatial resolution of the example halftone suffers due to the pixel replication and prefiltering.

Our approach to FM halftoning relies on forming minority pixel dot clusters by diffusing the quantization error from each pixel block equally to all samples within the neighboring pixel blocks. The error diffused to each block within the block-error filter mask will, however, be unequal since it is governed by the corresponding conventional error filter coefficients $\tilde{\mathbf{F}}$. Thus, for 2×2 pixel clusters, we use the diffusion matrix

$$\tilde{\mathbf{D}} = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}. \quad (16)$$



Fig. 3. Halftone generated by pixel replication induced block clustering. Here, the original image is filtered (to prevent aliasing) and downsampled. The downsampled image is then halftoned using conventional error diffusion. Pixel clustering is then induced by replicating each pixel to form a pixel block. Note the loss of high-frequency information and the blurred appearance.

In general, for an $N \times M$ pixel block, the diffusion matrix will take the form $(1/MN)\tilde{\mathbf{I}}_{MN \times MN}$ where $\tilde{\mathbf{I}}_{MN \times MN}$ is an $MN \times MN$ matrix with all its elements equal to 1. The motivation for using this diffusion matrix is that the error at any sample within the current pixel block that is diffused to an adjacent pixel block will be spread to all of the samples within the pixel block equally. The quantization decisions of all pixels within the modified pixel block will be biased in the same direction. Intuitively, this should result in the halftoned samples of that pixel block organizing themselves into a pixel block cluster.

Fig. 4 shows halftones obtained by using block-error diffusion with the diffusion matrix given by (16). Fig. 4(a) uses $\tilde{\gamma} = [(1/16) (5/16) (3/16) (7/16)]$, and Fig. 4(b) uses $\tilde{\gamma} = [(1/48) (3/48) (5/48) (3/48) (1/48) (3/48) (5/48) (7/48) (5/48) (3/48) (5/48) (7/48)]$, which correspond to the well-known Floyd–Steinberg [1] and Jarvis [22] error filters, respectively. The support for the Floyd–Steinberg and Jarvis error filters are shown in Fig. 5(a) and (b), respectively. For the rest of the paper, we fix $\tilde{\mathbf{F}} = [(1/16) (5/16) (3/16) (7/16)]$. There is no need to use a prefilter to prevent spatial aliasing. From visual inspection, the spatial resolution of the grayscale image is not compromised, and the dots are clustered in 2×2 blocks. The halftones even exhibit sharpening, which is characteristic of conventional Floyd–Steinberg and Jarvis error diffusion [23].

Using the method described above, it is possible to cluster the halftone dots into user-defined shapes and sizes. Halftones with dot clusters of 3×3 and 4×4 are produced using 9×9 and 16×16 diffusion matrices having all their elements equal to $(1/9)$ and $(1/16)$, respectively. Fig. 6(a) and (b) shows halftones with square dot clusters of 3×3 . Fig. 6(c) and (d) shows halftones with rectangular dot clusters of size 3×2 and 2×3 , respectively. The halftones of Fig. 6(c) and (d) was produced by using the diffusion matrix $(1/6)\tilde{\mathbf{I}}_{6 \times 6}$. Fig. 6(a)



Fig. 4. Block-error diffusion with block-error filters derived from conventional Floyd–Steinberg and Jarvis filters. Note the improved performance over pixel replication induced block clustering shown in Fig. 3.

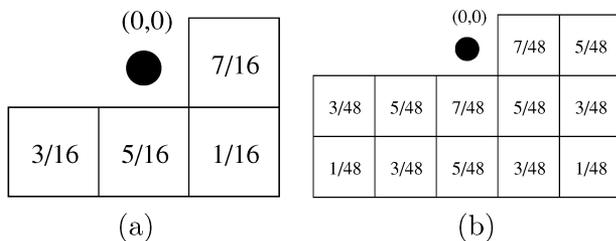


Fig. 5. Error filters commonly used in conventional error diffusion halftoning. The black dot represents the current pixel being halftoned. (a) Floyd–Steinberg. (b) Jarvis.

and (b) was produced by using the diffusion matrix $(1/9)\tilde{\mathbf{I}}_{9 \times 9}$, which is similar to the square block clustered-dot FM halftones described earlier. However, we quantize the minority pixel blocks by replacing them with the desired dot shape, while majority pixel blocks are quantized as per usual by using (7).

Fig. 7(a) and (b) shows the pixels within a pixel block (shaded) that are part of the dot shapes corresponding to the halftones of Fig. 6(a)–(d). The shaded pixels are to be interpreted as having a value -1 , while the unshaded pixels are to be interpreted as having a value $+1$. The quantization error is computed at each pixel block location as the vector difference between the output shape (if we are quantizing a minority pixel block) or the usual output pixel block (if we are quantizing a majority pixel block) and the corresponding pixel block in the input (image) to the quantizer. Thus, at each pixel block location, we need to determine if the current pixel block to be quantized will form a minority pixel block or a majority pixel block. This may be estimated by comparing the majority pixel type ($+1$ or -1) in the current pixel block quantized at mid-gray against the majority pixel type in the corresponding pixel block in the original grayscale image that is quantized at mid-gray. If the two are not equal, then the pixel block is a minority block and the output pixel block is replaced with the

desired dot-shape. If the two are equal, then the current pixel block is a majority pixel block, and the output pixel block is simply the quantization of the current pixel block at mid-gray. This is so since the minority pixel type depends on the input intensity level I . Assume $I \in [-1, 1]$. If $I < 0$, minority pixel type is $+1$ (white); i.e., dither patterns of dark levels have more dark than white pixels. If $I > 0$, minority pixel type is -1 (dark); i.e., dither patterns of bright levels have more white than dark pixels). Quantizing graylevel g at the mid-gray of 0 gives -1 if $g < 0$ and $+1$ if $g > 0$. This is opposite to the minority pixel type and the same as the majority pixel type.

V. IMAGE BARCODES

Our IBC system uses several components of the visually significant barcode (VSB) technology developed by Shaked *et al.* [24]. The VSB technology is a sophisticated image processing pipeline to enable efficient hardcopy information encoding on bi-level base images, such as company logo images. We make use of the print-scan channel model developed by Shaked *et al.* for VSBs. A VSB encodes information onto a bi-level base image by block-based XOR modulation, as illustrated by Fig. 8.

Photographic images, however, are continuous-tone and VSB is not directly applicable. If the continuous tone image is halftoned, an intermediate binary image results, on which VSB encoding may be attempted. This approach of generating a barcode representation of a grayscale input image is illustrated in Fig. 9. Fig. 10(a)–(c) shows encoded images that use various halftoning methods to produce an intermediate halftone from a grayscale base image.

VSB encoding is then performed on the intermediate halftones. Encoding information using the intermediate binary halftone image as a base image produces results with low visual quality, even when high-quality halftoning algorithms (such as error diffusion) are used to produce the intermediate

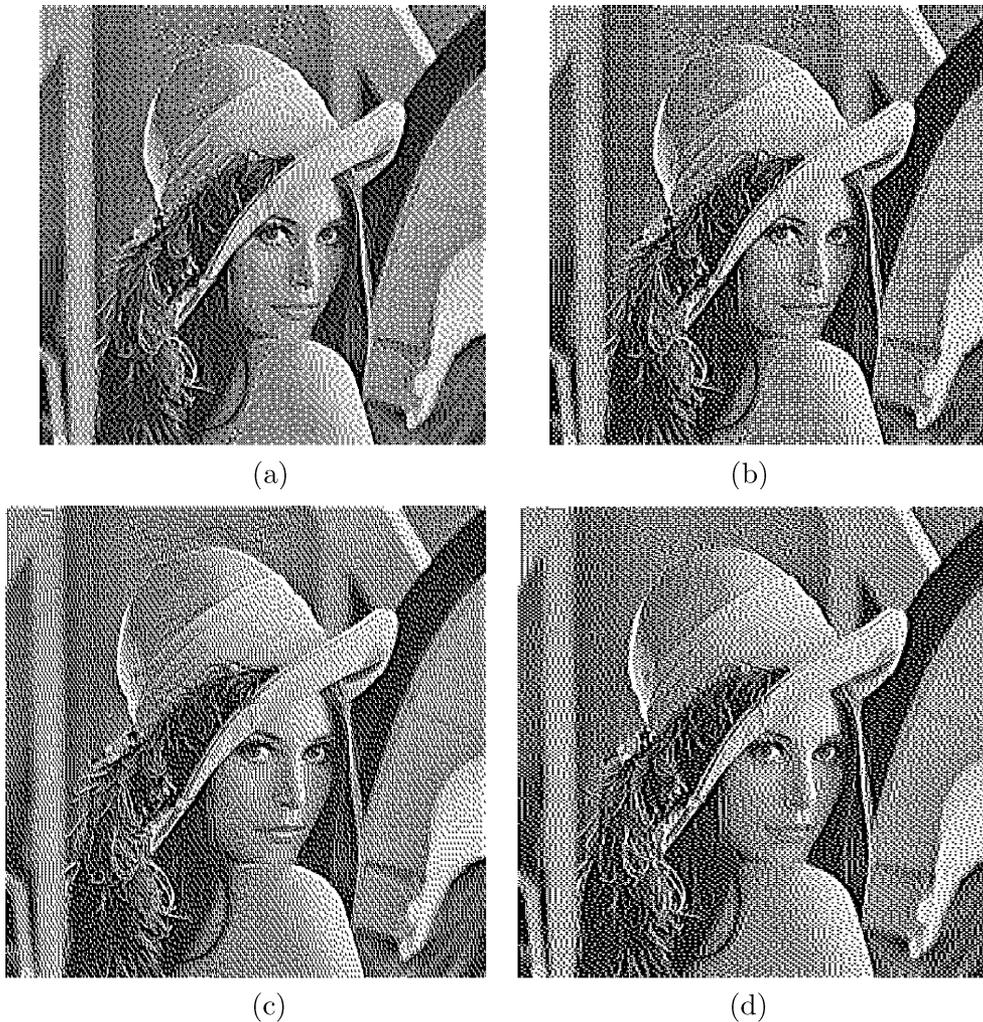


Fig. 6. Block-error diffused halftones with user controlled dot shapes. (a) "Multiply" dots. (b) "Plus" dots. (c) "L shape" dots. (d) "T shape" dots.

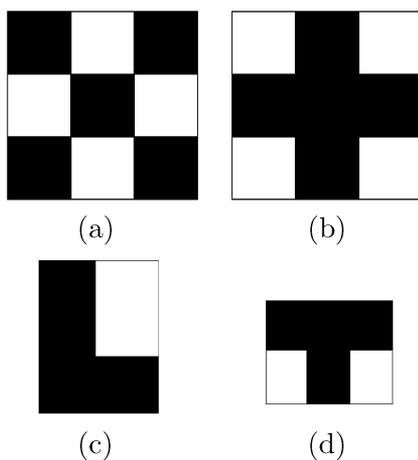


Fig. 7. FM halftone dot shapes. The shaded pixels indicate the pixels in the pixel blocks that are part of the halftone dot shape. (a) "Multiply" dots. (b) "Plus" dots. (c) "L shape" dots. (d) "T shape" dots.

halftone. The subsequent exclusive or (XOR) modulation based encoding disturbs the pixel distribution in the intermediate halftone, which yield the low visual quality in the encoded image.

The proposed IBC approach jointly encodes and halftones, based on the message symbols to optimize overall encoded image quality. This process is illustrated in Fig. 11. The halftoning stage has knowledge of the encoding stage and vice versa. Thus, the IBC is an extension of the VSB technology to handle continuous tone input images. Fig. 10(d) shows an example IBC.

An IBC is able to represent tone-variations in a given grayscale base image, while embedding a lot of information. This is achieved by using a modified version of block-error diffusion that supports joint encoding and halftoning. First an intermediate binary halftone image is generated that is dependent on the codewords as well as the base image. The output is encoded as a VSB on the resulting intermediate binary halftone image. Also the quantization errors due to the encoding are diffused to future blocks using block-error diffusion. In this manner, both halftoning and encoding can be performed jointly.

The IBC may be decoded from a scanned version of the printed image. When we know the grayscale base image at the decoder, we will show how we can use this information to decode the message signal using maximum-likelihood VSB decoding. Even when the base image is not known, decoding is possible. Under certain conditions, we may estimate the inter-

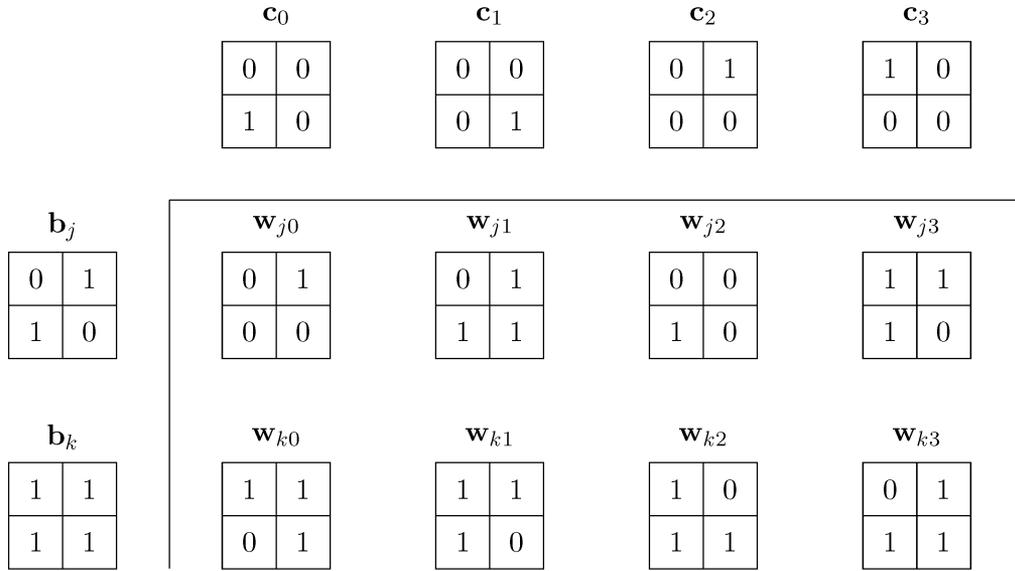


Fig. 8. Example XOR modulation encoding used in the VSB technology. The encoded block \mathbf{w} is the result of an XOR operation between the encoding codeword block \mathbf{c} and the bi-level base-image block \mathbf{b} . Thus, $\mathbf{w}_{lm} = \mathbf{c}_m \otimes \mathbf{b}_l$. Only two of the 2^4 possible base-image blocks are shown.

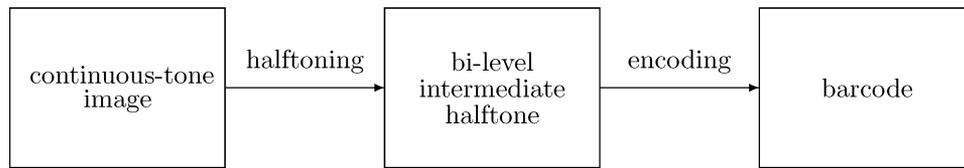


Fig. 9. Encoding continuous-tone images by generating an intermediate bi-level halftone and then using VSB encoding. The halftoning stage does not use knowledge of the encoding process.

mediate binary halftone directly from the observed data and use it to decode the message signal. We also discuss the stability of our proposed joint encoding and halftoning strategy and show how image quality may be traded for increased capacity.

Section V-A reviews VSB technology. Since we use many features of the VSB in our IBC system, we emphasize aspects of the VSB that are common to the IBC. Section V-B introduces the joint halftoning and encoding strategy used to encode information into an IBC. We show that the proposed algorithm is able to represent any continuous tone variation in a stable manner and is able to trade image quality for increased capacity. Section V-C discusses how a scanned IBC may be decoded when the original grayscale base image is either explicitly known at the decoder or is unknown at the decoder. We refer to the former case as guided decoding and the latter case as blind decoding.

A. Visually Significant Barcodes

Here we provide an overview of VSBs [24] for completeness. The VSB encoder first partitions the bi-level base image into blocks. At each block location a symbol is selected from a codebook and the output is computed by performing an XOR operation between the input block and message symbol block. This operation is illustrated in Fig. 8. If 2×2 blocks are used to encode four different codewords, then 2 bits are encoded per base-image block. To ensure robust transmission the codeword bitstream is mapped to a more redundant bitstream using error correcting codes. For example, if a $16 \rightarrow 31$ BCH code were

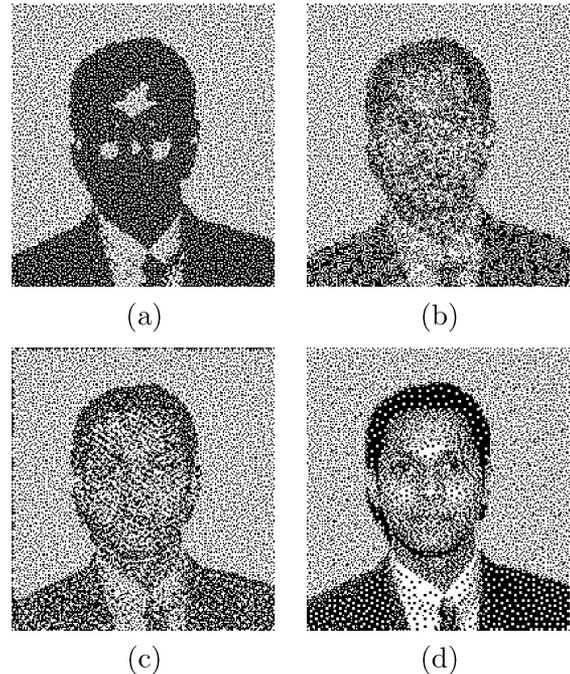


Fig. 10. Barcodes generated from various intermediate halftones. (a) Intermediate halftone generated via thresholding. (b) Intermediate halftone generated by Floyd–Steinberg error diffusion. (c) Intermediate halftone generated by Levien’s green noise error diffusion. (d) Intermediate halftone generated by proposed joint encoding/halftoning strategy. The actual rendering resolution is 150 dpi. The images may be viewed from a distance to simulate typical viewing.

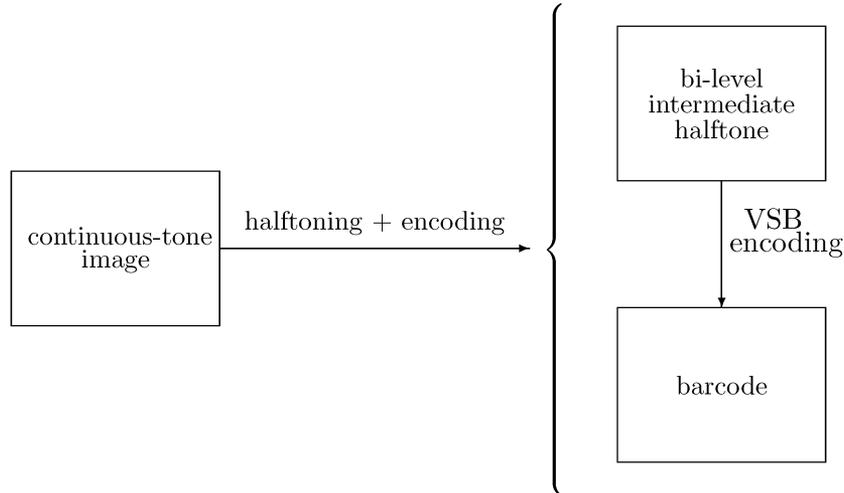


Fig. 11. Joint encoding and halftoning of a continuous-tone image to produce an IBC. The halftoning stage uses knowledge of the encoding process to improve overall encoded image quality. The encoded halftone is still a VSB with the intermediate halftone regarded as the base bi-level image.

used, the actual information capacity per block is halved. Fiducial marks are inserted at the corners to keep the decoder in alignment. Fig. 8 shows that the XOR encoding using two different codewords on two different base-image blocks can produce the same output encoded image block. Thus, the bi-level base image must be known or estimated if we were to estimate the message symbols from the output halftone symbols. If the target VSB print resolution is 100 dpi and the printer has a native resolution of 600 dpi, then each encoded halftone pixel is replaced by a 6×6 block by pixel replication before printing.

The VSB decoder operates on a scanned version of the encoded hardcopy image. Let us assume that the scanning resolution is 600 dpi. Then an observed 6×6 block corresponds to one encoded halftone pixel. The decoder first identifies fiducial marks to enable the decoder to compensate for global geometric distortions such as global translation, rotation and skew due to the print scan process. This step involves corner detection, estimation and application of a global geometric transform followed by bilinear interpolation. The result of this step is a rectangular image. Before the VSB may be decoded using maximum likelihood detection theory, the halftone dots in the encoded image must be matched up with corresponding grayscale observations in the scanned image. The VSB accounts for local space deformations arising due to the fact that dots corresponding to certain coordinates in the original image are located at different points in the copy. Most deformations were observed to be approximately separable, i.e., a dot expected at (x_0, \cdot) is located at $(x_0 + \Delta_x, \cdot)$, and similarly a dot expected at (\cdot, y_0) is located at $(\cdot, y_0 + \Delta_y)$. Row and column interfaces are aligned with pixel rows and columns, and the deformation is expressed only in their uneven distribution. The sum of absolute values of horizontal gradients in columns is used to determine the interface at columns (from peaks in the gradient sum). A similar procedure is used to find row interfaces. Dots are then virtually aligned by augmenting the scanned image with a list (describing a a potentially nonuniform grid) of dot centers.

Once the scanned barcode has been aligned with the encoded halftone, there is a correspondence between a measurement patch and an encoded halftone pixel. A linear discriminant y is

formed by multiplying the elements of the patch pointwise by a truncated Gaussian and summing. This reduces the dimensionality of the observations per encoded halftone pixel to unity. From an empirical analysis of the shape of the discriminant histogram for several inkjet and laserjet printers, Shaked *et al.* [24] propose the following asymmetric Laplacian probability model for the observations

$$P(y|0) = \begin{cases} \alpha_{0L} e^{-\alpha_{0L}(\mu_0 - y)} & y < \mu_0 \\ \alpha_{0R} e^{-\alpha_{0R}(\mu_0 - y)} & y > \mu_0 \end{cases} \quad (17)$$

$$P(y|1) = \begin{cases} \alpha_{1L} e^{-\alpha_{1L}(\mu_1 - y)} & y < \mu_0 \\ \alpha_{1R} e^{-\alpha_{1R}(\mu_1 - y)} & y > \mu_0. \end{cases} \quad (18)$$

For each individual image the parameters $\{\alpha_{0L}, \alpha_{0R}, \alpha_{1L}, \alpha_{1R}, \mu_0, \mu_1\}$ are different and are estimated using an expectation maximization approach [24]. Once the parameters are estimated, a likelihood score is computed for each possible message symbol. Thus, the decoded codeword at location \mathbf{m} is given by

$$\hat{\mathbf{c}}(\mathbf{m}) = \arg \max_{\{\mathbf{c} \in \mathcal{C}: \mathbf{w} = \mathbf{b}(\mathbf{m}) \otimes \mathbf{c}\}} \prod_{i=0}^{N^2-1} P(y_i | w_i) \quad (19)$$

where $\mathbf{b}(\mathbf{m})$ represents the known bi-level base-image block at location \mathbf{m} and \mathbf{c} represents a possible codeword block. Here, $w_i \in \{0, 1\}$ represents the encoded halftone value expected at location i if codeword \mathbf{c} was used to encode block $\mathbf{b}(\mathbf{m})$. The XOR operation is denoted by \otimes .

B. Encoding Information Into an IBC

1) *IBC Encoder*: Fig. 13 shows the system block diagram for the encoding of information into an IBC. We generate an IBC of a continuous-tone base image X (which may be a grayscale image having several graylevels) by jointly manipulating the base image X into a bi-level image B' , and an encoded image W (based on the message bits), such that application of a VSB on B' yields W . Constraints on B' may be enforced in the joint encoding process to allow B' to be estimated from W alone, using the methods described in Section V-C2. Once B' is determined from W , it plays the role of the base image in a regular

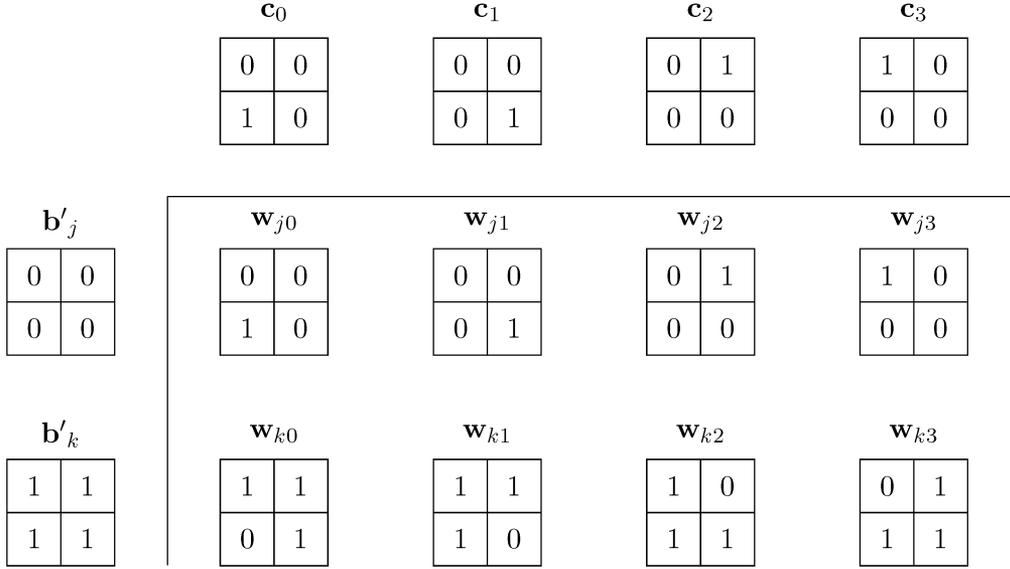


Fig. 12. Example XOR modulation encoding used in IBCs. The encoded block \mathbf{w} is the result of an XOR operation between the encoding codeword block \mathbf{c} and the bi-level intermediate half-tone block \mathbf{b}' . Thus, $\mathbf{w}_{lm} = \mathbf{c}_m \otimes \mathbf{b}'_l$. The case when the intermediate half-tone blocks are constrained to have either all black or all white pixels is shown. Such constraints are required for intermediate half-tone estimation at the decoder.

VSB, and the encoded IBC may be decoded as if it were a regular VSB.

- 1) The original image is divided into blocks. The blocks need not be rectangular, but must tile to cover the entire image. This is identical to the first step of VSB encoding [24].
- 2) The processing of the blocks usually proceeds in raster or serpentine scan order. At each block, a decision is made to represent that block with a block selected from finite number of possible blocks (composed of binary levels) of the same size. One example for possible allowed blocks is a block with all pixels equal to 1 (white) and another with all pixels equal to 0 (black). These blocks make up the intermediate half-tone image B' . The decision to allow a black block or a white block in the intermediate half-tone is made by simply thresholding the average modified input block value (modified by past errors).
- 3) The current message code word is modulated onto the binary block using XOR modulation, just as with a regular VSB. The modulated blocks make up the encoded image W . Fig. 12 shows codeword alphabet $C = \{[0\ 0\ 0\ 1]^T, [0\ 0\ 1\ 0]^T, [0\ 1\ 0\ 0]^T, [1\ 0\ 0\ 0]^T\}$ used to encode 2×2 intermediate half-tone blocks. The codewords are represented as row ordered vectors.
- 4) The quantization error between the modulated binary result and the current graylevel block is diffused to neighboring unprocessed graylevel blocks using a block-error filter with a diffusion matrix $D = (1/4)\mathbf{1}_{4 \times 4}$, which corresponds to diffusing the average block error to neighboring unprocessed pixel blocks.
- 5) The next graylevel block in the scan path is considered and steps 2)–5) are repeated until all image pixel blocks have been processed.

Consider the case when pixel blocks are quantized by allowing only solid black or solid white blocks in the intermediate half-tone image B' . This means that the resulting encoded

block obtained by modulating the message onto an intermediate half-tone pixel block, is always equal to the given codeword or its complement. Fig. 12 shows the possible encoded image blocks in this case. Since the codewords in a VSB are generated by toggling very few pixels (usually just one or two) of a uniform white or black block, the encoded pixel blocks of E are reproduced from B' with a change of just one or two pixels being corrupted due to the message. This yields overall robustness. The following equations describe the IBC encoding process using 2×2 block encoding. The quantizer $\mathbf{Q}(\cdot)$ converts the modified input block $\mathbf{u}(\mathbf{m})$ into the intermediate half-tone block $\mathbf{b}'(\mathbf{m})$

$$\mathbf{b}'(\mathbf{m}) = \mathbf{Q}(\mathbf{u}(\mathbf{m})) = \begin{cases} [1\ 1\ 1\ 1]^T, & \sum_{i=0}^3 u_i(\mathbf{m}) > \frac{1}{2} \\ [0\ 0\ 0\ 0]^T, & \text{else.} \end{cases} \quad (20)$$

The codeword $\mathbf{c}(\mathbf{m})$ is modulated onto the intermediate half-tone using XOR modulation

$$\mathbf{e}(\mathbf{m}) = \mathbf{c}(\mathbf{m}) \otimes \mathbf{b}'(\mathbf{m}). \quad (21)$$

The resulting quantization error block $\mathbf{e}(\mathbf{m})$ is diffused using a block-error filter $\tilde{\mathbf{h}}$ with matrix valued coefficients. We use the block-error filter with coefficients

$$\begin{aligned} \tilde{\mathbf{h}}(0, 1) &= \frac{7}{64}\mathbf{1}_{4 \times 4} \\ \tilde{\mathbf{h}}(1, 1) &= \frac{1}{64}\mathbf{1}_{4 \times 4} \\ \tilde{\mathbf{h}}(1, 0) &= \frac{5}{64}\mathbf{1}_{4 \times 4} \\ \tilde{\mathbf{h}}(1, -1) &= \frac{3}{64}\mathbf{1}_{4 \times 4}. \end{aligned}$$

This corresponds to diffusing the average error using the Floyd–Steinberg weights and distributing the error diffused to a pixel block equally to all elements within the pixel block.

2) *Stabilizing the IBC Encoder:* Since the encoding stage modifies the thresholding process in block-error diffusion based on the message signal, it is possible that the error might become

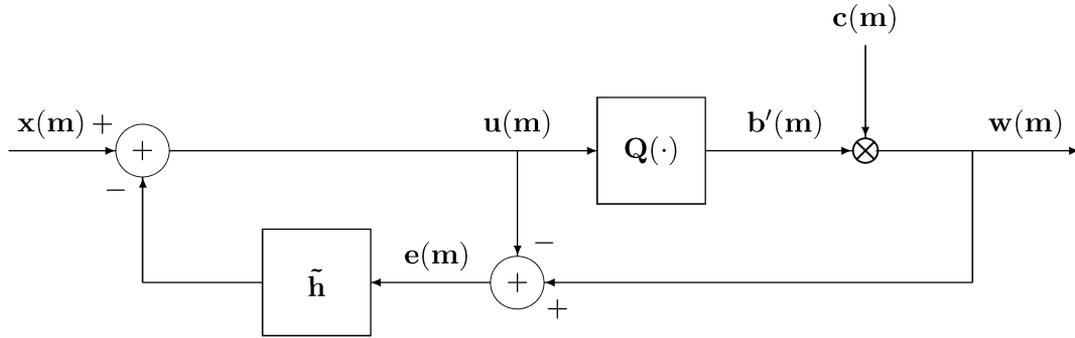


Fig. 13. System block diagram for the IBC encoder. $\tilde{\mathbf{h}}$ represents a fixed 2-D nonseparable FIR error filter with matrix valued coefficients. \mathbf{Q} denotes the block quantizer. The encoded image block $\mathbf{w}(\mathbf{m})$ is obtained by modulating the codeword $\mathbf{c}(\mathbf{m})$ onto the intermediate halftone block $\mathbf{b}'(\mathbf{m})$ using the XOR operation denoted by \otimes . The vector \mathbf{m} represents the 2-D index (m_1, m_2) .

unbounded resulting in unstable encoder behavior and degradation in image quality. For 2×2 blocks, with the IBC encoder using the codewords given in Fig. 12 the following stability results hold.

Stability Result 1: If every continuous-tone image block $\mathbf{x}(\mathbf{m})$ satisfies the condition $\sum_{i=0}^3 x_i(\mathbf{m}) \in [(1/4), (3/4)]$, then the encoder is stable.

Proof: The proof of this result follows by induction. If $\sum_{i=0}^3 x_i(\mathbf{m}) \in [(1/4), (3/4)]$, then the average quantization error for the first image block $\sum_{i=0}^3 e_i(\mathbf{m}) \in [-(1/4), (1/4)]$. This is true as the average of the output encoded image block lies in the set $\{(1/4), (3/4)\}$ irrespective of the particular codeword that is used. Since the error filter coefficients sum to unity the feedback signal $f(\mathbf{m} + \delta)$ at the next scan location lies in the range $[-(1/4), (1/4)]$. Since $\sum_{i=0}^3 x_i(\mathbf{m} + \delta) \in [(1/4), (3/4)]$, the modified input block at the next scan location $\mathbf{u}(\mathbf{m})$ satisfies $\sum_{i=0}^3 u_i(\mathbf{m}) \in [0, 1]$ which results in $\sum_{i=0}^3 e_i(\mathbf{m} + \delta) \in [-(1/4), (1/4)]$. Thus, the average error due to the encoding is bounded, and, hence, the encoder is stable.

Stability Result 2: If the fictitious codeword $\mathbf{c}_f = [0\ 0\ 0\ 0]^T$ is used whenever $\sum_{i=0}^3 x_i(\mathbf{m}) \notin [(1/4), (3/4)]$ then the encoder is stable for all continuous-tone inputs.

Proof: The proof of this result follows from the fact that the fictitious code word \mathbf{c}_f simply reproduces the intermediate halftone block as the output encoded halftone block. It is a do-nothing code that does not embed any information. The proof is similar to the one presented above, and follows by induction.

If $\sum_{i=0}^3 x_i(\mathbf{m}) \in [0, 1]$, then the average quantization error for the first image block $\sum_{i=0}^3 e_i(\mathbf{m}) \in [-(1/2), (1/2)]$. This is true as the average of the output encoded image block lies in the set $\{0, (1/4), (3/4), 1\}$ irrespective of the particular codeword that is used. Since the error filter coefficients sum to unity, the feedback signal $f(\mathbf{m} + \delta)$ at the next scan location lies in the range $[-(1/2), (1/2)]$. Further, $\sum_{i=0}^3 x_i(\mathbf{m} + \delta) \in [0, 1]$, the modified input block at the next scan location $\mathbf{u}(\mathbf{m})$ satisfies $\sum_{i=0}^3 u_i(\mathbf{m}) \in [-(3/2), (3/2)]$, which results in $\sum_{i=0}^3 e_i(\mathbf{m} + \delta) \in [-(1/2), (1/2)]$. Hence, the average error due to the encoding is bounded and the encoder is stable for all continuous tone input images when fictitious codewords are allowed.

Fictitious codewords are used at the encoder to ensure encoder stability. As we shall see in the next section, fictitious codewords may also be used to trade information capacity

for image quality. In either case, they must be handled by the decoder.

3) *Trading Information Capacity for Image Quality:* Ignoring fictitious codewords the information capacity of an IBC using the codeword set in Fig. 12 is 2 bits per block. Typically, half of this capacity is not actually used to embed message codewords due to error correction coding. This reduces the effective capacity to 1 bit per pixel block. If fictitious codewords are used, then the information capacity is image dependent and is given by

$$\text{Capacity (in bits)} = (T - F) \times \text{BPP} \times \text{ECC} \quad (22)$$

where T is the total number of image blocks, F is the total number of image blocks coded with fictitious codes, BPP is the bits encoded by the encoding alphabet per block, and ECC is the loss fraction due to error control coding. At each block location one has the choice of encoding information or not encoding information by using fictitious codewords. If a distortion metric is used to determine which blocks use fictitious codewords, then the image quality is enhanced at these locations since no information is embedded. Fig. 10(d) shows an example 216×198 IBC with 1.2 kB of embedded biography information. The actual rendering resolution is 150 dpi. In the following section, we discuss how the information embedded within an IBC may be recovered from a scanned version.

C. Decoding IBCs

As described in Section V-A, first the corners are determined using the detected fiducial marks. Then a global geometric transform is applied and local shape deformations are compensated for. At this stage the scanned barcode pixel blocks correspond to the encoded halftone pixels as described in Section V-A. The barcode is now ready for probabilistic decoding. Two different scenarios for decoding exist depending on whether (or not) the decoder has knowledge of the continuous-tone base image. Section V-C1 deals with the case when the continuous-tone base image is known at the decoder. Section V-C2 describes the situation when the decoder has no information regarding the continuous-tone base image and must decode the information by estimating the bi-level intermediate halftone image directly from the scanned data.

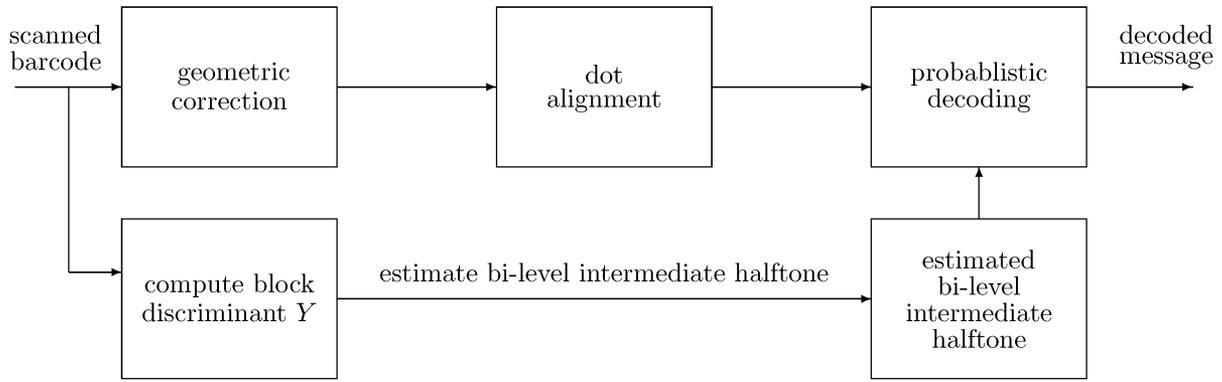


Fig. 14. Decoding pipeline for IBCs when the base image is not known. The bi-level intermediate halftone can be estimated in this case.

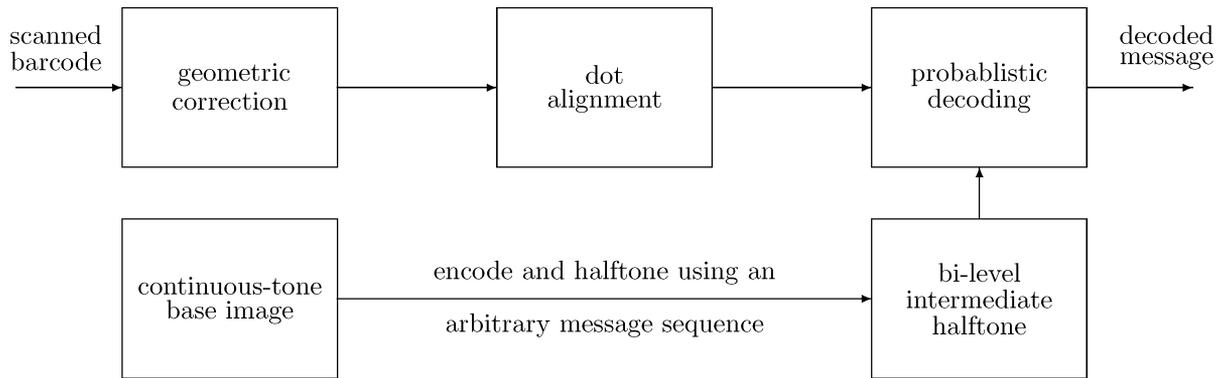


Fig. 15. Decoding pipeline for IBCs when the base image is known. The bi-level intermediate halftone can be determined exactly in this case.

1) *IBC Decoding When Base Image Is Known:* Fig. 15 outlines the pipeline used to implement base image-guided decoding of IBCs. To use the maximum likelihood decoding strategy of (19) as the probabilistic decoding scheme we need to estimate the bi-level intermediate halftone image on which the encoding codewords were modulated. From Fig. 13 we see that in general the intermediate halftone B' depends not only on the continuous-tone base image but on the message codewords as well, due to the feedback mechanism in the block-error diffusion. However, by proper choice of codewords and block-error filter we can ensure that the intermediate halftone in-fact depends only on the continuous-tone base image and not on a particular message codeword used. In fact if the block-error filter has a diffusion matrix $D = (1/4)\mathbf{1}_{4 \times 4}$ (which corresponds to diffusing the average block error to neighboring unprocessed pixel blocks), and the codewords all have the same average value as in Fig. 12, the diffused error block is independent of the particular codeword that is used. Thus, an arbitrary message could be used to generate the intermediate halftone B' from the known continuous-tone base image B . Note that the locations of the fictitious codewords are known *a priori* from the continuous-tone base image, and they can be incorporated in the exact determination of the intermediate bi-level intermediate halftone image. Once the bi-level intermediate halftone image is determined, (19) is used after the Laplacian probability model given by (17) and (18) is determined from the scanned data to decode the message codewords.

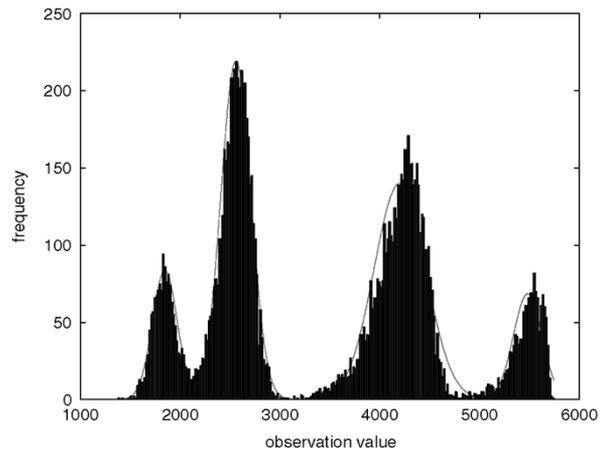


Fig. 16. Histogram of observed block discriminant Y and a Gaussian mixture model fit to the observed statistics using expectation maximization. The extreme modes correspond to fictitious codewords.

2) *Blind Decoding of IBCs:* Fig. 15 outlines the pipeline used to implement blind decoding of IBCs. The intermediate halftone must be estimated from the scanned IBC without knowledge of the continuous tone base image. This imposes natural restrictions on the number of allowable intermediate patterns in the intermediate halftone image. Fig. 8 shows that the XOR encoding using two different codewords on two different base-image blocks can produce the same output encoded image block. Thus, it is impossible to decide which codeword was used in encoding without knowledge of the intermediate halftone in this case. If, however, the intermediate halftone

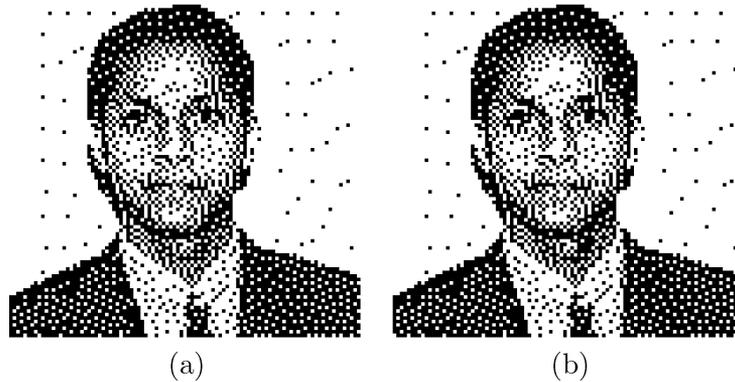


Fig. 17. Intermediate halftone estimation quality. (a) Actual intermediate halftone. (b) Estimated intermediate halftone.

were constrained to a restricted set of block patterns, then this decoding ambiguity is easily resolved. For example if we allow the intermediate halftone image blocks to be either all white or all black only as in Fig. 12 then the intermediate halftone block is clear by inspection of the corresponding encoded halftone image block. More specifically, the number of patterns allowed for an intermediate block of B' depends on a bound that permits blind decoding. For blind decoding, we must be able to infer the blocks of the intermediate halftone image B' directly from the encoded image blocks. This means that if there are L distinct code words and the block was composed of N pixels, then we must necessarily have

$$2^N \geq b'L \quad (23)$$

where there are b' distinct patterns in B' yielding $b'L$ output patterns. Thus, if we are dealing with 2×2 blocks with four codewords, $b' \leq 4$. Note that in theory, we can use more than one distinct codeword block to encode a single unique codeword to improve image quality in the encoding process. For example if we were using 2×2 blocks and encoding 2 bits/block, using four codeword templates with all black pixels except one, we can improve quality by adding four vertical/horizontal edge templates.

In practice the estimation of the bi-level intermediate halftone image must be performed from imperfect observations due to print-scan channel degradations and decoder preprocessing. After alignment there is a correspondence between a measurement patch and an encoded halftone pixel. A linear discriminant $y_i(\mathbf{m})$ is formed by multiplying the elements of the patch pointwise by a truncated Gaussian and summing. If 2×2 blocks were used in the encoding process, a new discriminant $Y(\mathbf{m})$ is formed by averaging the pixel-level discriminants $y_i(\mathbf{m})$. Thus

$$Y(\mathbf{m}) = \frac{\sum_{i=0}^3 y_i(\mathbf{m})}{4}. \quad (24)$$

Fig. 16 shows the histogram of a block-level discriminant Y . From the observation $Y(\mathbf{m})$ we need to determine the intermediate the intermediate halftone image block $\hat{\mathbf{b}}'(\mathbf{m})$. This may be done by deriving optimal thresholds to classify the codewords from the observations. This can be framed as a maximum-likelihood estimation problem. First, a Gaussian mixture model is fit to the observations using the expectation maximization para-

digm. The estimation of the intermediate halftone blocks is then reduced to

$$\hat{\mathbf{b}}'(\mathbf{m}) = \arg \max_{\mathbf{b}' \in \mathcal{B}} P(Y|\mathbf{b}') \quad (25)$$

where \mathcal{B} represents the set of allowed intermediate halftone patterns. The individual class conditional probability distributions $P(Y|\mathbf{b}')$ are Gaussians obtained from the Gaussian mixture model. After the intermediate halftone has been estimated, decoding proceeds by using (17) and (18) to determine the pixel-level Laplacian probability model and then finding the codeword used to encode a given intermediate halftone image block as

$$\hat{\mathbf{c}}(\mathbf{m}) = \arg \max_{\{\mathbf{c} \in \mathcal{C} \cup \mathcal{C}_f: \mathbf{w} = \hat{\mathbf{b}}'(\mathbf{m}) \otimes \mathbf{c}\}} \prod_{i=0}^3 P(y_i|w_i). \quad (26)$$

where \mathcal{C} and \mathcal{C}_f denote the set of information encoding and fictitious codewords, respectively. Note that the fictitious codewords must be explicitly estimated in the blind decoding case while this was not required when the decoder had knowledge of the continuous-tone base image. Fig. 17(a) shows the actual intermediate halftone image. Fig. 17(b) shows the estimated intermediate halftone image from a scanned version of the IBC.

3) *Decoder Performance*: We tested the performance of the IBC decoder using around 25 test images printed on three different InkJet and LaserJet printers. We scanned them in at 600 dpi. We found the decoder to be extremely robust. We were able to achieve 100% correct decoding for print resolutions up to 150 dpi using knowledge of the base image. The decoding algorithm is robust to minor image rotations while scanning and a variety of image degradations. For example, a single line drawn with a pen across the image or minor dust did not affect the decoding. This is attributed to the strong error correction codes inserted at the encoding stage. Robust blind decoding was achieved at resolutions up to 100 dpi.

VI. CONCLUSION

In this paper, we have introduced a general framework for producing FM and AM-FM halftones with user-controlled shape and size. Standard enhancement techniques used in conjunction with scalar error diffusion may be extended in a straightforward manner for use in the context of block-error diffusion [6], [25]–[28]. We have shown how information may be embedded into hardcopy images using dot shape modulation within the block-error diffusion framework. We have also

shown how this information may be recovered from a scan of an IBC with and without decoder knowledge of the base image. As suggested by an anonymous reviewer, the block-error diffusion framework may be suitable for high-addressability systems, e.g., 600×300 addressable spots with a minimum spot size of 300×300 . This is a topic of future research.

REFERENCES

- [1] R. Floyd and L. Steinberg, "An adaptive algorithm for spatial grayscale," in *Proc. Soc. Image Display*, vol. 17, 1976, pp. 75–77.
- [2] L. Velho and J. M. Gomez, "Digital halftoning with space filling curves," *Comput. Graph.*, vol. 25, pp. 81–90, Jul. 1991.
- [3] T. Scheermesser and O. Bryngdahl, "Control of texture in image halftoning," *J. Opt. Soc. Amer. A*, vol. 13, pp. 81–90, Aug. 1996.
- [4] D. L. Lau, G. R. Arce, and N. C. Gallagher, "Green-noise digital halftoning," *Proc. IEEE*, vol. 86, no. 12, pp. 2424–2442, Dec. 1998.
- [5] —, "Digital color halftoning with generalized error-diffusion and green-noise masks," in *IEEE Trans. Image Process.*, vol. 9, May 2000, pp. 923–935.
- [6] R. Levien, "Output dependent feedback in error diffusion halftoning," *IS&T Imag. Sci. Technol.*, vol. 1, pp. 115–118, May 1993.
- [7] Z. He and C. A. Bouman, "AM-FM halftoning: A method for digital halftoning through simultaneous modulation of dot size and dot density," in *Proc. SPIE Color Imaging: Device-Independent Color, Color Hardcopy, and Applications VII*, vol. 4663, Dec. 2001, pp. 322–334.
- [8] N. Damera-Venkata and B. L. Evans, "FM halftoning via block error diffusion," in *Proc. IEEE Conf. Image Processing*, Oct. 2001, pp. 1081–1084.
- [9] Z. Fan, "Dot-to-dot error diffusion," *J. Electron. Imag.*, vol. 2, pp. 62–66, Jan. 1993.
- [10] R. Eschbach, "Pixel-based error-diffusion algorithm for producing clustered halftone dots," *J. Electron. Imag.*, vol. 3, pp. 198–202, Apr. 1994.
- [11] N. Damera-Venkata and J. Yen, "Image barcodes," in *Proc. SPIE Color Imaging: Processing, Hardcopy, and Applications VIII*, vol. 5008, Jan. 2003, pp. 493–503.
- [12] K. Tanaka, Y. Nakamura, and K. Matsui, "Embedding secret information into a dithered multi-level image," in *Proc. IEEE Military Communications Conf.*, Sep. 1990, pp. 216–220.
- [13] Z. Baharav and D. Shaked, "Watermarking of dither halftoned images," HP Labs Tech. Rep., 1998.
- [14] —, "Watermarking of dithered halftone images," in *Proc. SPIE Symp. Electron. Imag.*, 1999.
- [15] D. Kacker and J. P. Allebach, "Joint halftoning and watermarking," *IEEE Trans. Signal Process.*, to be published.
- [16] K. T. Knox and S.-G. Wang, "Digital watermarks using stochastic screens," *Proc. SPIE*, to be published.
- [17] G. Sharma and S.-G. Wang, "Show-through watermarking of duplex printed documents," presented at the SPIE Symp. Electronic Imaging, Jan. 2004.
- [18] M. S. Fu and O. C. Au, "A multi-bit robust watermark for halftone images," presented at the IEEE Int. Conf. Multimedia and Expo, 2003.
- [19] M. Analoui and J. Allebach, "Model based halftoning using direct binary search," in *Proc. SPIE Human Vision, Visual Processing, and Digital Display III*, vol. 1666, Feb. 1992, pp. 109–121.
- [20] M. Fu and O. Au, "Data hiding watermarking for halftone images," *IEEE Trans. Image Process.*, no. 4, pp. 477–484, Apr. 2002.
- [21] N. Damera-Venkata and B. L. Evans, "Design and analysis of vector color error diffusion halftoning systems," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1552–1565, Oct. 2001.
- [22] J. Jarvis, C. Judice, and W. Ninke, "A survey of techniques for the display of continuous tone pictures on bilevel displays," *Comput. Graph. Image Process.*, vol. 5, pp. 13–40, May 1976.
- [23] T. D. Kite, B. L. Evans, and A. C. Bovik, "Modeling and quality assessment of halftoning by error diffusion," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 909–922, May 2000.
- [24] D. Shaked, A. Levy, Z. Baharav, and J. Yen, "A visually significant two dimensional barcode," Hewlett-Packard Lab. Tech. Rep.—HPLTR 2000-164, 2000.
- [25] K. Knox and R. Eschbach, "Threshold modulation in error diffusion," *J. Electron. Imag.*, vol. 2, pp. 185–192, Jul. 1993.
- [26] R. Eschbach and K. Knox, "Error-diffusion algorithm with edge enhancement," in *J. Opt. Soc. Amer. A*, vol. 8, Dec. 1991, pp. 1844–1850.
- [27] N. Damera-Venkata and B. L. Evans, "Parallel implementation of multifilters," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 6, Jun. 2000, pp. 3335–3338.
- [28] N. Damera-Venkata, "Analysis and design and of vector error diffusion systems for image halftoning," Ph.D. thesis, Dept. Elect. Comput. Eng., Univ. Texas, Austin, Austin, TX, Dec. 2000.



Niranjan Damera-Venkata (S'97–M'00) received the B.E. degree in electronics and communication engineering (ranking third at his university) from the University of Madras, Madras, India, in 1997, and the M.S. and Ph.D. degrees in electrical engineering from The University of Texas, Austin, in 1999 and 2000, respectively.

Since July 2000, he has been with the Imaging Technology Department, Hewlett-Packard Research Laboratories, Palo Alto, CA, where he is currently a Senior Research Scientist and Technical Lead for projector pipeline research. He contributed key image processing algorithms to HP "Wobulation" technology that enables high-quality resolution doubling for projection displays. His current research interests are in video processing for projector pipelines. He has published work in the areas of digital halftoning, document security, symbolic design and analysis tools, image and video quality assessment, and digital camera image processing.

Dr. Damera-Venkata is a member of Sigma Xi. He won a 1998–1999 Texas Telecommunications Engineering Consortium Graduate Fellowship from The University of Texas.



Jonathan Yen is a Senior Researcher at the Imaging Systems Laboratory, Hewlett Packard Laboratories, Palo Alto, CA.

His research interests include digital image processing and information embedding, and his current work includes automatic re-eye removal and data mining on camera metadata.



Vishal Monga (S'97–M'05) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology (IIT), Guwahati, in May 2001, and the M.S.E.E. and Ph.D. degrees from The University of Texas, Austin, in May 2003 and August 2005, respectively.

His research interests span signal and image processing, applied linear algebra, and information theory. He has researched problems in color imaging, particularly halftoning and device color correction, information embedding in multimedia, and perceptual media hashing.

Dr. Monga is a member of SPIE and IS&T. He received the IS&T Raymond Davis scholarship in 2004, a Texas Telecommunications Consortium (TxTec) Graduate Fellowship from The University of Texas for the year 2002 to 2003, and the President's Silver Medal in 2001 from IIT, Guwahati.



Brian L. Evans (SM'97) received the B.S.E.E.C.S. degree from the Rose-Hulman Institute of Technology, Terre Haute, IN, in 1987, and the M.S.E.E. and Ph.D.E.E. degrees from the Georgia Institute of Technology, Atlanta, in 1988 and 1993, respectively.

From 1993 to 1996, he was a Postdoctoral Researcher at the University of California, Berkeley. Since 1996, he has been with the Department of Electrical and Computer Engineering at The University of Texas, Austin, where he is currently an Associate Professor. In signal processing, his

research group is focused on the design and real-time software implementation of wireless OFDM base stations and ADSL transceivers for high-speed Internet access. His group developed the first time-domain equalizer training method that maximizes a measure of bit rate and is realizable in real-time in fixed-point software. In image processing, his group is focused on the design and real-time software implementation of high-quality halftoning for desktop printers and smart image acquisition for digital still cameras.

He is a member of the Design and Implementation of Signal Processing Systems Technical Committee of the IEEE Signal Processing Society. He is the recipient of a 1997 U.S. National Science Foundation CAREER Award.