HEWLETT PACKARD

# Entropy Constrained Halftoning Using Multipath Tree Coding

Ping Wah Wong
Computer Peripherals Laboratory
HPL-95-82
July, 1995

digital halftoning, tree coding, delayed decision, entropy constraint, lossless compression

We suggest an optimization based method for halftoning that involves looking ahead into the future before a decision for each binary output pixel is made. We first define a mixture distortion criterion that is a combination of a frequency weighted mean square error and a measure depending on the distances between minority pixels in the halftone. A tree coding approach with the *ML*-algorithm is used for minimizing the distortion criterion and generating a halftone. While this approach generates halftones of high quality, these halftones are not very amenable to lossless compression. We introduce an entropy constraint into the cost function of the tree coding algorithm, which optimally trades-off between image quality and compression performance in the output halftones.

# 1 Introduction

The objective of image halftoning is to produce a bi-level image from a given continuous tone image so that both the continuous tone image and its corresponding halftone appear similar when observed from a distance. Popular techniques for image halftoning include ordered dithering [1–4], error diffusion [5–7], and optimization based techniques [8–14]. In the optimization approach for halftoning, one selects an error metric that gives the distortion between the continuous tone and halftone images, and then finds a bi-level image that minimizes the distortion. For an image of size $M$ by $N$, the total number of possible halftone images equals $2^{MN}$. For typical values of $M$ and $N$, such a number is so large that makes exhaustive searching infeasible. As a result, many practical but suboptimal search techniques have been proposed to perform the minimization procedure. They include direct binary search [8], blocking with greedy bit flipping [12], blocking with branch and bound minimization [13], diffusion-reaction model [14], and one-dimensional Viterbi algorithm [10].

Block based minimization techniques suffer from a disadvantage that the procedure is greedy, *i.e.,* a locally optimum decision made at any particular pixel location does not in general guarantee that the overall solution is globally optimum. Potentially one can improve on image quality by looking ahead before a decision on a pixel is made, *i.e.,* incorporating a delay in the decision process. In [10, 15], a one-dimensional minimization based halftoning algorithm that uses the looking ahead idea has been proposed. Each scan line of the continuous tone image is processed independently of the others, where one dimensional minimization is performed using the Viterbi algorithm [16]. The resulting images, while near optimum in the one dimensional sense, contain artifacts arising from the independent processing of each line.

Tree coding is a class of encoding technique using delayed decisions [17], *i.e.,* keeping multiple paths along the tree generated by the possible future code streams, where the retained paths are chosen according to a distortion criterion. The optimality of tree coding for independent and identically distributed random sources has been reported [18, 19]. In practical applications, tree coding has been shown to be very powerful in speech coding [20], image coding [21], and chain coding [22, 23]. In this paper, we apply the tree coding approach with a two-dimensional error metric to develop an optimization based halftoning algorithm. While the looking ahead is only performed within a scan line, the distortion metric considers the bit patterns in a two-dimensional fashion. The tree coding based halftoning algorithm interprets the halftone output as a binary tree, and is similar to the Viterbi decoding algorithm [16] in that it looks a predetermined number of steps into the future before a decision is made at each pixel location. As a result, the usual disadvantage of greedy optimization can be alleviated and hence better halftone images can be produced.

The distortion measure is an extremely important building block in any optimization based halftoning algorithm, as it directly impacts the quality of the output halftones. The fre-

quency weighted mean square error, where the frequency weights typically has a low pass characteristic that resembles the human visual system response, is very popular in the literature because of its tractability and its intuitive meaning. It is well known that human observers prefer halftones where the black and white dots are spatially spread out as uniformly as possible[1], which is consistent with the observation that a good halftone should have a *blue noise* characteristic [6]. Consequently, the frequency weighted mean square error is not completely satisfactory because it does not address the question of the spatial distribution of dots (details in a later section). As a matter of fact, this distortion measure tends to ignore or deemphasize high frequency contents due to the low pass characteristics in the frequency weights. In this paper, we propose a mixture distortion criterion that is a combination of weighted mean square error and a measure based on the distances between minority pixels. This is used in conjunction with a well known tree coding algorithm, *viz.*, the $ML$-algorithm [17], to generate the halftones that are of very high quality.

While the tree coding algorithm with the new mixture distortion criterion together generate very high quality halftones, these bi-level images are not amenable to lossless compression using popular techniques such as Lempel-Ziv compression [24,25] and arithmetic coding [26, 27]. Experimental results in a later section show that one can typically achieve compression ratios in the neighborhood of 1.6 for the outputs of the tree coding halftoner using a standard JBIG coder [28] and standard test images. It is well known in rate distortion theory [29] that one can trade distortion with compression ratio. We take this approach in this paper, and incorporate an entropy constraint into a cost function with the distortion. Then we proceed to minimize the cost function using the tree coding algorithm.

In Section 2, we propose a new mixture distortion measure for comparing a continuous tone image with its corresponding halftone. Section 3 considers the concept of looking ahead in halftoning, and develops a halftoning algorithm using the famous $ML$-algorithm and the mixture distortion measure. Experimental results using this approach is also presented. Section 4 considers the compression performance of the halftones generated by the tree coding algorithm. An entropy constrained halftoning method using the tree coding approach is developed, and experimental results are shown. Section 5 summarizes the results in this paper.

# 2 Mixture Distortion Criterion

Let $x_{m,n}$ be a continuous tone image with pixel values in the range between 0 (black) and 1 (white), and let $b_{m,n} \in \{0,1\}$ be a halftone (bi-level) image. Given $x_{m,n}$, the optimization approach to halftoning finds $b_{m,n}$ that minimizes a distortion measure $E[d(x_{m,n}, b_{m,n})]$. Hence the distortion measure has a direct impact on the quality of the generated halftone.

[1] We are only considering the monochrome case in this paper. Similar remarks and algorithms developed in this paper can be extended to the case of color halftoning.

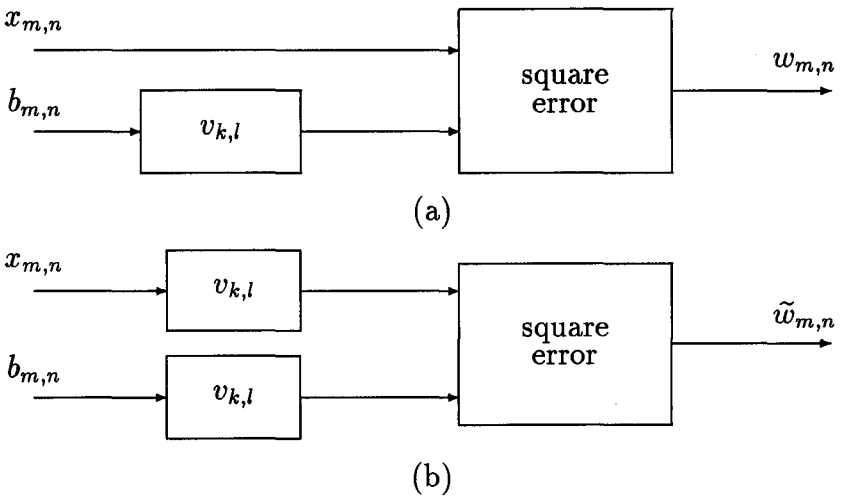


Fig. 1. Two different forms of frequency weighted mean square error.

From a human observer's point of view, an important feature of a good halftone image is that both $x_{m,n}$ and $b_{m,n}$ appear similar when viewed from a distance. A very popular measure for describing the distortion between $x_{m,n}$ and $b_{m,n}$ is the frequency weighted mean square error criterion $w_{m,n}$ [8,10,12,13]. Specifically, we let

$$w_{m,n} = (x_{m,n} - (v * b)_{m,n})^2 \tag{1}$$

where $v_{k,l}$ is an impulse response that approximates the characteristics of the human visual system, and $*$ denotes convolution. The operation of (1) can be represented by the block diagram in Fig. 1 (a). It makes good intuitive sense as it suggests that we measure the difference between an original continuous tone image and its corresponding halftone image as the halftone is perceived by the human visual system. Another form of frequency weighted mean square error that is also frequently used in the literature is

$$\tilde{w}_{m,n} = ((v * (x - b))_{m,n})^2, \tag{2}$$

which can be represented by Fig. 1 (b). In this form, both $x_{m,n}$ and $b_{m,n}$ are low pass filtered by $v_{k,l}$. Both (1) and (2) are used in the literature, and have been shown to produce good results in halftoning [8,10,12,13]. For the rest of this paper, we will use the form given in (1).

Digital halftoning, by its nature, relies on the spreading of black and white pixels to give a perception of gray levels. For high visual quality, one prefers the spatial distribution of black and white pixels to be as "uniform" as possible, since uniformly spaced dots generally gives visually smooth renditions of graylevels. This is consistent with designing halftones that has a *blue noise* (high frequency noise) characteristic [2–4,6], meaning that the energy in the error spectra between continuous tone and halftone images should preferably be concentrated at

(a) (b)

Fig. 2. Two possible halftone dot patterns for a constant gray patch at the graylevel $g = 0.75$.

the high frequency range. As an example, Fig. 2 (a) and (b) represent two possible halftone dot patterns for a constant gray patch at the graylevel $g = 0.75$. It is perhaps obvious that the pattern of Fig. 2 (a) is preferred over the one of Fig. 2 (b) for good subjective halftone quality. We therefore would like to use a distortion measure that favors halftone patterns with a blue noise characteristic.

It is evident from (1) that the spatial distribution of the black and white pixels is not explicitly reflected by the frequency weighted mean square error. To illustrate this, we show in Appendix A using a one-dimensional example that a halftone with a uniform (periodic) spatial distribution of black and white pixels can incur a larger frequency weighted mean square error than a more irregular binary sequence. It means that an optimization based halftoning algorithm that relies solely on the frequency weighted mean square error can lead one to obtain suboptimal results in the sense that the dot patterns in the output halftones may not be the most subjectively pleasing.

A different way to interpret this line of reasoning is the following: Since we want to generate a halftone to have a blue noise characteristic, we would like to be able to control the high frequency behavior of a halftone. Since the filter $v_{k,l}$ in (1) often exhibits a low pass characteristic, the weighted mean square error does not adequately reflect the high frequency characteristics of the halftones. To introduce some emphasis on the high frequency components, we propose to add an additional term—a dot distance based distortion—to the weighted mean square error, so that they form a mixture distortion measure.

To this end, we consider a measure based on the distances between the *minority* pixels [6] in a halftone, which has been experimentally verified [6, 30] to be a crucial factor to the quality of halftone images. If the gray scale of a local smooth region in an image is between 0 and 0.5, then the number of black pixels in the corresponding region of a halftone must be larger than the number of white pixels for the graylevel to be rendered correctly. In such case the white pixels are called minority pixels. Similarly, the black pixels are minority pixels when the local graylevel has a value between 0.5 and 1. Let

$$\rho_{m,n} = \begin{cases} 1 & \text{if } 0 \leq x_{m,n} < 0.5 \\ 0 & \text{if } 0.5 \leq x_{m,n} \leq 1 \end{cases}$$

be the value of the minority pixel at location $(m,n)$. Based on an approximation using square packing, one can define the *principal distance* $d_p$ [6] as the average distance between minority pixels in a halftone. Specifically,

$$d_p(g) = \begin{cases} \sqrt{1/g} & \text{if } 0 \leq g < 0.5 \\ \sqrt{1/(1-g)} & \text{if } 0.5 \leq g \leq 1, \end{cases}$$

where $g$ is the local gray level. Note that $d_p(g)$ is infinite for $g = 0$ or $g = 1$, as it should because no minority pixel should be inserted for complete black or white gray values. Let $d_{m,n}$ be the distance from the position $(m,n)$ to the nearest minority pixel. We can define a distortion measure using the distances between minority pixels by

$$u_{m,n} = \begin{cases} 0 & \text{if } d_{m,n} \geq d_p(x_{m,n}) \\ & \quad \text{and } b_{m,n} = \rho_{m,n} \\ 0 & \text{if } d_{m,n} < d_p(x_{m,n}) \\ & \quad \text{and } b_{m,n} \neq \rho_{m,n} \\ \left( \dfrac{d_p(x_{m,n}) - d_{m,n}}{d_p(x_{m,n})} \right)^2 & \text{otherwise.} \end{cases} \tag{3}$$

Note that $u_{m,n}$ favors putting a majority pixel at $(m,n)$ if the distance from the nearest minority pixel is less than $d_p(x_{m,n})$, while it favors a minority pixel at $(m,n)$ if the distance from the nearest minority pixel is larger than $d_p(x_{m,n})$.

Consider an example with $g = 0.75$. Hence we have $\rho = 0$, *i.e.,* the minority pixels are black pixels, and $d_p(g) = 2$. In the two cases shown in Fig. 3 (a) and (b), all the existing minority pixels in the halftone are more than a distance of 2 away from the location being considered. It would have be desirable if we could put a black pixel at some "past" locations so that the distance between minority pixels could be kept to $d_p(g)$. Since we cannot change the pixels that are already on the page, we would want to put a black pixel at the current location. Consequently we assign a penalty to case (b), and no penalty to case (a). On the other hand, the distance from $(m,n)$ to the nearest minority pixel is only $\sqrt{2}$ in the cases (c) and (d), which is smaller than the principal distance. In such a situation, we would want to put a white pixel at position $(m,n)$. Consequently, we put a penalty to case (c), and no penalty to case (d). The specific penalty as defined in (3) is given by the relative error between the principal distance and the actual distance to the nearest minority pixel. The criterion in (3), when incorporated into a distortion measure for an optimization based halftoning algorithm, explicitly encourages the minority pixels in a halftone to be located apart by the principal distance. As a result, this allows a smooth rendition of the continuous tone image, which leads to good subjective halftone quality. Note that a similar approach, that explicitly considers the distance between minority pixels, has also been introduced to error diffusion [30] to obtain good output quality.

Using the frequency weighted mean square error and the distance from the nearest minority

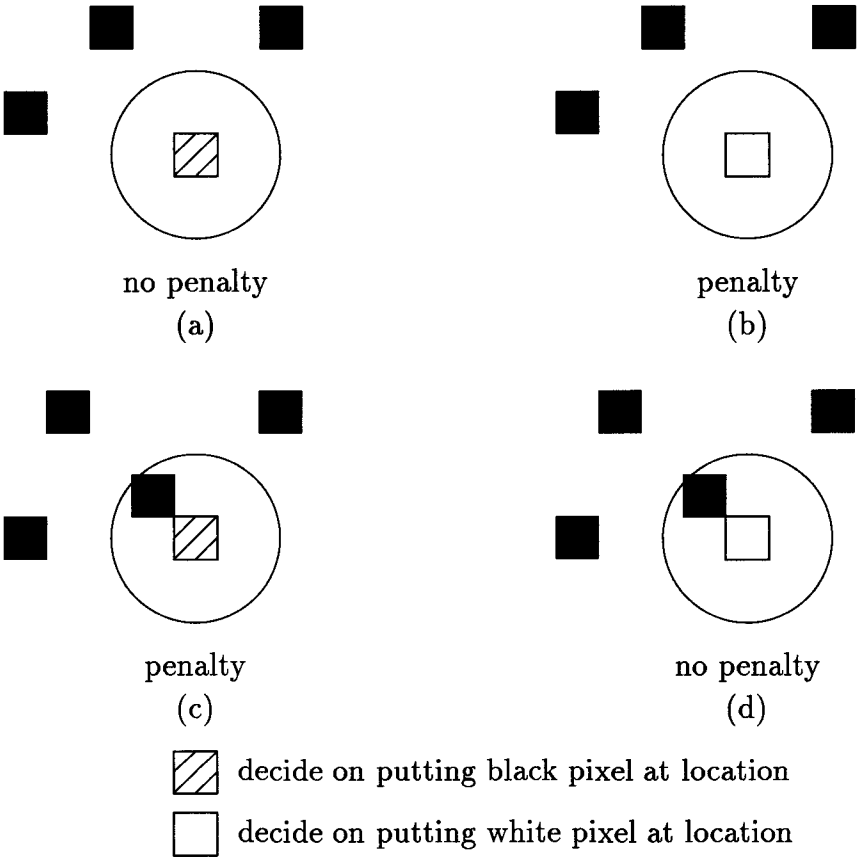


Fig. 3. Examples showing the four different situations in the dot distance based distortion measure of (3). In these examples, we have $g = 0.75$, $\rho = 0$ and $d_p(g) = 2$. The circle in each case is of radius 2, which equals the principal distance $d_p(g)$ at the graylevel used in this example.

pixel, we define a mixture distortion measure as

$$e_{m,n} = w_{m,n} + \gamma u_{m,n} \tag{4}$$

where $\gamma$ is a parameter that controls the weighting between $w_{m,n}$ and $u_{m,n}$. We will use this mixture distortion criterion in a a halftoning algorithm based on tree coding, and experimentally determine a good value of $\gamma$.

# 3 Tree Coding Halftoning Algorithm

In a typical minimization approach to halftoning, we can process $x_{m,n}$ in a raster scan fashion (*i.e.,* from the top to the bottom on a row by row basis, and for each row from the left to the right), and then choose $b_{m,n}$ sequentially at each location $(m,n)$ to be either 0 or 1 so that $e_{m,n}$ is minimized. If the processing of $x_{m,n}$ is to be performed according to a raster scan fashion, we can choose the filter $v_{m,n}$ to have a causal support so that $b_{m,n}$ can be generated sequentially. Similarly, the search area for the nearest minority pixels, that determines the value of $d_{m,n}$, is confined to the causal half plane.

Since the human visual system perceives halftone images through a form of local averaging over the binary pixels, the value of each pixel will affect the distortion at some "future" pixel locations with respect to the scanning strategy. As a result, the aforementioned approach to halftoning, although minimizes $e_{m,n}$ at each location, does not necessarily produce the best possible halftone image because of the greedy nature of the procedure. We can alleviate the drawbacks of greedy minimization by applying the tree coding approach [17], where a decision for each binary pixel $b_{m,n}$ is made after we have looked ahead a certain number of steps. The advantage of looking ahead in halftoning has also been observed in [10,15], where a one-dimensional approach to halftoning is proposed using the Viterbi decoding algorithm. Here we use a tree coding algorithm to perform the looking ahead in the scan direction, in conjunction with a two-dimensional distortion measure given in Section 2.

To apply the tree coding algorithm in halftoning, we first choose a fixed number $L$, representing the number of steps that we look ahead before we would make a binary decision for each output pixel. We then consider for each location $(m,n)$ all the possible binary output sequences for a depth of $L$ steps, *i.e.,* all possible bit streams $b_{m,n}, b_{m,n+1}, \ldots, b_{m,n+L}$. These $2^{L+1}$ possible bit streams can be put into a binary tree of depth $L+1$. An example for $L=2$ is shown in Fig. 4. Each branch at depth $k$ is identified with an output value of $b^{(i)}_{m,n+k}$ where $i$ identifies the particular path in the collection. Hence each leaf of the tree is associated with a particular output sequence $\{b^{(i)}_{m,n+k} : k = 0,1,\ldots,L\}$. We compute for each path a cumulative distortion defined by

$$D^{(i)}_{m,n} = \sum_{k=0}^{L} e^{(i)}_{m,n+k}. \tag{5}$$

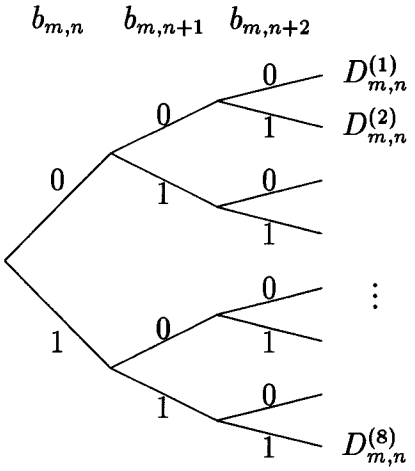


Fig. 4. A binary tree representing all the possible output bit patterns $b_{m,n}, b_{m,n+1}, \ldots, b_{m,n+L}$ for $L = 2$. Associated with each branch is a distortion $e_{m,n}$. The cumulative distortion of each path is calculated from (5). The superscript $i$ in $D^{(i)}_{m,n}$ identifies the path.

The output bit $b_{m,n}$ at location $(m, n)$ is then decided using a minimum distortion criterion. Specifically, we can use either one of the following strategies:

1. Choose the path with the smallest distortion, *i.e.*, let

$$i^* = \arg \min_i D^{(i)}_{m,n}$$

and then choose $b_{m,n}$ by

$$b_{m,n} = b^{(i^*)}_{m,n}. \tag{6}$$

2. Calculate for each possible output value (0 and 1) the average distortion

$$\mathcal{D}_{m,n,j} = \operatorname*{avg}_{\{i: b^{(i)}_{m,n} = j\}} D^{(i)}_{m,n} \qquad j = 0, 1. \tag{7}$$

That is, $\mathcal{D}_{m,n,0}$ is the average distortion of all the paths where the output pixel $b_{m,n}$ would be 0, and similarly $\mathcal{D}_{m,n,1}$ for the paths where $b_{m,n}$ would be 1. Then we decide on the output pixel $b_{m,n}$ by

$$b_{m,n} = \arg \min_j \mathcal{D}_{m,n,j}. \tag{8}$$

We have experimentally found that we can consistently obtain halftones of better quality using (8) than those obtained using (6).

Following the decision on $b_{m,n}$, all the paths where the bits $b^{(i)}_{m,n}$'s are not equal to $b_{m,n}$ are deleted. The bit values and cumulative distortions of the surviving paths are stored for

processing at the next step. The number of surviving paths for the tree coding algorithm with a depth of $L$ is $2^L$. We then move on to the next pixel location, extend all the surviving paths by one bit (hence doubling the number of paths), and calculate the cumulative distortion for each path. The next output bit is then decided similarly as before. This procedure is repeated until the entire image has been processed.

For any fixed $L$, the number of paths that need to be kept grows exponentially with $L$; an undesirable feature considering the memory and computational complexities. A popular approach in tree coding [17] is to choose a fixed integer $M$, and restricts the maximum number of paths that are kept at each stage to $M$. This is the well known $ML$-algorithm, which has been successfully applied in other coders [17,22,23]. In the rest of this paper, we will consider using the $ML$-algorithm for halftoning. Note that the special case $M = 2^L$ is identical to the generic tree coding algorithm where the full tree is used at every step. Another special case with $L = 0$ and $M = 1$, is equivalent to the greedy minimization approach where a decision on each binary pixel is made instantaneously.

The complexity of the $ML$-algorithm is proportional to $M$, since $2M$ distortion values (one for each extended path) must be computed at each step. The parameter $L$ actually has no effect on the complexity of the algorithm, as it only affects the set up stage of the algorithm. To see this, consider the procedures performed at a general location $(m, n)$, where we want to decide the pixel value $b_{m,n}$. A path $i$ at the current stage must be an extension of some surviving path, say the $j^{\text{th}}$, of the previous stage. We can then use (5) to write

$$D^{(i)}_{m,n} = \sum_{k=0}^{L-1} e^{(i)}_{m,n+k} + e^{(i)}_{m,n+L} = \sum_{k=0}^{L-1} e^{(j)}_{m,n+k} + e^{(i)}_{m,n+L} = D^{(j)}_{m,n-1} - e^{(j)}_{m,n-1} + e^{(i)}_{m,n+L}. \qquad (9)$$

But $e^{(j)}_{m,n-1}$'s for all surviving paths $j$'s are identical, since $b^{(j)}_{m,n-1} = b_{m,n-1}$ for all the surviving paths $j$'s. Keeping in mind that the path distortions are used only for comparative purposes, we can ignore $e^{(j)}_{m,n-1}$ from (9), and use $D^{(j)}_{m,n-1} + e^{(i)}_{m,n+L}$ in the decision procedure. This means that the complexity of the $ML$-algorithm is independent of $L$.

The multipath tree coding approach for halftoning using the $ML$-algorithm is summarized in Algorithm 1. The essential steps in the $ML$-algorithm is illustrated by an example in Fig. 5.

***Algorithm 1:*** Halftoning using the $ML$-algorithm.

1. Initialize $(m, n)$ to the first pixel.
2. Consider all the possible $2^{L+1}$ initial paths, and calculate $D^{(i)}_{m,n}$ for each path.
3. Calculate $\mathcal{D}_{m,n,j}$ from (7) for $j = 0$ and 1, and set the value of $b_{m,n}$ that results in the minimum average distortion according to (8).

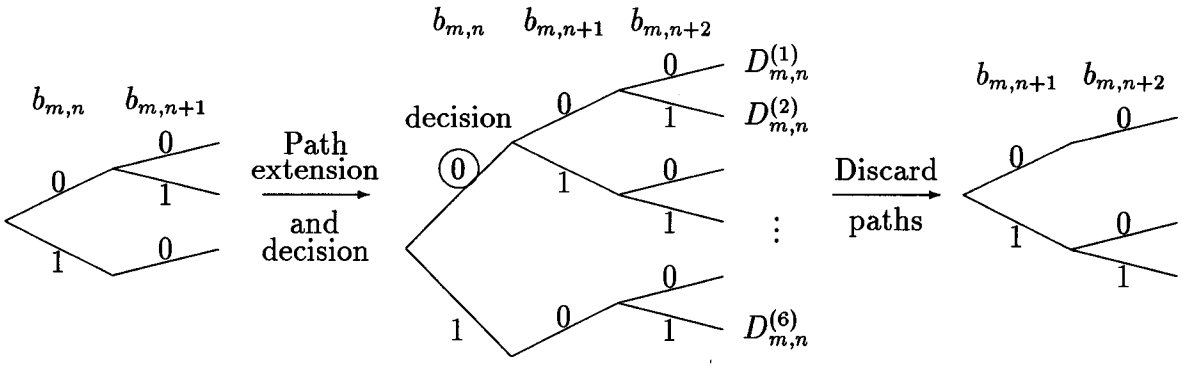


Fig. 5. An example illustrating the key steps in using the $ML$-algorithm for halftoning, with $M = 3$ and $L = 2$. In this example, the output at position $(m, n)$ is chosen to be 0, *i.e.*, $b_{m,n} = 0$.

4. Discard all the paths where the bit $b^{(i)}_{m,n}$ is not equal to the output bit $b_{m,n}$.
5. If less than $M$ paths remaining, save all. Otherwise, save the $M$ surviving paths with smallest cumulative distortions $D^{(i)}_{m,n}$.
6. Increment $(m, n)$ to the next position. If passed the end of the scan line, move on to the next scan line, reinitialize the tree, and go to step 3.
7. If passed the last pixel of the image, stop.
8. Extend all the saved paths by one bit (both 0 and 1), hence doubling the number of paths. Calculate $D^{(i)}_{m,n}$ for each path.
9. Go to step 3.

Here we consider the application of the $ML$-algorithm in conjunction with the mixture distortion of (4) for image halftoning. In particular, we consider a one-pass algorithm where the continuous tone image is scanned once in a raster scanned fashion while the halftone is generated. The approach can be easily extended to incorporate strategies that are often used in other popular halftoning techniques. For example, one can incorporate a serpentine scanning strategy in the $ML$-algorithm. One can also use multiple passes as in other optimization based halftoning methods [8, 12, 15]; That is, one applies one pass of the tree coding algorithm to generate an initial halftone, and then applies additional passes to refine the halftone for improved image quality.

To obtain $v_{m,n}$ with a causal support, we have chosen a truncated and normalized version of a filter used in [12], which is based on a measurement of the human visual contrast sensitivity

Fig. 6. Lena image of size 512 by 512 pixels halftoned using Floyd Steinberg error diffusion. The printing resolution is 150 dpi. The compression ratio achieved using a JBIG coder is 1.89.

function [31]. The filter coefficients are given by

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | $\boxed{0.2219}$ | 0.1439 | 0.0355 | 0.0116 |
| 0.0091 | 0.0306 | 0.0980 | 0.1439 | 0.0980 | 0.0306 | 0.0091 |
| 0.0030 | 0.0174 | 0.0306 | 0.0355 | 0.0306 | 0.0174 | 0.0030 |
| −0.0029 | 0.0030 | 0.0091 | 0.0116 | 0.0091 | 0.0030 | −0.0029 |

Note that the $(0,0)$ element 0.2219 in the impulse response is marked by a box.

Applying the tree coding algorithm with the distortion criterion given by (7), we have experimentally determined that the optimum value of $\gamma$ is 0.03. We use both the Lena and the pepper images of sizes 512 by 512 to demonstrate the experimental results. All the halftones are generated and printed at a resolution of 150 dots per inch. Fig. 6 shows a halftone of Lena generated using Floyd Steinberg error diffusion [5] with raster scanning. In order to make a comparison between the tree-coding halftoner with generic error diffusion, we have not incorporated techniques such as adding random noise or using serpentine scanning in the error diffusion examples. Fig. 7 shows the Lena image halftoned using the mixture distortion criterion in greedy optimization, *i.e.,* no looking ahead is performed in the optimization procedure for generating this image. This is equivalent to setting $M = 1$ and $L = 0$ in the $ML$-algorithm, and hence greedy optimization is a special case of tree coding. Comparing Figs. 6 and 7, we notice that the quality of the halftone generated by greedy

Fig. 7. Lena image of size 512 by 512 pixels halftoned using the mixture distortion criterion and greedy optimization (without looking ahead). The printing resolution is 150 dpi.

optimization is actually worse than that of error diffusion, confirming that greedy optimization is generally not a good approach. Fig. 8 shows the halftone of Lena generated using the $ML$-algorithm for $M = 8$ and $L = 5$. It demonstrates that the tree coding algorithm can generate halftones of higher quality than those generated by greedy optimization and error diffusion. Fig. 9 shows the pepper image halftoned using greedy optimization, while Fig. 10 shows the halftone of the pepper image generated with the $ML$-algorithm for $M = 4$ and $L = 4$. They show again that the quality of halftones can be significantly improved by using the idea of looking ahead through the multipath tree coding method. This is consistent with previous results on improved performance by applying tree coding to speech, image, line arts, *etc.* [10,17,20–23].

On testing with grayscale images of sizes 512 by 512 pixels in the USC database, we found that $M = 8$ is generally sufficient for good quality output, while the values of $L$ are typically no larger than 6. This means that the $ML$-algorithm, where only a partial tree is kept at every step, can generate halftones that are similar in quality to the generic tree coding algorithm where the full tree is kept. Recall that the complexity of the $ML$-algorithm only depends on $M$. In our implementation using C code on an HP 735 workstation, it takes about 68 seconds to generate a halftone of size 512 by 512, using the value $M = 8$. Error diffusion for images of the same size, on the other hand, requires only about 0.5 seconds on the same machine. Although error diffusion is more computationally efficient, the tree coding halftoner can offer much better image quality. Furthermore, as we shall see in the

Fig. 8. Lena image of size 512 by 512 pixels halftoned using the mixture distortion criterion in the $ML$-algorithm with $M = 8$ and $L = 5$. The printing resolution is 150 dpi. The compression ratio achieved using a JBIG coder is 1.52.

Fig. 9. Pepper image of size 512 by 512 pixels halftoned using the mixture distortion criterion in greedy optimization. The printing resolution is 150 dpi.

Fig. 10. Pepper image of size 512 by 512 pixels halftoned using the mixture distortion criterion in the $ML$-algorithm with $M = 4$ and $L = 4$. The printing resolution is 150 dpi. The compression ratio achieved using a JBIG coder is 1.71.

next section, the minimization approach also offers a natural mechanism, through adding an entropy constraint in the cost function, for generating halftones that trades-off between image quality and compression performance.

# 4 Entropy Constraint in Tree Coding

We have described in the previous section the application of the $ML$-algorithm for minimizing a mixture distortion and producing optimized halftones, such as the ones shown in Figs. 8 and 10. Using a standard JBIG (lossless) coder [28], it is found that one can achieve compression ratios of 1.52 for the Lena image of Fig. 8, and 1.71 for the pepper image of Fig. 10. Experimental results using other images give similar compression performance. As a comparison, we can achieve a compression ratio of 1.89 using JBIG on the error diffused image of Fig. 6.

Although the tree coding algorithm is capable of generating high quality halftones, it is desirable to improve the compression performance on the resulting bi-level images. This is useful, for example, in printing applications for reducing both the transmission time and memory requirement, or in communications applications where halftones are to be transmitted over bandwidth constrained channels. The fact that the halftones generated by the tree coding algorithm are not amenable to lossless compression is not surprising, because the

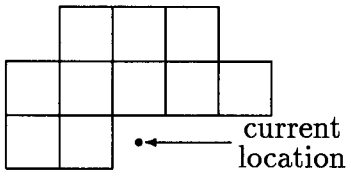


Fig. 11. Template of previous output pixels used as a context for computing conditional entropy.

optimization procedure in Section 3 has been performed to solely minimize distortion. Note that the tree coding based halftoner can be viewed as a lossy encoder for the continuous tone image, where the output alphabet is constrained to be binary. As indicated by Shannon's rate-distortion theory [29], one can trade distortion for compression performance in any lossy coder. To this end, one formulates the optimization problem by adding an entropy constraint to the cost function. More specifically, we can minimize $J = D + \lambda H_b$ where $D$ is the mixture distortion as defined in Section 2, $H_b$ is the *entropy rate* of the halftone $b_{m,n}$, and $\lambda$ is a Lagrangian parameter. The parameter $\lambda$ determines the location of the resulting halftone on the rate distortion curve, where $-\lambda$ has an interpretation of the gradient of the rate distortion function. We can then proceed to choose $b_{m,n}$ for minimizing the cost function using the $ML$-algorithm.

In order to incorporate the entropy constraint into the cost function, we need a way to estimate the statistics of the halftone. Since the $ML$-algorithm works on the image in a raster scan fashion, it is convenient to use the "past" halftoned data as a basis for estimating the probability distribution, and continuously update the statistics as one performs the halftoning. An efficient way to accomplish this is the usage of a conditional probability, *i.e.*, the probability of the current binary pixel value based on a window of "past" halftoned data. To this end, we use the concept of *context* introduced in [26,27].

Let $c_{m,n}$ be a context defined by a window of neighboring pixels at position $(m,n)$. Typically, we choose the neighbors to be the "previous" pixels according to the scanning strategy in the processing. To be specific in generating experimental results in this paper, we have chosen a context using a template of ten "previous" output pixels as shown in Fig. 11, and hence there are 1024 different possible context values. Note that this template is identical to the one used in a standard JBIG encoder. Let $h_{m,n}(c_{m,n}, b_{m,n})$ be the conditional entropy of $b_{m,n}$ conditioned on the context $c_{m,n}$. It is well known [32] that if $b_{m,n}$ is stationary and ergodic, then

$$\operatorname*{avg}_{m,n} h_{m,n}(c_{m,n}, b_{m,n}) \longrightarrow H_b$$

as the image size goes to infinity. Consequently, we can equivalently minimize the cost function that is defined using the conditional entropy, *viz.*,

$$J_{m,n} = e_{m,n} + \lambda h_{m,n}(c_{m,n}, b_{m,n}).$$

Although it is questionable that images of natural scenes are stationary, we still use the

aforementioned cost function because of the ease of implementation.

Since each output pixel $b_{m,n}$ is binary valued, the conditional entropy is completely determined by

$$p_{m,n}(0,c) = \Pr\{b_{m,n} = 0 | c_{m,n} = c\} = 1 - p_{m,n}(1,c).$$

Similar to what is done in adaptive arithmetic coding [26], these probabilities are estimated using the statistics of the past output pixels, and are continuously updated as the halftone image is being generated. An estimate of the conditional probability can be obtained as

$$\widehat{p}_{m,n}(b,c) = \frac{N_{m,n}(b,c)+1}{N_{m,n}(c)+2} \qquad b = 0,1 \tag{10}$$

where $N_{m,n}(c)$ is the frequency count of the context $c$ up to the location $(m,n)$, *i.e.,* the number of times that the context $c$ has occurred; and $N_{m,n}(b,c)$ is the frequency count of the context $c$ and output $b$, *i.e.,* the number of times that the output value $b$ follows context $c$. It is evident that

$$N_{m,n}(c) = N_{m,n}(0,c) + N_{m,n}(1,c).$$

Note that when a particular context value $c$ occurs for the first time, there is no prior data and hence $N_{m,n}(0,c) = N_{m,n}(c,1) = 0$. The bias values of "1" and "2" in (10) have been inserted to avoid the situation of dividing by zero. They also have the property that when there is no data, we have $\widehat{p}_{m,n}(0,c) = \widehat{p}_{m,n}(1,c) = 0.5$, *i.e.,* neither 0 nor 1 is favored for the first occurrence of $c$. Using these estimates (that are being constantly updated when the image is being halftoned), we use at each pixel location the cost function

$$\widetilde{J}_{m,n} = e_{m,n} - \lambda \log(p_{m,n}(b_{m,n}, c_{m,n})),$$

where it is well known that the term $-\log(p_{m,n}(b_{m,n},c))$ approximates the average length of a codeword required to describe $b_{m,n}$. As such, we are performing the minimization with respect to the *operational* rate distortion function, and $-\lambda$ now has the interpretation of the gradient of the convex hull supporting the operational rate distortion function.

Recall that we can only obtain a compression ratio of 1.52 using a standard JBIG coder on the halftone of Lena (Fig. 8) generated by the tree coding halftoner without using an entropy constraint. The error diffused version of Lena (Fig. 6), while of lower image quality than that of Fig. 8, can be compressed to a compression ratio of 1.89. The two images of Figs. 12 and 13 are the halftones generated by the entropy constrained tree coding halftoner at two different values of $\lambda$, and hence are of different entropy. Using a standard JBIG encoder, one can achieve a compression ratio of 2.09 for the image of Fig. 12, and 2.35 for the one of Fig. 13. As expected, the image quality of Fig. 13 is lower than that of Fig. 12, which in turn is lower than that of Fig. 8. The compression performance, on the other hand, ranks in the reverse direction. In other words, as the compression ratio improves, the distortion also increases as predicted by rate distortion theory [29]. Fig. 14 is a plot of the mixture

Fig. 12. Halftone of Lena generated using the entropy constrained tree coding halftoner. The compression ratio achieved using a JBIG coder is 2.09. The printing resolution is 150 dpi.

Fig. 13. Halftone of Lena generated using the entropy constrained tree coding halftoner. The compression ratio achieved using a JBIG coder is 2.35. The printing resolution is 150 dpi.

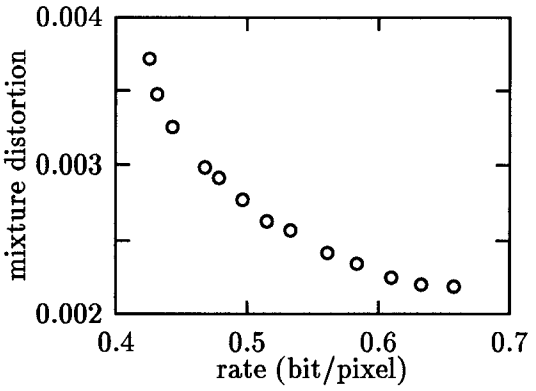


Fig. 14. Distortion-rate performance of the entropy constrained tree coding halftoner.

distortion versus the average bit rate required for a JBIG encoder to encode the output generated by the entropy constrained tree coding halftoner. The curve exhibits the usual form as predicted by rate distortion theory.

Comparing Figs. 8 and 12, we judge visually that the quality difference is relatively small, but the compression ratios between them differ by more than 25%. It is interesting to note that the halftone of Fig. 12 is *both* of better quality and more amenable to compression than the error diffused halftone of Fig. 6, showing the advantage of optimization based halftoning approach.

# 5 Conclusion

We have suggested a method using multipath tree coding for generating halftone images. The technique is a special case of dynamic programming, which looks ahead into the future along the paths of possible bit patterns and then makes a decision based on an averaged cumulative distortion. Specifically, the well known $ML$-algorithm is applied to perform the multipath minimization. For used in the optimization procedure, we have suggested a mixture distortion criterion that is a combination of frequency weighted mean square error and a measure based on distances between minority pixels. We have shown that halftones generated using the mixture distortion criterion is of better quality than those generated using the frequency weighted mean square error. Furthermore, halftones generated using the tree coding technique are observed to be of better quality than those obtained using either error diffusion or greedy optimization.

Although the tree coding algorithm generates halftones of high quality, the resulting halftones are not amenable to lossless compression. In view of this, we have incorporated an entropy constraint into the cost function, and use the tree coding algorithm to generate halftones that optimally trade quality with compression performance. The trade-off in the minimization is specified by a Lagrangian parameter, and hence the compression performance of the output

halftone can be controlled. We have shown that we can generate halftones that have better quality than generic error diffusion, and at the same time more amenable to compression than error diffused images.


## Acknowledgment

The author gratefully acknowledges the help of Dr. Gregory Yovanof, who supplied an experimental JBIG coder for testing the compression performance of the halftones.


# Appendix

## A An Example Concerning Frequency Weighted Mean Square Error and Uniformity of Spatial Distribution of Pixels in A Halftone

Consider a one-dimensional constant gray "image" of length 64

$$x_n = 0.25 \qquad n = 0, 1, \ldots, 63.$$

Two possible halftone representations for $x_n$ are

$$b_n = \begin{cases} 1 & \text{if } n = 4k \\ 0 & \text{otherwise} \end{cases} \qquad 0 \le n < 64$$

and

$$c_n = \begin{cases} 1 & \text{if } n = 8k \text{ or } n = 8k+3 \\ 0 & \text{otherwise} \end{cases} \qquad 0 \le n < 64.$$

Note that the ratios of black pixels to white pixels for both $b_n$ and $c_n$ are 3:1, giving an average intensity of 0.25. It is evident that the spatial distribution of black and white pixels is more "uniform" in $b_n$ than in $c_n$. As a result, $b_n$ would appear to be visually smoother than $c_n$, and hence $b_n$ is preferred over $c_n$. Consider the frequency weighted mean square error between the halftone patterns and the dc signal $x_n$ using a filter with a frequency response (64-point DFT of an impulse response $h_n$)

$$H_m = \begin{cases} 1 & \text{if } m = 0, 8, 16, 48, 56 \\ 0 & \text{if } m = 32 \\ \epsilon & \text{if } m = 24, 40 \\ \text{arbitrary} & \text{otherwise} \end{cases} \qquad 0 \le m < 64 \qquad \text{(A1)}$$

where $\epsilon$ is a parameter to be decided soon. Note that $m = 32$ corresponds to one half of the sampling frequency, and that we have only specified in this example the response of $H_m$ for

several values of $m$. It is evident that the specified $H_m$ is consistent with the characteristics of low pass filters. It can be verified that the DFT's of $b_n$ and $c_n$ are

$$B_m = \begin{cases} 16 & \text{if } m = 0, 16, 32, 48 \\ 0 & \text{otherwise} \end{cases} \qquad 0 \le m < 64$$

and

$$C_m = \begin{cases} 16 & m = 0 \\ 2.3431 - j5.6569 & m = 8 \\ 8 + j8 & m = 16 \\ 13.6569 - j5.6569 & m = 24 \\ 13.6569 + j5.6569 & m = 40 \\ 8 - j8 & m = 48 \\ 2.3431 - j5.6569 & m = 56 \\ 0 & \text{otherwise} \end{cases} \qquad 0 \le m < 64.$$

Here $j = \sqrt{-1}$. The frequency weighted mean square errors are

$$d(x_n, b_n) = |B_{16}|^2 + |B_{48}|^2 = 512$$

and

$$d(x_n, c_n) = |B_8|^2 + |B_{16}|^2 + |B_{48}|^2 + |B_{56}|^2 + \epsilon^2(|B_{24}|^2 + |B_{40}|^2) = 331 + \epsilon^2 437.$$

Hence if $\epsilon < 0.6435$, then $d(x_n, c_n) < d(x_n, b_n)$. That is, although $b_n$ is visually preferred over $c_n$ as a halftone, $c_n$ incurs a smaller frequency weighted mean square error than $b_n$. Note that the sequences $b_n$ and $c_n$, as well as the responses specified in (A1) only serve as a convenient example. There are many other situations that can lead to the same conclusion, *i.e.,* that the frequency weighted mean square error does not generally reflect the uniformity in the distribution of black and white dots in a halftone.

# References


[1] B. E. Bayer, "An optimum method for two-level rendition of continuous-tone pictures," in *Proceedings of IEEE International Conference in Communications*, pp. 26.11–26.15, 1973.

[2] T. Mitsa and K. J. Parker, "Digital halftoning technique using a blue-noise mask," *Journal of Optical Society of America A*, vol. 9, pp. 1920–1929, 1992.

[3] R. Ulichney, "The void-and-cluster method for dither array generation," in *Proceedings of SPIE, vol. 1913*, pp. 332–343, February 1993.

[4] Q. Lin, "Improving halftone uniformity and tonal response," in *Proceedings of IS&T International Congress on Advances in Non-impact Printing Technologies*, (New Orleans LA), pp. 377–380, November 1994.

[5] R. Floyd and L. Steinberg, "An adaptive algorithm for spatial grey scale," in *SID International Symposium, Digest of Technical Papers*, pp. 36–37, 1975.

[6] R. A. Ulichney, "Dithering with blue noise," *Proceedings of the IEEE*, vol. 76, pp. 56–79, January 1988.

[7] P. W. Wong, "Adaptive error diffusion and its application in multiresolution rendering." To appear in *IEEE Transactions on Image Processing.*

[8] M. Analoui and J. Allebach, "Model-based halftoning using direct binary search," in *Proceedings of SPIE, vol. 1666*, (San Jose CA), pp. 96–108, February 1992.

[9] S. Kollias and D. Anastassiou, "A progressive scheme for digital image halftoning, coding of halftones, and reconstruction," *IEEE Journal on Selected Areas in Communications*, vol. 10, pp. 944–951, June 1992.

[10] D. L. Neuhoff, T. N. Pappas, and N. Seshadri, "One-dimensional least-squares model-based halftoning," in *Proceedings of ICASSP*, (San Francisco, CA), pp. III 189–192, March 1992.

[11] K. R. Crounse, T. Roska, and L. O. Chua, "Image halftoning with cellular neural networks," *IEEE Transactions on Circuits and Systems*, vol. 40, pp. 267–283, April 1993.

[12] R. A. Vander Kam, P. A. Chou, and R. M. Gray, "Combined halftoning and entropy-constrained vector quantization," in *SID Digest of Technical Papers*, (Seattle, WA), pp. 223–226, May 1993.

[13] A. Zakhor, S. Lin, and F. Eskafi, "A new class of B/W halftoning algorithms," *IEEE Transactions on Image Processing*, vol. 2, pp. 499–509, October 1993.

[14] A. Sherstinsky and R. W. Picard, "M-lattice: A novel non-linear dynamical system and its application to halftoning," in *Proceedings of ICASSP*, (Adelaide, Australia), pp. (II–565)–(II–568), April 1994.

[15] T. N. Pappas, C. K. Dong, and D. L. Neuhoff, "Measurement of printer parameters for model-based halftoning," *Journal of Electronic Imaging*, vol. 2, pp. 193–204, July 1993.

[16] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269, April 1967.

[17] J. B. Anderson and S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Transactions on Communications*, vol. COM-32, pp. 169–176, February 1984.

[18] F. Jelinek, "Tree encoding of memoryless time-discrete sources with a fidelity criterion," *IEEE Transactions on Information Theory*, vol. 15, pp. 584–590, September 1969.

[19] C. R. Davis and M. E. Hellman, "On tree coding with a fidelity criterion," *IEEE Transactions on Information Theory*, vol. IT-21, pp. 373–378, July 1975.

[20] J. B. Anderson and J. B. Bodie, "Tree encoding of speech," *IEEE Transactions on Information Theory*, vol. IT-21, pp. 379–387, July 1975.

[21] J. W. Modestino, V. Bhaskaran, and J. B. Anderson, "Tree encoding of images in the presence of channel errors," *IEEE Transactions on Information Theory*, vol. IT-27, pp. 677–697, November 1981.

[22] M. Vembar and S. Mohan, "Tree encoding of line drawings," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-36, pp. 1542–1549, September 1988.

[23] R. Sriraman, J. Koplowitz, and S. Mohan, "Tree searched chain coding for subpixel reconstruction of planar curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-11, pp. 95–104, January 1989.

[24] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. IT-23, pp. 337–343, May 1977.

[25] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory*, vol. IT-24, pp. 530–536, September 1978.

[26] G. G. Langdon, Jr. and J. Rissanen, "Compression of black-white images with arithmetic coding," *IEEE Transactions on Communications*, vol. 29, pp. 858–867, June 1981.

[27] J. Rissanen and G. G. Langdon, Jr., "Universal modeling and coding," *IEEE Transactions on Information Theory*, vol. 27, pp. 12–23, January 1981.

[28] ISO/IEC International Standard 11544, "Coded representation of bi-level and limited-bits-per-pixel grayscale and color images," February 1993.

[29] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in *IRE National Convention Record, Part 4*, pp. 142–163, March 1959.

[30] R. Levien, "Output dependent feedback in error diffusion halftoning," in *Proceedings of IS&T's* $46^{\text{th}}$ *Annual Conference*, (Cambridge, MA), pp. 115–118, May 1993.

[31] J. G. Robson, "Spatial and temporal contrast sensitivity functions of the visual system," *Journal of Optical Society of America*, vol. 56, pp. 1141–1142, 1966.

[32] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, 1948.