High-speed first- and second-order frequency modulated halftoning

Sasan Gooran and Björn Kruse

Linköping University Post Print

Tweet

N.B.: When citing this work, cite the original article.

Original Publication:

Sasan Gooran and Björn Kruse, High-speed first- and second-order frequency modulated halftoning, 2015, Journal of Electronic Imaging (JEI), (24), 2.
http://dx.doi.org/10.1117/1.JEI.24.2.023016
Copyright: Society of Photo-optical Instrumentation Engineers (SPIE)
http://spie.org/

Postprint available at: Linköping University Electronic Press
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-115791

Journal of
Electronic Imaging

JElectronicImaging.org

High-speed first- and second-order frequency modulated halftoning

Sasan Gooran
Björn Kruse

SPIE.
IS&T
imaging.org

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

Journal of Electronic Imaging 24(2), 023016 (Mar/Apr 2015)

High-speed first- and second-order frequency modulated halftoning

Sasan Gooran* **and Björn Kruse**
Linköping University, Department of Science and Technology, Norra Grytsgatan 10A, Norrköping 601 74, Sweden

Abstract. Halftoning is a crucial part of image reproduction in print. First-order frequency modulated (FM) halftones, in which the single dots are stochastically distributed, are widely used in printing technologies, such as inkjet, that are able to stably print isolated dispersed dots. Printers, such as laser printers, that utilize electrophotographic technology are not able to stably print the isolated dots and, therefore, use clustered-dot halftones. Periodic clustered-dot, i.e., amplitude modulated halftones are commonly used in this type of printer, but they suffer from an undesired periodic interference pattern called moiré. An alternative solution is to use second-order FM halftones in which the clustered dots are stochastically distributed. The iterative halftoning techniques that usually result in well-formed halftones operate on the whole input image and require extensive computations and thus, are very slow when the input image is large. We introduce a method to generate image-independent threshold matrices for first- and second-order FM halftoning. The first-order threshold matrix generates well-formed halftone patterns and the second-order FM threshold matrix can be adjusted to produce clustered dots of different sizes, shapes, and alignment. Using predetermined and image-independent threshold matrices makes the proposed halftoning method a point-by-point process and thereby very fast. *© 2015 SPIE and IS&T* [DOI: 10.1117/1.JEI.24.2.023016]

Keywords: halftoning; first-order frequency modulated; second-order frequency modulated; screening; clustered dot; dispersed dot.

Paper 14734 received Nov. 19, 2014; accepted for publication Mar. 3, 2015; published online Mar. 19, 2015.

1 Introduction

Many reproduction devices, e.g., printers, have a limited number of output states, leaving the choice of printed and nonprinted spots in order to reproduce a shade. Thus, continuous tone gray-scale or color images need to go through a process called halftoning before being printed. Because of the fact that the human eye is limited in its capacity to resolve small dots and dots close to each other, if the viewing distance is long or the dots are small enough, the human eye is not able to distinguish between the original image and the halftone one. Hence, since the human eye acts as a low-pass filter, the halftones appear pleasing if the difference between the original and the halftone is small in the low-frequency region.

Halftoning algorithms are commonly categorized into two main subgroups, called amplitude modulated (AM) and frequency modulated (FM). In AM, i.e., periodic clustered-dot halftones, different shades of gray are reproduced by changing the size of the dots while keeping their spacing constant. In first-order FM, dispersed-dot halftones, on the other hand, the size of the dots is constant while their density (or frequency) is variable. There is also another type of halftones, which we call second-order FM in this paper, in which both the size and the frequency are variables. In these halftones, the clustered dots are stochastically distributed. In literature, this type of halftones is also referred to as stochastic clustered-dot halftones and even green-noise dither patterns. In Ref. 1, the radially averaged power spectrum (RAPS) curves for these three types of halftones, i.e., AM, first-order and second-order FM, are illustrated, which helps to study their spectral characteristics in different frequency ranges. The well-formed first-order FM halftones usually have the blue-noise characteristic meaning that the quantization noise produced by the halftoning process is shifted to a higher frequency where the human eye is less sensitive.[2,3] The choice of the appropriate halftone is, however, not always based on their frequency characteristic, but sometimes based on the properties and limitations of the printing devices. For example, inkjet printers are able to stably print dispersed isolated dots while printers using electrophotographic (EP) technology cannot stably print isolated dots. Today, EP technology is used in xerographic reproduction devices such as laser printers. Clustered dots are, therefore, preferred for this type of printer.[1,4] Besides printers using EP technology, there are other printing technologies such as flexography in which clustered dots are preferred especially in the midtones because the dot gain is lower and better controlled when the dots are clustered.[5] However, although periodic clustered (AM) halftones are quite smooth, they usually suffer from moiré, which occurs because of the periodic interference of different colorant channels. This issue is usually dealt with by adjusting each colorant channel at a specific angle in four-channel CMYK printing.[6] This adjusting causes another type of pattern, called rosettes, which occur with higher frequency than moiré and are not visible if the screen frequency is high enough. However, when the quality of the paper is not high, for example, newspaper paper, the screen frequency cannot be high enough and the rosette patterns are visible. Furthermore, in multichannel printing, which uses more than the conventional four CMYK inks, the problem with moiré is more serious and not so easy to handle when using AM halftones. Therefore, second-order FM halftones provide a solution because of their

*Address all correspondence to: Sasan Gooran, E-mail: sasan.gooran@liu.se

1017-9909/2015/$25.00 © 2015 SPIE and IS&T

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

stochastic nature of distributing the clustered dots, which makes them free from the problem of the periodic interference of different colorants.

This is why there have been many commercial screenings, such as Kodak Staccato screening and Fujifilm's TAFFETA screening, developed for producing stochastic clustered dots. There have also been inventions registered as patents that describe models to produce such halftones.[7–11] Furthermore, many models have been proposed in literature for producing stochastic clustered-dot halftones. Some of them are based on error-diffusion.[12–14] Levien[12] proposed an extension to error diffusion halftoning by an output-dependent feedback term, which can control the halftone patterns and texture with a minimum of computational expense. These new patterns were shown by Lau et al.[13] to be green-noise like, containing neither low-frequency nor high-frequency components. Li and Allebach used Levien's output-dependent feedback term to modify the threshold to gain more control over dot size and dot shape and also reduce the midtone artifacts.[14]

Besides these models based on error diffusion, there are also other models for second-order FM halftoning reported in the literature.[15–17] Lau et al.[15] proposed a technique for generating green-noise halftones by employing a dither array referred to as a green-noise mask. In Ref. 16, the authors proposed a donut filter approach to produce pleasing stochastic clustered-dot halftone patterns, referred to as AM-FM halftones. Gupta et al.[17] proposed a method based on direct binary search (DBS), which was originally developed for first-order FM halftoning. The DBS procedure is modified by use of different filters in the initialization and update phases. The proposed technique also gives the possibility to the user to control the clustered dot size to suit their needs.

One of the biggest challenges for halftoning methods to be applied in practice is their operational time. The iterative models, which commonly produce better halftones, generally operate on the whole or part of the input image. These types of methods are image dependent, and are directly applied to the original continuous tone image. For example, in order to produce a printed image of the size of an A4-page, i.e., approximately 8×11 in., at 1200 dpi an image of size $9600 \times 13{,}200$ pixels is to be halftoned. If the iterative methods were directly applied to such a large image it would require a large amount of data to be processed, which makes the computational procedure very slow. On the other hand, halftoning models, such as ordered dithering, that are operating point-by-point, are very fast. These types of methods are image independent and use predetermined threshold matrices, making the only computation for halftoning be the comparison between each pixel value in the original image with the corresponding value in the threshold matrix. This makes these techniques feasible to be used in practice and especially in prints using high print resolutions.

In a recently published paper, a stochastic clustered-dot halftoning method was introduced that parametrically controls the dot shape and seed placement adaption to the local image structure.[18] The proposed method is an extension of methods for the control of periodic halftones to irregular seed structures by using a spot function to define thresholds to be applied to an input image on a point-by-point basis.[18,19] By adjusting the involved parameters in the spot function, the dot cluster growth, touch points, cluster angles, and eccentricity in the halftone image are controlled.[18] However, the proposed stochastic technique is not able to produce perfectly symmetric patterns.[18] An extension of the proposed monochromatic halftoning method to a dot-off-dot vector halftoning is also introduced in Ref. 18. For an input pixel having, for example, three nonzero colorants, different thresholds are used for the darkest, second darkest, and lightest colorant.[18]

In this paper, we proposed a method to generate image-independent threshold matrices for first-order and second-order FM halftoning. Despite the stochastic nature of the proposed technique, it is possible to achieve symmetrical halftone structures, i.e., symmetrical clusters and voids in the two corresponding sides of the midtone. The proposed method is based on our previous iterative first-order FM model,[20] referred to as the iterative method controlling the dot placement (IMCDP) in the present paper. The approach in IMCDP is used to generate an image-independent first-order FM threshold matrix and is also modified to generate second-order FM threshold matrices. By choosing appropriate filters and filter parameters, the designer is given control over the clustered dot size, halftone structures, dot shape, and alignment. The proposed method also offers the possibility to control the dot size for more than one graytone by using different filter parameters in different graytones. An extension of the proposed monochromatic halftoning method to a color halftoning method producing dot-off-dot structures is also introduced in this paper. There are two possibilities to generate the threshold matrices for different colorant channels. The one introduced in this paper is to generate one threshold matrix for one colorant and calculate the other two based on the first one to achieve a dot-off-dot structure, meaning that the different colorant channels can be thresholded simultaneously using these three matrices and no further check of the colorant values in different channels is needed. The other possibility is to simultaneously generate the threshold matrices for different colorant channels to both maintain dot-off-dot structure and also homogeneously place the dots in each colorant channel with respect to the dots in the other channels.

The remainder of this paper is organized as follows. Section 2 provides a brief description of the original IMCDP method. Section 3 includes a description of the new method to generate first-order FM halftoning together with important parameters that affect the resulting halftones. In Sec. 4, we describe how the filter used in the model can be modified to generate second-order FM threshold matrices and how the halftone structure, the clustered dot size, shape, and alignment can be adjusted by the appropriate choice of filters and filter parameters. In Sec. 5, we illustrate halftone results using the proposed methods. The extension of the proposed method to dot-off-dot halftoning is introduced in Sec. 6 and Sec. 7 provides a brief conclusion.

In order to illustrate the results in a way that makes it possible to study the characteristics of the halftones and also minimizes the effect of dot gain, all halftones in this paper are printed at 150 dpi.

2 Iterative Method Controlling the Dot Placement

In this section, we briefly describe the IMCDP, which was originally published in Ref. 20 as "monochromic halftoning method."

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

2.1 *Iterative Method Controlling the Dot Placement Method*

In IMCDP, halftone dots are placed iteratively with the goal of reducing the difference between the original and the halftone image. The generation of the halftone image starts with a blank image the same size as the original. The total number of dots to be placed in the halftone image (or in a number of different graytone regions) is dependent on the original image's overall lightness/darkness (or its average tone value in different graytone regions) and, therefore, is known in advance. Starting with a blank initial image, in the first iteration, the algorithm finds the position of the darkest pixel (the pixel holding the maximum value) in the original image and places the first dot at that location in the halftone image. In the next step, the low-pass filtered version of the halftone image is subtracted from the low-pass filtered version of the original image. The low-pass filter used is a Gaussian filter with standard deviation 1.3 truncated to 11×11 pixels. This operation is addressed in Ref. 20 as the feedback process, which is also going to be referred to as the feedback process in the present paper. Subtracting the filter from the image around the found pixel reduces the pixel values in the neighborhood of that pixel, meaning that the chance of the neighboring pixels to be picked as the next maximum is reduced. Then, the location of the maximum pixel value of the subtracted image is found and at that location on the halftone image the next dot is placed. The process continues until the known number of dots is placed and the final halftone image is achieved.

2.2 *Filter Design*

Using an 11×11 Gaussian filter makes the method work quite well for almost all kinds of images. However, the dots in the extreme highlights (or shadows) are not placed as homogeneously as one would expect. The reason is that the 11×11 filter is not big enough to homogeneously distribute the dots in those regions. The average distance between the dots in a halftone, i.e., the principal wavelength, is decided by $\lambda_g = 1/\sqrt{g}$ for $0 < g \leq 1/2$, where g is the gray level.[2,3] Thus, the average distance between the dots in, for example, a halftone at 1%, i.e., $g = 0.01$, is 10 which requires a 21×21 filter. In order to distribute the dots as homogeneously as possible in the very light (and very dark) regions in the proposed method, the size of the Gaussian filter (or its standard deviation) is a variable of the gray level of the region where the maximum is found. The mentioned 11×11 Gaussian filter is used in the areas with tonal values between 4% and 96% and for the rest of the image, a filter with varying size (or standard deviation) is used. The principal wavelength corresponding to the tones decides the size of the filter.[20]

3 First-Order Frequency Modulated

In this section, how to design the threshold matrix for the first-order FM is described. The goal is to generate halftones having a blue-noise characteristic, meaning a homogeneous distribution of dots in the halftones.

3.1 *Threshold Matrix Generation*

The procedure for generating an image-independent threshold matrix is very similar to that of halftoning an image by IMCDP. From now on, the abbreviation TMG is used to refer to the threshold matrix generation method in this paper. The main difference between IMCDP and TMG is that in the former method the input is the image being halftoned while in the latter the input is an image holding random numbers. Let us describe TMG by describing how a 256×256 threshold matrix is generated. The input image (matrix) is the same size as the intended threshold matrix (i.e., 256×256) holding uniformly distributed pseudorandom numbers. The random numbers can, in principal, vary within any interval, but the feedback filter has to be chosen accordingly, see Sec. 3.2. In our method, the input matrix contains random numbers on the open interval (0, 0.01). In IMCDP, a predecided number of pixels are iteratively set to 1 in the initial blank image. In TMG, the initial blank matrix is iteratively filled by the numbers from 1 to $256^2 = 65{,}536$. Note that in the halftones, it is very important to have both the "black pixels" and the "white pixels" homogeneously distributed in the highlights and shadows, respectively. It is also very important to have symmetrical dot distributions (or dot shapes) on both sides of the midtone, i.e., 50%. For example, the black dots at 40% should have the same structure/shape as the "white dots" at 60%. Since the smallest threshold values being placed, i.e., 1, 2, 3, etc., are important for the highlights, and the largest values being placed, i.e., 65,534, 65,535, 65,536, are important for the shadows, in our design, we place two threshold values in each iteration, one small and one large. This way the designed threshold matrix will generate symmetric halftone structures in the highlights and shadows. In order to do that, we create two input matrices containing random numbers on the open interval (0, 0.01). For the sake of simplicity let us call these two matrices P for the light tones up to 50% and Q for the dark tones from 50% to 100%. At the first iteration, a 1 is first put in the initial blank threshold matrix at the position where P holds the maximum value and a large negative number is put at this position in both P and Q to make sure that this position would not be found as the maximum anymore. Then, the feedback process, i.e., subtracting a filter around the found maximum from the input matrix, is performed on P. After this, in the same iteration, the largest threshold value, i.e., 65,536, is put at the position where Q holds the maximum value and a large negative number is put at this position in both P and Q to make sure that this position would not be found anymore. Then, the feedback process is performed on Q. After that, the first iteration is terminated and the pixel positions where the subtracted matrices, i.e., modified P and Q, hold the maximum values are found again and set to 2 and 65,535 in the threshold matrix. This procedure continues until the last iteration, i.e., iteration $256^2/2 = 32{,}768$, where the last two empty positions in the threshold matrix are filled with 32,768 and 32,769. Note that when the threshold matrix is generated, it must be normalized between 0 and 1 if the original image is scaled to [0, 1]. This is done by dividing all values in the generated threshold matrix by the maximum value plus 1. From now on in this paper, we assume that the generated threshold matrices and the images being halftoned are normalized between 0 and 1, 0 representing white and 1 representing black. It is obvious that the filter plays a significant role in the generation of the threshold matrix. Before describing how to design the filter in Sec. 3.2, there is one important point worth mentioning here.

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

The important point is how to perform the feedback process when the found maximum is close to an edge or a corner of the matrix. In IMCDP, the filter could be cut when outside the border of the image. In TMG, this situation has to be considered because the generated threshold matrix might be smaller than the image being thresholded and, therefore, has to be tiled to be the same size as the image. Cutting the feedback filter when outside the border of the matrix will cause boundary artifacts where the matrices are joined. This issue is coped with in TMG by performing a wrap-around process. For example, if the found maximum is close to the right edge, those parts of the filter that are outside the right border will be subtracted from the mirror side of it on the left edge. If the found maximum is, for example, close to the up-right corner, parts of the filter are also subtracted from the corresponding parts of the down-left corner. By using this wrap-around process, the boundary artifact is prevented, which will be further discussed in Sec. 3.3.

3.2 *Filter Design*

As discussed in Sec. 3.1, the generation of a well-formed threshold matrix and thus, well-formed halftones are very much dependent on the filter being used in the feedback process. Thus, in this section, how to design an optimal filter resulting in well-formed first-order FM halftones is discussed.

3.2.1 *Standard deviation of the filter*

As in IMCDP, the following Gaussian filter, Eq. (1), is used to perform the feedback process

$$f(m,n) = \mathrm{e}^{\frac{-(m^2+n^2)}{2\sigma^2}}, \tag{1}$$

where (m,n) and σ denote the position and the standard deviation, respectively. The goal is, as in IMCDP, to place the dots as far apart as possible in the highlights (and shadows). This means, the smallest (and largest) threshold values should be placed farther apart than those in the middle. For example, for tonal coverages of $g = 0.01$, $g = 0.02$, and $g = 0.04$, the principal wavelength (the average distance between the dots in a halftone) is $\lambda_g = 1/\sqrt{g} = 10$, 7.1, and 5, respectively. A well-formed (blue noise) halftone pattern of a fixed gray-level should consist of isolated dots with an average distance close to λ_g.[2,3] Therefore, in order to have well-formed halftones, the consecutive threshold values should be placed with a distance close to λ_g. To give an indication of how to choose an appropriate σ, assume that the filter is truncated when its weights are smaller than 0.001. Then an appropriate σ can be found in Eq. (2) for a given gray level g

$$\mathrm{e}^{\frac{-1}{2g\sigma^2}} = 0.001, \tag{2}$$

where the square of the distance to the center of the filter, i.e., $m^2 + n^2$, in Eq. (1) has been replaced by $\lambda_g^2 = 1/g$. Equation (2) gives $\sigma = 2.7$, $\sigma = 1.9$, and $\sigma = 1.3$, for $g = 0.01$, 0.02, and 0.04, respectively. In IMCDP, which is an image-dependent method, in each iteration σ was a variable of the tonal value of the region where the maximum was found. If we now use the same strategy, then $\sigma = 2.7$ has to be used to fill the threshold values from 1 to 655 ($\approx 0.01 \times 256^2$) in order to give a principal wavelength close to 10. Then $\sigma = 1.9$ should be used to fill the threshold values from 666 to 1311 ($\approx 0.02 \times 256^2$), and so on. Since TMG is supposed to generate an image-independent threshold matrix, such a big change in σ over a small tonal variation will not result in a well-formed threshold matrix. Consider an image of constant gray level 0.01 that is thresholded with this type of threshold matrix with varying σ according to Eq. (2). The halftone result will look good and homogeneous because only the dots corresponding to threshold values 1 to 655 are being placed in the halftone. The reason is that those positions in the threshold matrix hold values less than 0.01. Consider now another image of constant gray level 0.1 being halftoned with the same threshold matrix. All the dots corresponding to threshold values from 1 to 6554 ($\approx 0.1 \times 256^2$) are now placed in the halftone, but many of them are not placed using an appropriate σ for $g = 0.1$. Therefore, these types of threshold matrices only result in good halftones at very low coverages. Figure 1 shows halftones at $g = 0.02$ and $g = 0.1$ generated by a threshold matrix with variable σ according to Eq. (2) and IMCDP. It must be pointed out that in order to avoid sudden changes in σ, it was varied very slowly, and not, for example, suddenly from 2.7 at 0.01 to 1.9 at 0.02. This gradual change has been done by interpolation with a step of 0.001. Figure 2 shows these halftones' RAPS curves.[2,3] The principal frequencies, defined by $f_g = 1/\lambda_g = \sqrt{g}$ for $0 < g \leq 1/2$, are 0.14 and 0.32 for the two examples and are shown in Fig. 2. As shown in Figs. 1 and 2, the result of threshold halftoning a halftone at 2% looks very good and its RAPS shows a well-formed blue-noise characteristic, similar to the halftone generated by IMCDP. Figures 1(c) and 2(b) solid curve show that the thresholded halftone at 10% is not well formed and the peak of its RAPS curve is not at the principal frequency. The IMCDP halftone at 10%, on the other hand, has a blue-noise characteristic, verified by its

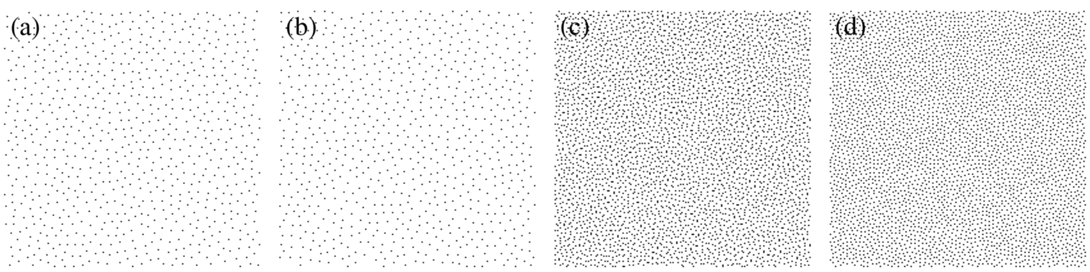

Fig. 1 Two halftones at 2% and 10% generated: (a), (c) by a threshold matrix with variable sigma according to Eq. (2) and (b), (d) by iterative method controlling the dot placement (IMCDP).

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

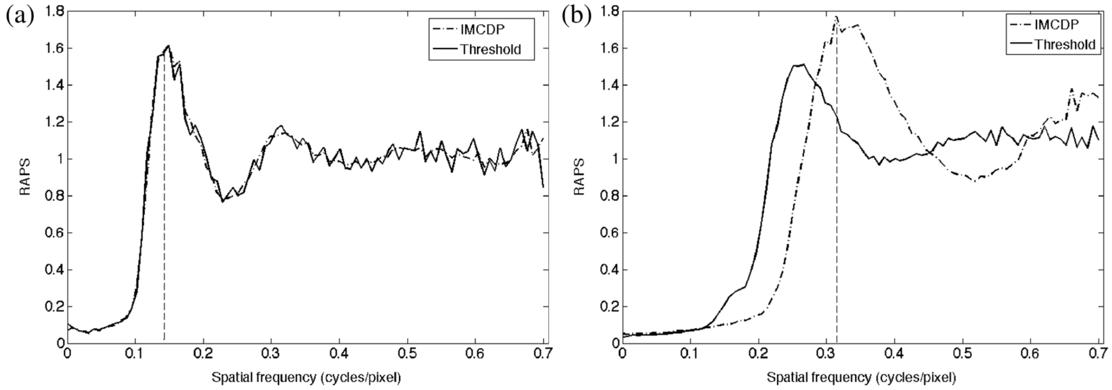

Fig. 2 Radially averaged power spectrum (RAPS) curves for halftones using IMCDP and threshold matrix with varying standard deviation: (a) 2% coverage and (b) 10% coverage. The principal frequency is shown.

RAPS curve. This means that a variable standard deviation according to Eq. (2) cannot be applied to the threshold generation process unless it changes very little, which will be further discussed in Sec. 3.2.3.

3.2.2 *Optimizing the filter*

As discussed in Sec. 3.2.1, a threshold matrix with varying standard deviation according to Eq. (2) only results in well-formed halftones at low coverages. Therefore, in generation of the threshold matrix, the standard deviation of the filter should either be constant or vary over a small interval. Let us first focus on finding an optimized constant standard deviation for the generation of the threshold matrix. A large standard deviation will surely work better for very light tones and a small one better for a bit darker tones. Thus, the optimized σ is somewhere in between. For the very light tones, we already know that σ should be around 2.7 or larger. Let us now find out what the smallest appropriate value of σ could be. According to Eq. (2), for a halftone at 25%, we need to use $\sigma = 0.54$. Figure 3 shows RAPS curves for a halftone at 25% being halftoned by threshold matrices using four different constant standard deviations, namely $\sigma = 0.54$, 1, 1.2, and 1.4. As clearly shown in Fig. 3, a standard deviation according to Eq. (2), in this case $\sigma = 0.54$ for $g = 0.25$, is very small and causes a nonwell-formed halftone pattern. The reason is that using a small σ causes periodic structures in this tonal range. By comparing the curves in Fig. 3, it is noticed that σ around 1.2 is optimal for this halftone. The principal frequency, $\sqrt{0.25} = 0.5$, is also shown.

Therefore, according to Fig. 3, the smallest appropriate value for standard deviation cannot be smaller than 1.0. In order to find the optimal standard deviation we use the following measure. For each halftone, we calculate the distance from a dot to its closest dot, which gives a set of distances. The average of this set gives the average distance between dots in the halftone, corresponding to the principal wavelength. The average distance can be used as one measure. However, a high average distance does not necessarily mean homogeneously placed dots. For that, the standard deviation of the set needs to be calculated as well. A small standard deviation means that the distances are close to the average, which means homogeneously placed dots. Another useful measure could be the ratio of this standard deviation to the average distance. Note that if the dots are placed in a grid, then although the standard deviation is zero, its RAPS curve will show a spike (or spikes) indicating that the halftone does not have the intended blue-noise characteristic.[1] Therefore, this measure can only be used if further checks with RAPS curves are made. In order to find the optimal σ, halftone patches at 1%, 2%, and up to 25% coverage were created using the threshold matrix with different constant σ:s between 1 and 2.7 with a step of 0.1. For each σ, the sum of the average distances, the standard deviations, and the ratios of standard deviation to the distance were calculated. The largest sum of the average distances occurred for $\sigma = 1.3$, but all σ:s from 1.1 to 1.5 gave almost the same sum. The sum of the standard deviations was minimized for $\sigma = 1.1$, but σ between 1.1 and 1.3 gave a very close sum. The smallest sum of the ratios occurred for $\sigma = 1.1$. Therefore, a σ around 1.1 is an appropriate and optimized standard deviation according to the used measures. The PARS curves for all patches have been checked to make sure that no periodic structure occurs. Figures 4(a) and 4(c) show halftones at 2% and 10% generated by the threshold matrix using a constant $\sigma = 1.1$. Their corresponding RAPS curves (dashed curves) are shown in Fig. 5. Setting $\sigma = 1.1$ in Eq. (2) gives $g = 0.06$, meaning that $\sigma = 1.1$ is too small to result in well-formed halftones for tones

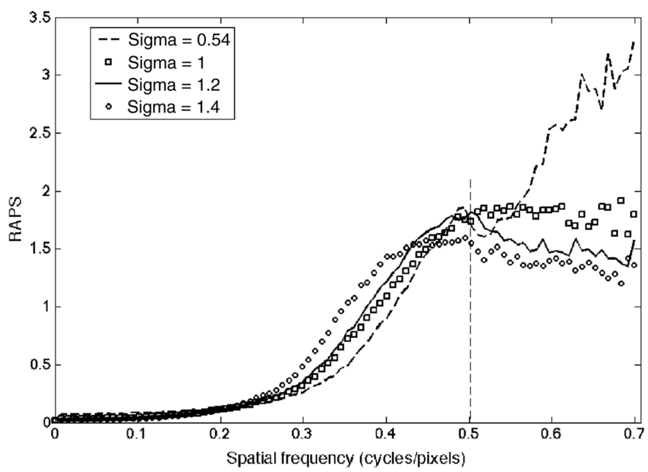

Fig. 3 RAPS curves for a 25% patch halftoned by threshold matrices using four different standard deviations. The principal frequency is shown.

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

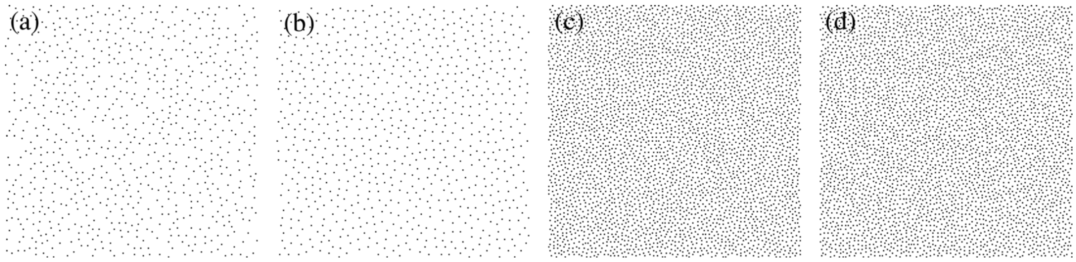

Fig. 4 Two halftones at 2% and 10% generated by a threshold matrix: (a), (c) with constant sigma = 1.1 and (b), (d) with variable sigma.

lighter than 6%, which is also verified by Figs. 4(a) and 5(a). Therefore, if a filter with a constant σ is being used and the very light (and dark) tones are of importance, it is better to use a bit larger σ, e.g., around 1.5.

3.2.3 *Variable standard deviation*

According to the results of the distance measures, $\sigma = 1.1$ gave the optimal results on the tonal interval (0, 0.25], but a bit larger σ up to 1.5 also resulted in fairy good halftones. Thus, a σ varying from 1.5 at 1% to 1.1 at 6% and then constant will surely result in better halftones for very light tones and almost the same halftones for other tones. Recall that $\sigma = 1.1$ corresponds to $g = 0.06$ (6%) according to Eq. (2). As already discussed in Sec. 3.2.1, if a variable sigma is being used then it should not change over a large interval. The question now is how large this interval can be made. In order to figure it out, we changed the sigma on the interval $[1.1, x]$, x being the variable, and halftoned patches from 1% to 25% with generated threshold matrices and calculated the distance measures to find an optimal x. According to the results, if x is around 1.7 the halftones at 2% coverage and lighter are fairly good and much better than using a constant $\sigma = 1.1$. The halftones between 2% and 6% are well formed and better than using a constant sigma because in this range the correct sigma according to Eq. (2) is used. For darker tones, the results are almost as good as using a constant sigma.

Figures 4(b) and 4(d) show halftones at 2% and 10% being halftoned with the threshold matrix generated with a variable σ varying from 1.7 at 0.01 to 1.1 at 0.06 and then equal to 1.1 for darker tones. The corresponding RAPS curves for halftones shown in Figs. 4(b) and 4(d) are shown in Fig. 5. As shown in Figs. 4(a), 4(b), and 5(a), the halftone created by using a variable sigma shows a much better blue-noise characteristic at 2%. By comparing the images in Figs. 4(c) and 4(d) and the curves in Fig. 5(b), it can be concluded that using variable sigma results in almost the same well-formed halftone at 10% as using a constant sigma. Note that here only the light tones were explained because, for the sake of symmetry, the same variable sigma is always used in the very dark tones. This means that σ is 1.7 for lighter tones than 0.01 and varies from 1.7 at 0.01 to 1.1 at 0.06. After that σ is kept constant until 94%, and then it is slowly increased to 1.7 at 99% and is kept constant at 1.7 for darker tones.

3.3 *Tiling Effect*

In ordered dithering algorithms, a deterministic threshold matrix is used for halftoning. The threshold matrix is designed based on a number of factors such as print resolution (dpi), screen frequency (lpi), and halftone dot shape, ordered dispersed or clustered dots, screen angle, etc. The size of the threshold matrix is also directly related to the number of gray levels being represented.[2] The larger the threshold matrix, the more gray levels it can represent. For example, a 15×15 threshold matrix can represent up to 226 gray levels. A screen at angles other than 0 or 90 deg requires slightly larger threshold matrices to reproduce the same number of gray levels. However, in ordered dithering the size of the threshold matrix is much smaller than the images being halftoned. Thus, the small threshold matrices are repeated (or tiled) to make a larger matrix the same size as the original image in order to be used in ordered dithering. Then each pixel value in the image is compared

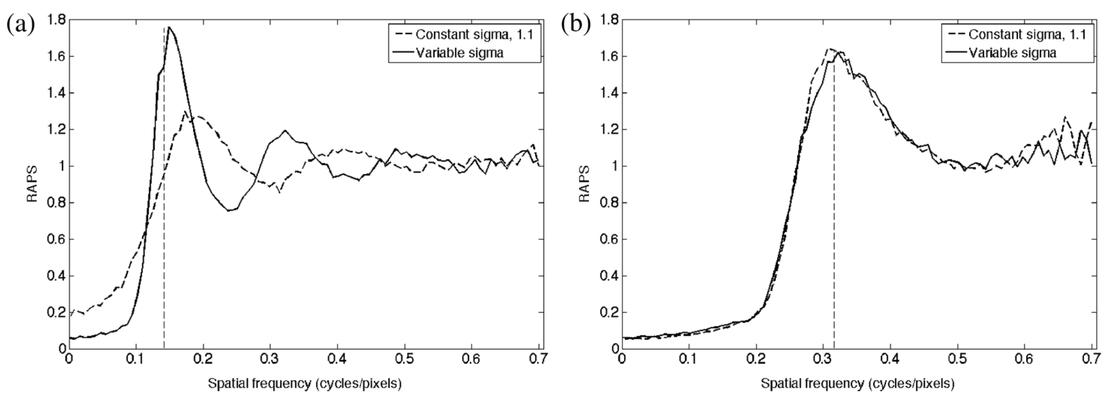

Fig. 5 RAPS curves for halftones using a threshold matrix with constant and variable sigma: (a) 2% coverage and (b) 10% coverage. The principal frequency is shown.

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

with its corresponding threshold value in the larger threshold matrix. Depending on the pixel value being greater or less than the threshold, a 1 or 0 is put at that pixel position in the output halftone image. Therefore, images halftoned by such threshold matrices contain periodic structures because of the tiling, most commonly seen in conventional AM halftones. In the proposed approach for generating a threshold matrix, there is no constraint on the size of the threshold matrix and it can be generated as large as possible to avoid tiling. Since the matrix is designed once and can be used thereafter independent of the image being halftoned, the operating time for generating a large threshold matrix is not an issue. However, there is a risk that a very large threshold matrix would not be able to accurately reproduce the tones and would also lose details in small regions of the original image. In this section, we will first demonstrate what would happen if identical threshold matrices are tiled to halftone a larger image by focusing on two important issues. The first one is to study the blue-noise characteristic of the halftones after being thresholded by tiling a number of identical threshold matrices. The second one is to study whether boundary artifacts occur at the junctions between the tiled threshold matrices.

In order to study the former issue, images of constant gray levels were created and halftoned by a threshold matrix of the same size and other threshold matrices built by tiling a number of identical smaller matrices. Then their RAPS curves were studied to analyze their blue-noise characteristic. Figure 6 shows a 256×256 image of constant gray level 0.1 being halftoned by threshold matrices with a size of 256×256, 128×128, 64×64, and 32×32. The RAPS curves of the first three and the principal frequency ($\sqrt{0.1} = 0.316$) are also shown in Fig. 6.

By observing the patches and the RAPS curves in Fig. 6, it is clearly seen that when the threshold matrix contains 4×4 or more identical submatrices, the blue-noise characteristics of the halftones are not preserved because of the periodic structure caused by tiling. This can be verified by both looking at the patches and also visually observing the oscillations and the spikes in the RAPS curves. Hence, when halftoning an image containing large homogeneous parts by the generated first-order FM threshold matrix, the best result would be obtained if tiling of identical threshold matrices is avoided as much as possible.

The other important issue is to study the presence of the boundary artifacts. As explained in Sec. 3.1, in the generation of the threshold matrix, this issue was taken into account when performing the feedback process. This was done by utilizing the wrap-around strategy, i.e., subtracting the filter values from the pixels on the mirror side of the found maximum if it was close to an edge or a corner. In order to study the possible discontinuities, many tests have been done, mostly by studying gray-scale ramps being halftoned. In Fig. 6, one of these ramps is shown. The ramp is divided into two parts, i.e., the upper part from 0% to 50% (left to right) and the lower part from 50% to 100% (right to left). Each part is filled by three rectangle tiles. No discontinuity is observed in this image. This was also verified by actual test prints at high resolutions using both offset at 1200 dpi and inkjet at 600 dpi. The conclusion is that the transitions are very smooth and the tiling does not add any discontinuities to the halftones.

As discussed earlier in this section, there is, however, a risk that a large threshold matrix would miss small details or/and would not be able to accurately reproduce the gray-tones in small portions of an image. If there is a demand for

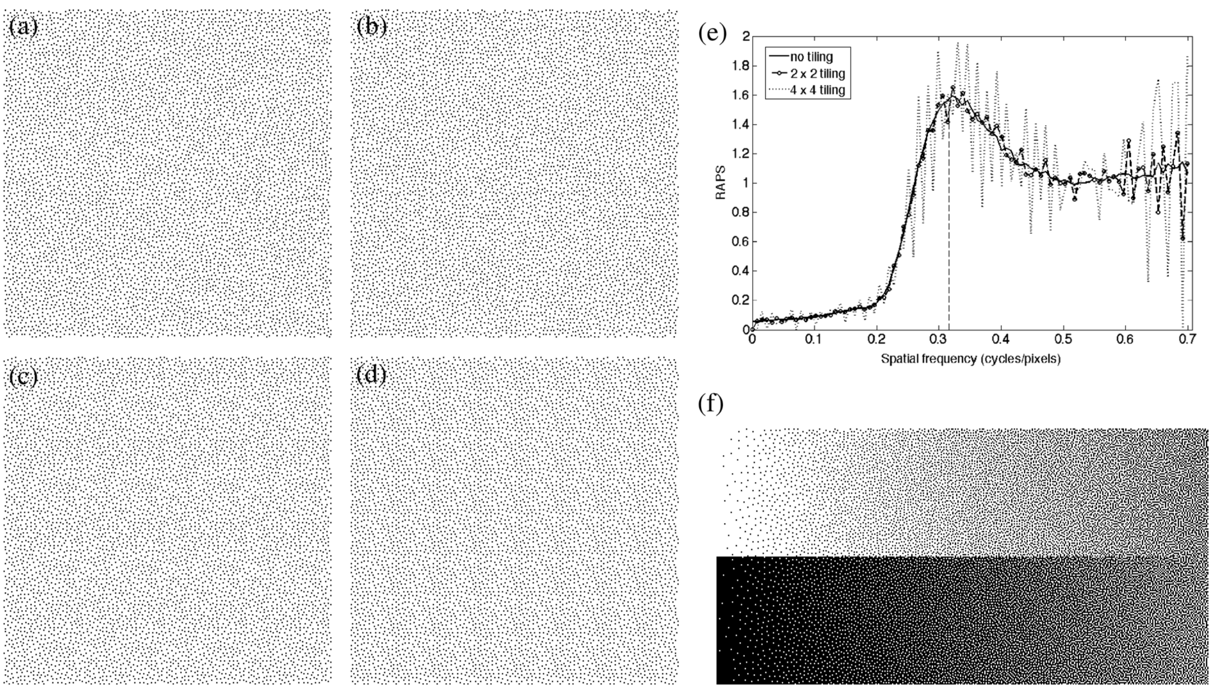

Fig. 6 Constant image at $g = 0.1$ and gray-scale ramp halftoned by tiled threshold matrices. (a) Threshold matrix is the same size as the image. (b) Threshold matrix consists of 2×2 smaller identical matrices. (c) Threshold matrix consists of 4×4 smaller identical matrices. (d) Threshold matrix consists of 8×8 smaller identical matrices. (e) The RAPS for three of the patches are shown. (f) The ramp is shown in two parts and each part is filled by three rectangle tiles.

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

not having the threshold matrix too large and at the same time avoiding the tiling effect, the proposed approach can be slightly modified to meet this demand. The tiling effect and how to reduce it have been addressed in the literature.[21,22] In Ref. 21, a new class tiling designed dot diffusion was proposed to reduce the periodic artifacts by manipulating the class tiling with comprising rotation, transpose, and alternative shifting of the class matrices. According to their inspection, the source of the periodic artifacts in the dot diffusion algorithm is its regular class matrix arrangement with the same direction and relative positions. This is coped with by replacing the regular arrangement with the conditional random generated manner to design class tiling. Kacker and Allebach showed that the random tiling of the screens would eliminate the periodicity in the halftones.[22] In Ref. 22, the screens are designed using DBS halftoning method and the screens are designed in such a way that the boundary artifacts are prevented. The screens were also trained using a database of high-resolution images to improve halftone quality. Both approaches suggest a random tiling to arrange the threshold matrices to reduce the periodic artifacts. In the following, we propose two approaches suited to the halftoning method in the present paper to randomly tile the threshold matrices.

The first approach is very similar to what has been proposed in Ref. 22, in which a number of threshold matrices are randomly tiled to remove periodicity. The main concern is to design the threshold matrices in a way that the boundary artifacts are prevented. Assume that K small threshold matrices are being generated to be randomly tiled to make a large threshold matrix. In the proposed approach, these K threshold matrices are generated simultaneously as described in Sec. 3.1. There are, therefore, $2K$ input matrices of pseudorandom numbers. In each iteration and for each of these K threshold matrices, a threshold number is put in the position where the maximum is found and the feedback process is performed as explained in Sec. 3.1. The only difference here is that if a maximum is found close to a border in any of the threshold matrices, the parts of the filter outside the matrix boundary are subtracted from its mirror side in all of the K threshold matrices. For example, if a found maximum for a threshold matrix is close to the right edge, those parts of the filter that are outside the right edge are subtracted from the mirror side of it in the left edge of all of the K threshold matrices. Note that since the border pixels in each matrix are affected K times more than the inside pixels, the filter values have to be divided by K when being subtracted from the border pixels. This way all the K threshold matrices can be randomly tiled from any direction without causing any boundary artifacts. Figures 7(a) and 7(b) show the RAPS curves for a 1024×1024 constant image at 10% being halftoned by a 256×256 threshold matrix and four 256×256 threshold matrices generated as proposed and randomly tiled, respectively. This means that in the former case, only one 256×256 matrix and in the latter case four 256×256 matrices need to be saved in the memory. When using a 256×256 threshold matrix, identical matrices are tiled to make a 1024×1024 matrix; therefore, the halftone is highly structured because exactly the same structure is repeated 16 times. That is why the corresponding RAPS curve does not represent a well-formed blue-noise characteristic, see Fig. 7(a). The RAPS curve for the halftone using the proposed random tiling approach, on the other hand, indicates a better-formed halftone pattern as the oscillations and the spikes are less evident in Fig. 7(b). If more threshold matrices were generated and randomly tiled, the halftones would certainly suffer less from periodical structure.

In the proposed second approach to reduce the tiling effect, instead of using a predefined number of nonidentical small threshold matrices and randomly tiling them, all of the small threshold matrices making the large threshold matrix are nonidentical. Let us explain the second approach by using an example to generate a 1024×1024 threshold matrix by 16 nonidentical 256×256 matrices. Instead of generating a 1024×1024 threshold matrix by filling the empty initial matrix with numbers $1,2,\ldots,1024^2$, 16 256×256 threshold matrices are generated simultaneously, filling each one with numbers $1,2,\ldots,256^2$. When all of these 256×256 matrices are filled, they are tiled to be 1024×1024. The most suitable approach for doing that is to let the algorithm start with a 1024×1024 initial empty threshold matrix and two 1024×1024 input images containing pseudorandom numbers precisely like before. Call these two images containing random numbers M and N. Instead of searching for the maximum over the entire 1024×1024 image, in this modification, the maximum values at each of the 16 256×256 subimages are found. Therefore, in the first iteration, 16 threshold number 1s are placed where the subimages in M hold their maximum values and 16 threshold number $256^2 = 65{,}536$ are placed where the subimages in N hold their maximums. The feedback process is performed exactly like before. This procedure continues until 16 $258^2/2 = 32{,}768$ have

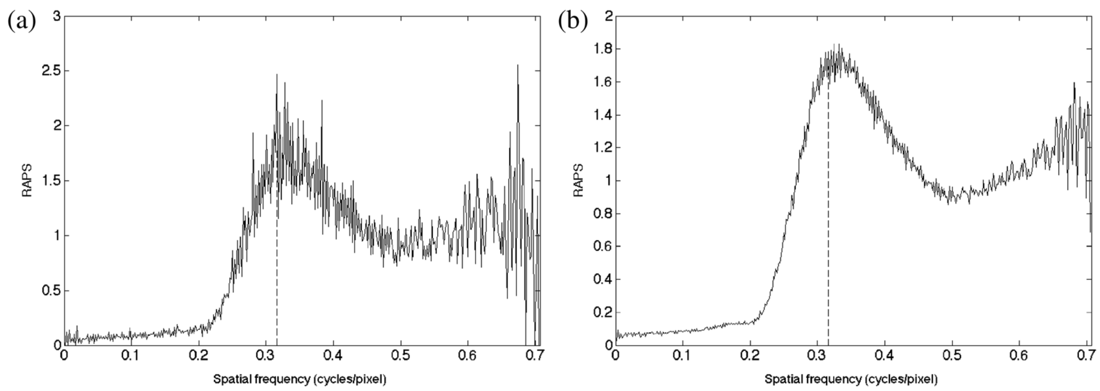

Fig. 7 RAPS curves for 1024×1024 halftones at 10% using: (a) a 256×256 threshold matrix, (b) four nonidentical threshold matrices randomly tiled. The principal frequency is shown.

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

been placed where the modified subimages in M hold their maximum values and 16 32,769 have been placed where the modified subimages in N have their maximum values. Now, if a 1024×1024 image is halftoned with such a threshold matrix, the periodic structure shown in Fig. 6 is significantly reduced because the distribution of dots is different over each 256×256 portion from that of the other 256×256 portions of the halftone. This means that the 1024×1024 threshold matrix is tiled by 16 nonidentical 256×256 threshold matrices in a randomized manner. The possible risk of losing details in small parts of an image because of a too large threshold matrix is also reduced. Figure 8 shows the RAPS curves for a 1024×1024 constant image at 10% being halftoned by a 1024×1024 threshold matrix, containing 16 256×256 nonidentical matrices, generated as proposed. This RAPS curve indicates a well-formed halftone pattern. Since, in this approach, 16 nonidentical matrices are tiled, the periodicity is removed and the result is better than using the first approach where four nonidentical matrices were randomly tiled. This can be verified by comparing the RAPS curve in Fig. 8 with that in Fig. 7(b). The disadvantage is that since in this example 16 nonidentical 256×256 matrices are used, the memory requirement is four times that required for the first approach.

Note that, in this example, if the image being halftoned is larger than 1024×1024, the halftone pattern is still periodic with a period of 1024×1024. However, although a 1024×1024 screen is large enough to avoid the perception of periodicity by repeated identical threshold matrices even at very high print resolutions,[22] it is possible to generate larger threshold matrices with the same approach. For instance, a 4096×4096 threshold matrix can be generated by 256 nonidentical 256×256 threshold matrices by the second approach. The memory requirement will, of course, be much higher.

A combination of the first and the second approach to avoid tiling effect is also a possible alternative approach to generate large threshold matrices consisting of many nonidentical and randomly tiled smaller threshold matrices.

Figures 7 and 8 show that the proposed methods are able to remove or reduce periodicity and at the same time make

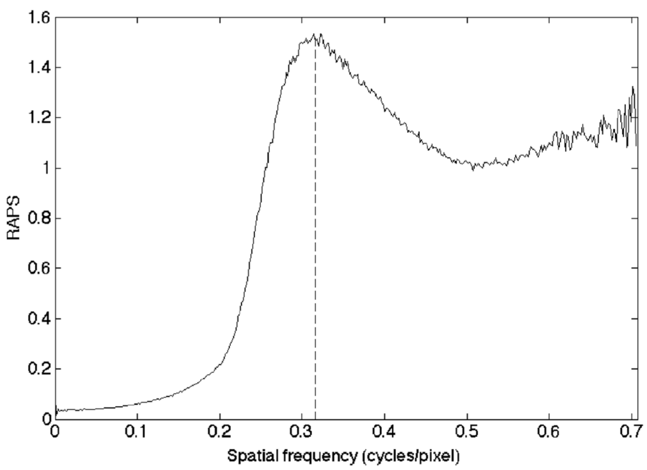

Fig. 8 RAPS curves for 1024×1024 halftones at 10% using a 1024×1024 matrix consisting of 16 nonidentical 256×256 threshold matrices generated by the second approach to avoid tiling effect. The principal frequency is shown.

the threshold matrix contain smaller nonidentical submatrices in order to avoid the possible loss of small details in a large image.

4 Threshold Matrix Generation Second-Order FM

In this section, how to design the threshold matrix for a second-order FM is described. Unlike the first-order FM where the main goal was to produce well-formed halftones, in generating second-order FM halftones the main goal is to obtain threshold matrices that fulfill the needs of the designer to change the halftone structure, clustered dot size, shape, and alignment by adjusting the filters and the involved parameters.

4.1 *Threshold Matrix Generation*

The procedure for generating an image-independent threshold matrix representing second-order FM is very similar to that of generating a first-order FM threshold matrix described in Sec. 3. The main difference between them is the filter being used. In first-order FM, we wanted the single dots to be as far apart as possible, while in the second-order, in addition to that, we also want them to grow in size when the tones get darker. The goal is, therefore, to fill the halftone with separated single dots until a certain tone level and then make them cluster and grow to a certain size. Thus, the function in Eq. (3), which is a Gaussian function subtracted from another Gaussian function with a larger standard deviation, is an appropriate filter for this purpose

$$h(m,n) = \mathrm{e}^{\frac{-(m^2+n^2)}{2\sigma_1^2}} - \mathrm{e}^{\frac{-(m^2+n^2)}{2\sigma_2^2}}, \tag{3}$$

where $\sigma_1 > \sigma_2$. Figure 9(a) shows the radial one-dimensional representation of $h(m,n)$, i.e., $h(r)$, versus the distance to the center of the filter, i.e., the radius $r = \sqrt{m^2 + n^2}$, for fixed $\sigma_1 = 3.3$ and three different σ_2: 0.7, 1.5, and 2.7. The maximum of $h(r)$ occurs at p, marked in Fig. 9(a), for which we have

$$p^2 = \frac{4\sigma_1^2\sigma_2^2 Ln\frac{\sigma_1}{\sigma_2}}{\sigma_1^2 - \sigma_2^2}. \tag{4}$$

Figure 9(b) shows p^2 versus σ_2 for three different σ_1: 1.3, 2.3, and 3.3. As shown in Fig. 9(b), p^2 increases almost linearly with respect to σ_2 for $0.5 < \sigma_2 < \sigma_1$.

If the feedback process is performed using $h(m,n)$ in Eq. (3), the pixel values around the found maximum are decreased with a radius decided by σ_1. After the single dots have been distributed, then the dots start to cluster and the maximum size of the clustered dots will depend on σ_2 (or p). For example, using filters with $\sigma_1 = 3.3$ and two $\sigma_2 = 1.5$ and $\sigma_2 = 2.7$ shown in Fig. 9(a) makes the maximum radius of the clustered dots be around 3 and 4, respectively, which means maximum clustered dot areas of approximately 28 and 50. These numbers are, of course, based on a rough estimate providing that all clustered dots are the same size and are perfectly circular, which is neither the case here nor the goal of second-order FM halftoning. Perfectly circular clustered dots with the same size can be achieved by an AM designed threshold matrix. However, for a fixed σ_1, a larger σ_2 will make the clustered dots grow faster and also reach a larger area. In Sec. 5, we

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

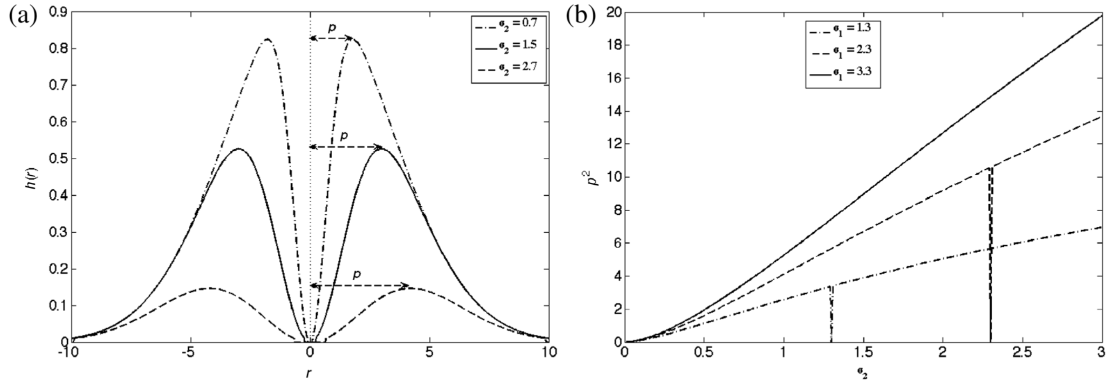

Fig. 9 (a) Filter $h(r)$ for fixed $\sigma_1 = 3.3$ and three different σ_2: 0.7, 1.5, and 2.7, (b) p^2 in Eq. (4) versus σ_2 for three different σ_1: 1.3, 2.3, and 3.3.

show some halftones using the proposed threshold matrix for second-order FM using fixed $\sigma_1 = 3.3$ and two different σ_2.

4.2 *Filter Design*

As shown in Sec. 4.1, the generation of a threshold matrix is very much dependent on the filter being used in the feedback process. Therefore, in this section, how to design the filter is discussed. The goal here is to discuss how appropriate choices of σ_1 and σ_2 can be made to meet a specific demand for the size of the clustered dots at a certain gray level.

4.2.1 *Standard deviations of the filter*

In order to have a better control over the filter size, let us truncate the Gaussian filters where the holding weights are less than 0.01, which is the maximum value in the input random image. Therefore, for the Gaussian filter distributing the single dots Eq. (5), can be used to decide σ_1

$$e^{\frac{-1}{2g\sigma_1^2}} = 0.01, \tag{5}$$

where g is the tonal value (gray level), see also Eq. (2). Assume that we want the dots to start being clustered at 0.01. Putting $g = 0.01$ in Eq. (5) gives $\sigma_1 = 3.3$. Now, if we choose $\sigma_2 = 1.5$, as shown in Fig. 9(a), the clustered dots will grow until their radius is around 3 (because p is around 3). Choosing $\sigma_2 = 2.7$ will make the clustered dots grow to a radius around 4. When the clustered dots reach this radius, then more single dots will be placed in empty spaces, which will then grow in size when the tones get darker. Choosing appropriate σ_1 and σ_2 is, therefore, dependent on the application. A larger σ_1, as discussed, makes the dots cluster at lighter tones and a very small σ_1 makes the halftone look like a first-order FM halftone. When σ_1 is decided, then a larger σ_2 makes the clustered dots grow faster and reach a larger area. A very small σ_2 makes the halftone look like a first-order FM halftone. Figures 10(a) and 10(b) show the average clustered dot areas versus σ_2 for three different σ_1, i.e., 2.3, 2.7, and 3.3, for halftones at 10% and 25% coverage, respectively. In all cases, σ_2 was varied from 0.5 to $0.85 \cdot \sigma_1$. The curves were obtained by first labeling each binary halftone to localize the separated clustered dots (connectivity 8 was used). Then the dots at the borders that would have been connected if the threshold matrix was repeated were also connected by giving them the same label. After that the average clustered dot area was simply calculated by taking the average of the size of the labels.

The first observation is that the relationships within the ranges they were calculated are linear, which was expected because of the linear relationship between p^2 and σ_2 shown in Fig. 9(b). Assume now, for instance, one wants to have an average cluster dot size of 16 at 25%. Just to give an indication what this size at 25% means, consider an AM halftone at 25% using 1200 dpi (print resolution) and 150 lpi (screen frequency). This means a halftone cell of the size 8×8,

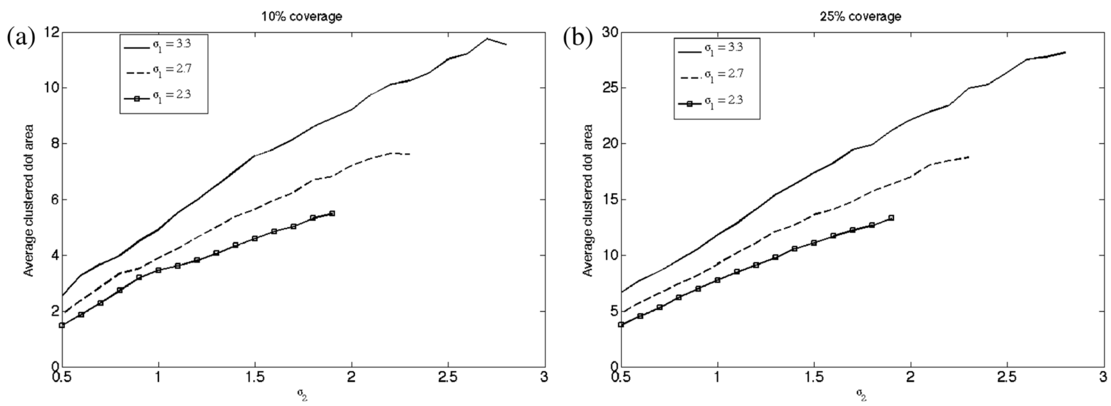

Fig. 10 The average clustered dot areas versus σ_2 for three different σ_1, i.e., 2.3, 2.7, and 3.3 for halftones at: (a) 10% coverage and (b) 25% coverage.

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

which for 25% will mean a halftone dot area of $0.25 \times 64 = 16$. According to Fig. 10(b), among these three plotted possibilities, there are two options; either ($\sigma_1 = 3.3$ and $\sigma_2 = 1.4$) or ($\sigma_1 = 2.7$ and $\sigma_2 = 1.84$). For $\sigma_1 = 2.3$, the clustered dot area does not reach 16 at 25%. For ($\sigma_1 = 3.3$ and $\sigma_2 = 1.4$) and ($\sigma_1 = 2.7$ and $\sigma_2 = 1.84$), the average clustered dot sizes at 10% are almost 7 and 6.7, respectively, see Fig. 10(a). A 10% AM halftone at 1200 dpi and 150 lpi means a halftone dot size of 6.4. Figure 11 shows two halftones at 10% and 25% being halftoned by the threshold matrix using ($\sigma_1 = 3.3$ and $\sigma_2 = 1.4$) and ($\sigma_1 = 2.7$ and $\sigma_2 = 1.84$).

As expected, the halftones using these two different options look quite similar. The main difference between these two options is that for larger σ_1 the dots start to cluster at lighter tones and the dots are placed more homogeneously for very light tones. As discussed earlier, according to Eq. (5), for $\sigma_1 = 3.3$, the dots start making clusters at 1% and for $\sigma_1 = 2.7$ at 1.5%, which is not a significant difference because the two σ_1 are very close.

Figure 12 shows the RAPS curves for the two halftones in Figs. 11(a) and 11(c), i.e., for $\sigma_1 = 3.3$ and $\sigma_2 = 1.4$. The curves for the other two halftones are very similar and are not displayed here. The halftones show a typical green-noise characteristic. The principal frequencies are also shown in Fig. 12. The principal frequencies were calculated using $f_g = \sqrt{g/M}$ for $0 < g \leq 1/2$, where g is the gray level and M is the size of the clusters.[13] The principal frequencies are thus, equal to $f_g = \sqrt{0.1/7} = 0.12$ and $f_g = \sqrt{0.25/16} = 0.125$ for the two halftones shown in Figs. 11(a) and 11(c), respectively.

4.2.2 *Optimizing the filter*

As discussed earlier, in a second-order FM, the choice of σ_1 and σ_2 is dependent on the application. Based on the size of the clustered dots at a certain gray level and/or at what point the dots start being clustered, one can choose an appropriate pair of σ_1 and σ_2. If there are a number of choices resulting in similar halftones with respect to the average dot size, like the two choices in Sec. 4.2.1, then it could be of interest to compare them with respect to other criteria. One of the criteria is to calculate the area of all clustered dots and then compute the standard deviation of the dot sizes for each patch. A smaller standard deviation means that the clustered dots are more homogeneous with respect to their area/size. This is done by first labeling the halftones and then giving the border dots that would have been connected if the threshold matrices were tiled the same label. By finding the size of each label, a set of dot sizes for each halftone is found and its standard deviation is calculated. For the two choices in Sec. 4.2.1, this standard deviation was calculated for halftones from 1% to 25%, and for all of them it was smaller for ($\sigma_1 = 2.7$ and $\sigma_2 = 1.9$).

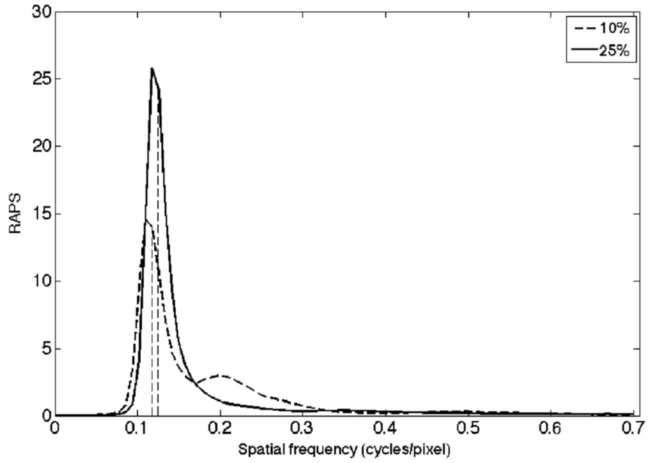

Fig. 12 RAPS curves for two halftones at 10% and 25% using second-order frequency modulated (FM) threshold matrix with $\sigma_1 = 3.3$ and $\sigma_2 = 1.4$. The principal frequencies are shown.

Another criterion is to figure out how homogeneously the clustered dots are placed. A homogeneous distribution of minority cluster pixels (center-to-center) characterizes the green-noise characteristic of the halftones.[13] We study this characteristic by first applying a morphological operation to shrink all clusters in the halftone to a center point and then calculating how homogeneously the center points are placed. The latter can be done the same way it was done for the first-order FM discussed in Sec. 3.2.1 by calculating a set of distances from a dot to its closest dot. The ratio of the standard deviation of this set to its mean can be used as a measure for homogeneousness. For the two choices in Sec. 4.2.1, this ratio was calculated for halftones from 1% to 25%, and for all of them it was smaller for ($\sigma_1 = 2.7$ and $\sigma_2 = 1.9$).

Therefore, when there are a number of options that fulfill the demands of an application, if the very light tones are important, choose the one with larger σ_1 because it gives more well-formed halftones for very light tones (and very dark tones). Otherwise, study them by using the two criteria discussed in this section and choose the one that produces

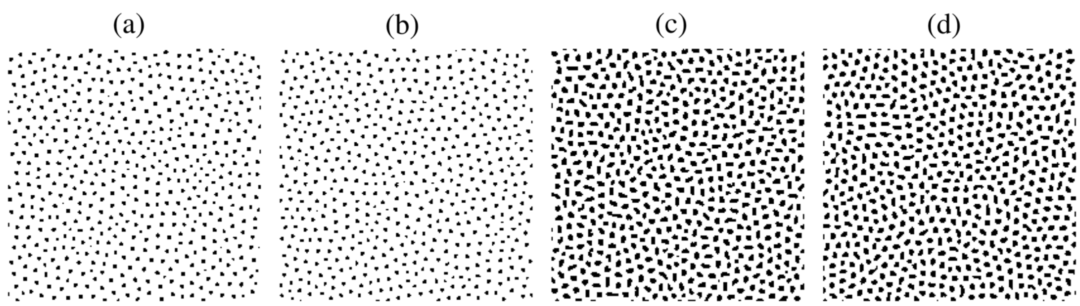

Fig. 11 Two halftones at 10% and 25% using a threshold matrix: (a), (c) $\sigma_1 = 3.3$ and $\sigma_2 = 1.4$, (b), (d) $\sigma_1 = 2.7$ and $\sigma_2 = 1.84$.

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

more homogeneous halftones with regard to the dot size and/or the distribution of the center points.

4.2.3 *Variable sigma*

As discussed in the previous sections by choosing appropriate σ_1 and σ_2, the size of the clustered dots can be adjusted for a specific gray level. When the pair σ_1 and σ_2 are chosen to meet the demand for that specific gray level, the size of the clustered dots at other gray levels will be dependent on this choice and cannot be controlled. In this section, we will discuss the possibility of varying the standard deviations so that the size of the clustered dots can be adjusted for more than one gray level. Let us describe this with an example for an application. For some printing technologies, such as flexography, it is very crucial not to have the dots smaller than a specific size, called the critical dot size, in order to be able to correctly reproduce the highlights (and shadows).[5] Assume as an example that there is a need to have the dots at 4% coverage not smaller than 3×3 in average, meaning a clustered dot area of approximately 9 at 4%. To make this possible, σ_1 must be quite large. The smallest possible σ_1 in this case is 4.4 with $\sigma_2 = 3.7$, see Fig. 13(a), dashed curve. Using this combination will make the average area of the clustered dots around 48 at 25%, see Fig. 13(a) solid curve. For an AM halftone at 1200 dpi, this size corresponds to a screen frequency around 50 lpi at 25%, which is very low. As shown in Fig. 13(a), in order to have smaller clustered dot size than 48 at 25%, σ_2 has to be a variable of the gray level, being 3.7 for tones lighter than 4% and then decreasing. It is also possible to vary both σ_1 and σ_2, but they give the same effect and result in very similar halftones. Therefore, let us now only focus on keeping $\sigma_1 = 4.4$ fixed and varying σ_2 from 3.7 to a certain minimum value. What this certain minimum value should be can be decided based on the application and the wanted average size of the clustered dots at, for example, 25%. For instance, if it is desirable to have a size of 16 at 25%, then σ_2 has to vary from 3.7 to 1.0, see Fig. 13(a) solid curve. Note that $\sigma_2 = 1$ corresponds to an average cluster dot size of 16 at 25%, shown in Fig. 13(a). This big change causes problems. In Fig. 13(b), the average clustered dot area versus tonal value (gray level) ranging from 0% to 25% is shown for three different choices of varying σ_2. For obtaining the dashed curve, σ_2 was 3.7 up to 4% and then was changed to 1.0. One observation is that despite this sudden change it was not possible to get the average area of the clustered dots at 25% reduced to 16, so it is around 20. Another observation is that between 4% and 10%, the average dot size is less than that at 4%, which contradicts the demand of the application that wanted the dot size larger than 9 for tones darker than 4%. It can be concluded that this demand of having the average dot area 9 at 4% and 16 at 25% is not achievable. Therefore, there must be a trade-off between these two demands. In Fig. 13(b), the dotted curve shows the average dot area for a σ_2 of 3.7 until 4% and then gradually decreasing to reach 1.0 at 15%. As seen, the demand of having an average area size of 9 at 4% is met, but the size at 25% is around 23.

The solid curve in Fig. 13(b) shows the average dot area for a smoother variation of σ_2, in which it was 3.7 up to 4% and then gradually decreased to reach 1.0 at 20%. In this case, the average area at 25% is around 29, but the change in the clustered dot areas is smoother than in the other two cases. From now on in this paper, when mentioning a variable sigma for second-order FM we are referring to the latter variation of σ_2.

Figure 14 shows halftones at 4%, 10%, 25%, and 30% halftoned by the generated threshold matrices for second-order FM using different choices of σ_2. In all cases, σ_1 is fixed at 4.4. In the upper and the middle row, $\sigma_2 = 3.7$ and $\sigma_2 = 1.0$ were used, respectively. In the lower row, a variable σ_2 was used. It can be seen, especially in the halftone at 10%, that a variable σ_2 will make the patch less homogeneous in terms of the clustered dot area. This makes sense, because using a large σ_2 in very light tones makes the dots grow very fast. Decreasing σ_2 will force the average dot area not to grow as fast for darker tones, which will make the algorithm place smaller dots in empty spaces, which is clearly seen in the 10% halftone. If a more homogeneous halftone with respect to the clustered dot size is demanded, then σ_2 should vary more smoothly and should also decrease to a larger value than 1.0. This way, the clustered dots will have a more homogeneous size but the average dot size at 25% will be larger than 29.

4.3 *Tiling Effect*

The tiling discussed in Sec. 3.3 has almost the same effect on second-order FM halftoning. Figure 15 (dashed curve) shows the RAPS curve for a 1024×1024 halftone at 10% using a 256×256 threshold matrix with $\sigma_1 = 3.3$ and

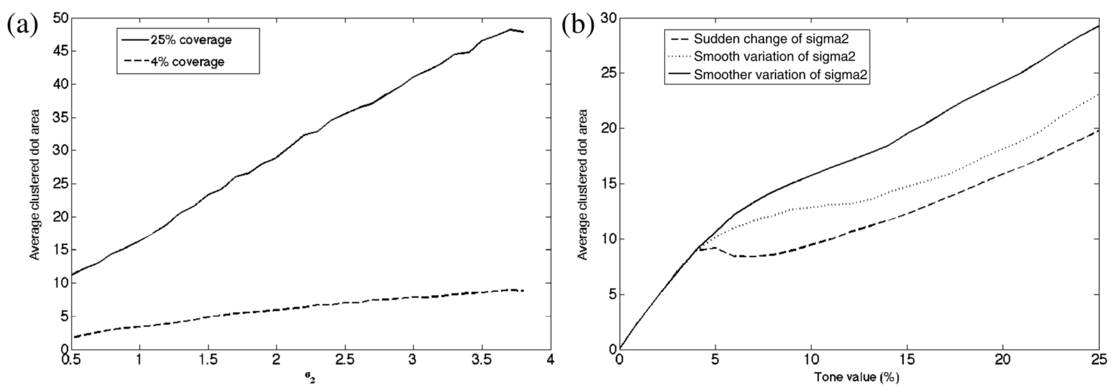

Fig. 13 (a) The average clustered dot areas versus σ_2 using $\sigma_1 = 4.4$ for halftones at 4% and 25% coverage. (b) The average clustered dot area versus tonal value for three different variations of σ_2.

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

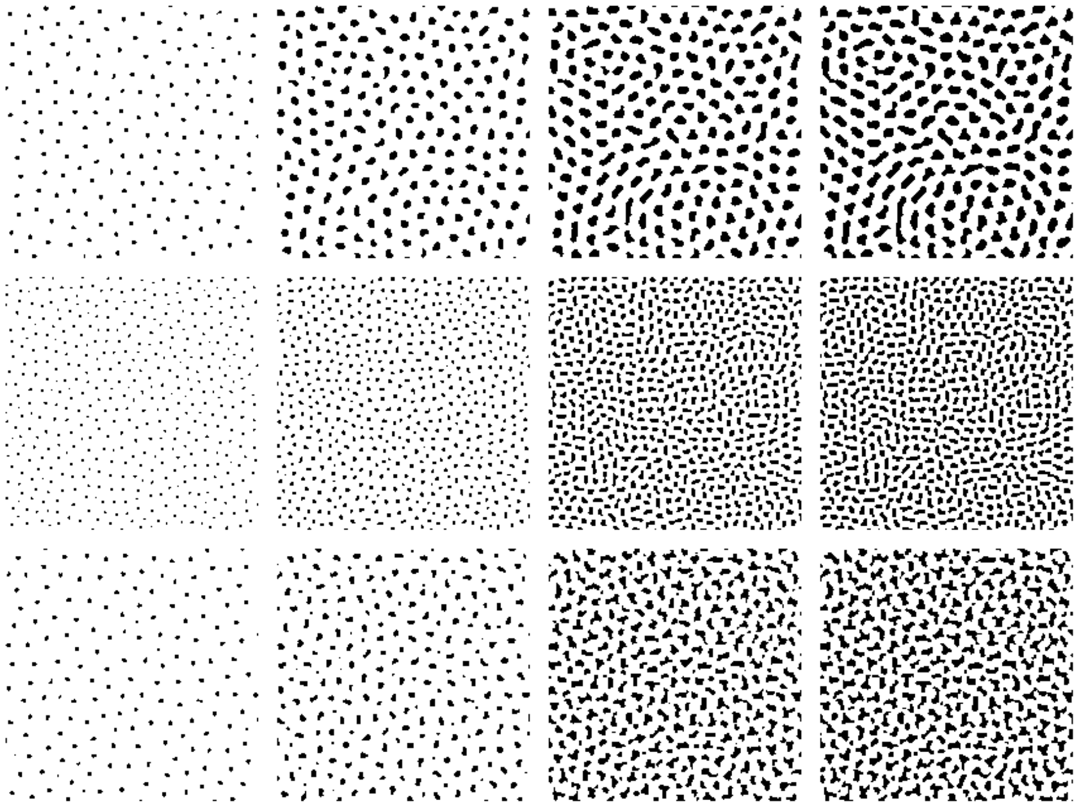

Fig. 14 Halftones at 4%, 10%, 25%, and 30% halftoned by second-order FM threshold matrix: (up) $\sigma_1 = 4.4$ and $\sigma_2 = 3.7$, (middle) $\sigma_1 = 4.4$ and $\sigma_2 = 1.0$, (down) $\sigma_1 = 4.4$ and variable σ_2.

$\sigma_2 = 1.4$. The tiling effect is verified by visually noticing the small oscillations and the spikes in this curve. The second approach proposed in Sec. 3.3 is used to reduce the tiling effect. The solid curve in Fig. 15 shows the RAPS curve for the second-order FM halftone using the 1024×1024 threshold matrix generated by the second approach proposed in Sec. 3.3. These two curves verify that the proposed random tiling reduces the periodic artifacts.

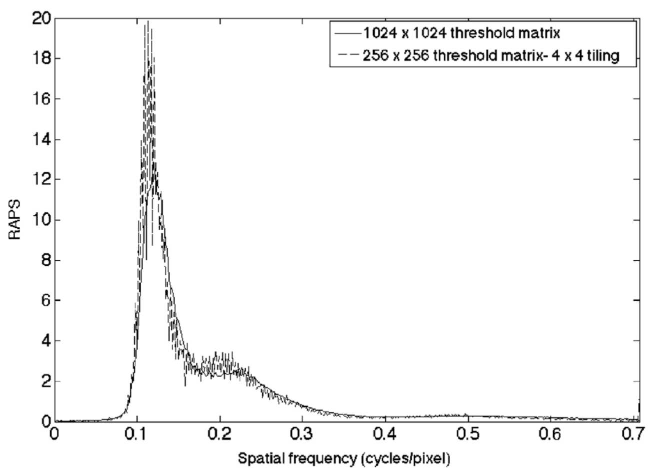

Fig. 15 RAPS curves for 1024×1024 halftones at 10% using 256×256 second-order FM threshold matrix with $\sigma_1 = 3.3$ and $\sigma_2 = 1.4$ and a 1024×1024 threshold matrix generated by the proposed modification to reduce the tiling effect.

4.4 *Dot Shape and Alignment*

So far we have shown and discussed how different choices of σ_1 and σ_2 can change the clustered dot size and to some extent the halftone structures. Here, we give a brief discussion on how the shape of the clustered dots and their alignment can be changed by using appropriate filter/filters. The filter used so far is the one shown in Eq. (3), which is a Gaussian filter subtracted from another Gaussian filter. By this choice, the clustered dots would symmetrically grow in all directions. Using a nonsymmetrical filter and other non-Gaussian filters can produce different halftone structures, dot shapes, and alignment. Let us keep the larger Gaussian filter in Eq. (3) unchanged and illustrate how different choices of the smaller filter can change the dot shape and alignment. In order to make the dots grow faster in one direction, for example, the Y-direction, instead of using the filter in Eq. (3) we can use the one shown in Eq. (6) using, e.g., $k_1 = 1$ and $k_2 > 1$

$$f_1(m,n) = e^{\frac{-1}{2\sigma_1^2}(m^2+n^2)} - e^{\frac{-1}{2\sigma_3^2}\left(\frac{m^2}{k_1}+\frac{n^2}{k_2}\right)}. \tag{6}$$

By varying k_1 and k_2, it is possible to adjust how fast the clustered dots grow in a specific direction. Figures 16(a)–16(c) show a second-order halftone at 10% using $\sigma_1 = 3.3$, $\sigma_3 = 1.4$, and $k_1 = 1$, with $k_2 = 1$ and $k_2 = 2$ and $k_2 = 3$, respectively. Note that $k_1 = k_2 = 1$ makes the filter in Eq. (6) be the same as the one in Eq. (3). It is also possible to make the dots grow faster in other directions by rotating the filter in Eq. (6) by a specific angle. Figure 16(d)

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

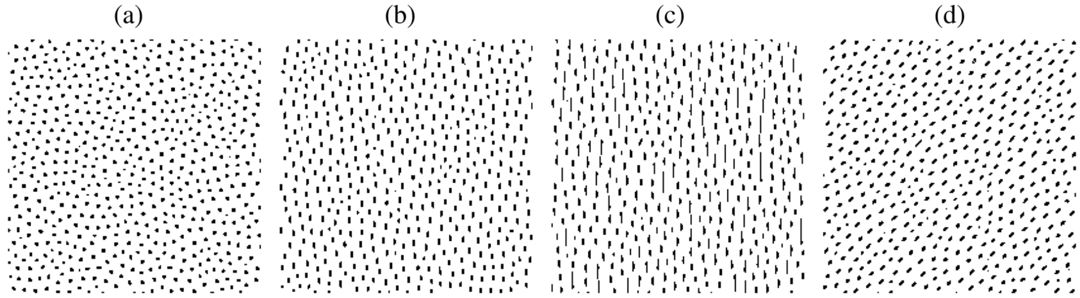

Fig. 16 Second-order halftone at 10% using the filter in Eq. (6) with $\sigma_1 = 3.3$, $\sigma_3 = 1.4$, and $k_1 = 1$: (a) $k_2 = 1$, (b) $k_2 = 2$, (c) $k_2 = 3$, and (d) $k_2 = 2$ and the filter is rotated by 45 deg.

shows the same halftone using the filter in Eq. (6) with $k_1 = 1$ and $k_2 = 2$ rotated clockwise by 45 deg.

Figures 17(a)–17(e) show gray-scale ramps halftoned with a second-order FM threshold matrix using the filter in Eq. (6) with a number of different σ_1, σ_3, k_1, and k_2 to illustrate how different choices of the involved parameters can affect the halftone structure, cluster dot shape, and alignment.

It is also possible to achieve other halftone structures and dot shapes by designing the filters differently. Here,

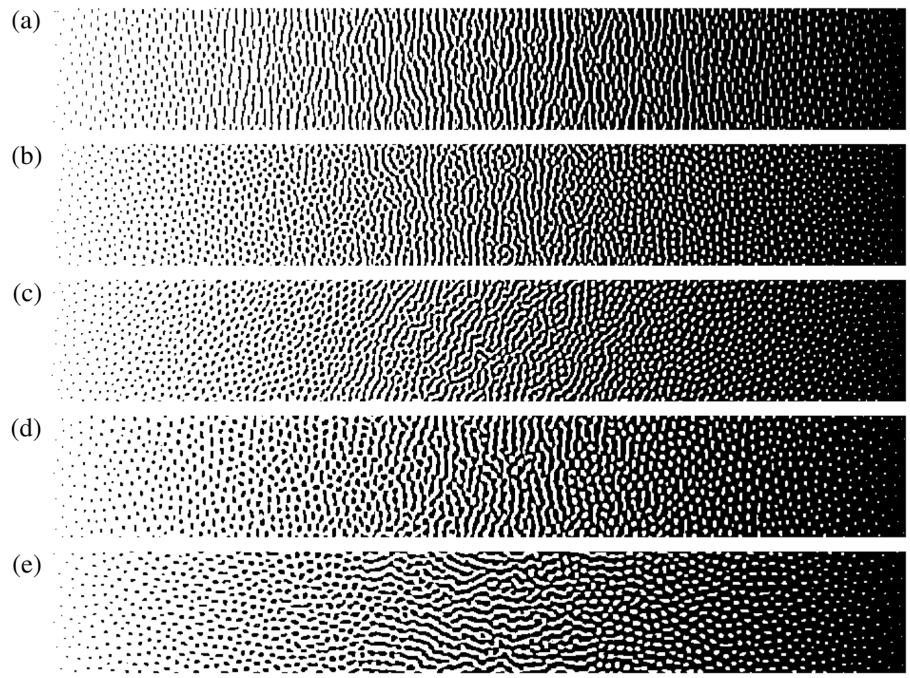

Fig. 17 Gray-scale ramp halftoned with the filter in Eq. (6) using the following parameters: (a) $\sigma_1 = 3.3$, $\sigma_3 = 1.4$, $k_1 = 1$, and $k_2 = 2.5$, (b) $\sigma_1 = 3.3$, $\sigma_3 = 1.4$, $k_1 = 1$, and $k_2 = 1.5$, (c) $\sigma_1 = 3.3$,$\sigma_3 = 1.4$, $k_1 = 1$, and $k_2 = 1.5$, rotated 30 deg, (d) $\sigma_1 = 4.3$, $\sigma_3 = 1.4$, $k_1 = 1$, and $k_2 = 1.5$, and (e) $\sigma_1 = 4.3$, $\sigma_3 = 1.4$, $k_1 = 1.5$ and $k_2 = 1$.

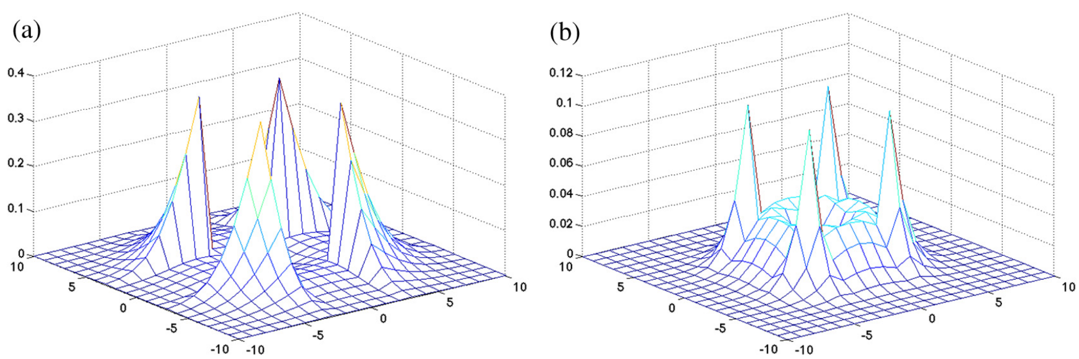

Fig. 18 The filter in Eq. (7) using: (a) $\sigma_1 = 3.0$, $\sigma_3 = 2.9$, and $l = 2$ and (b) $\sigma_1 = 2.0$, $\sigma_3 = 1.9$, and $l = 2$.

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

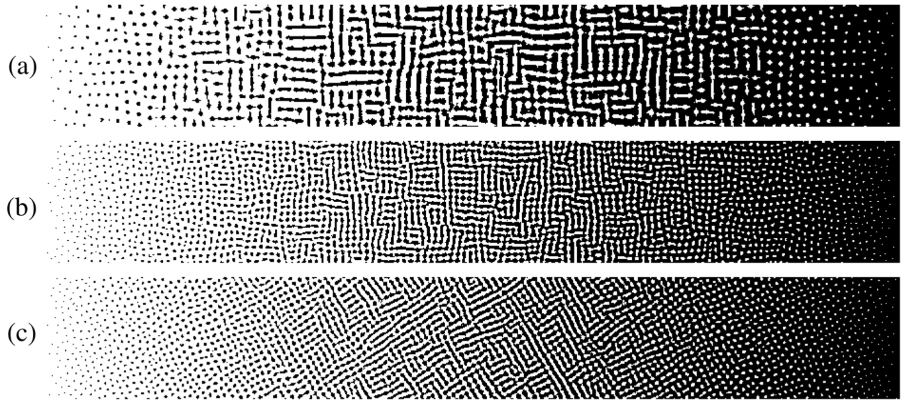

Fig. 19 The gray-scale ramp being halftoned using the filter in Eq. (7) with: (a) $\sigma_1 = 3.0$, $\sigma_3 = 2.9$, and $l = 2$, (b) $\sigma_1 = 2.0$, $\sigma_3 = 1.9$, and $l = 2$, (c) $\sigma_1 = 2.0$, $\sigma_3 = 1.9$, and $l = 2$, rotated 30 deg.

we illustrate another example using the filter shown in Eq. (7):

$$f_2(m,n) = e^{\frac{-1}{2\sigma_1^2}(m^2+n^2)} - g(m,n), \tag{7}$$

where

$$g(m,n) = \begin{cases} e^{\frac{-1}{2\sigma_4^2}(m^2+n^2)}, & \text{if } |m| \le l \text{ or } |n| \le l, \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where l, σ_1, and σ_4 are the variables. If σ_4 is chosen to be slightly smaller than σ_1, the filter in Eq. (7) will produce labyrinth/maze-like halftone structures. Figures 18(a) and

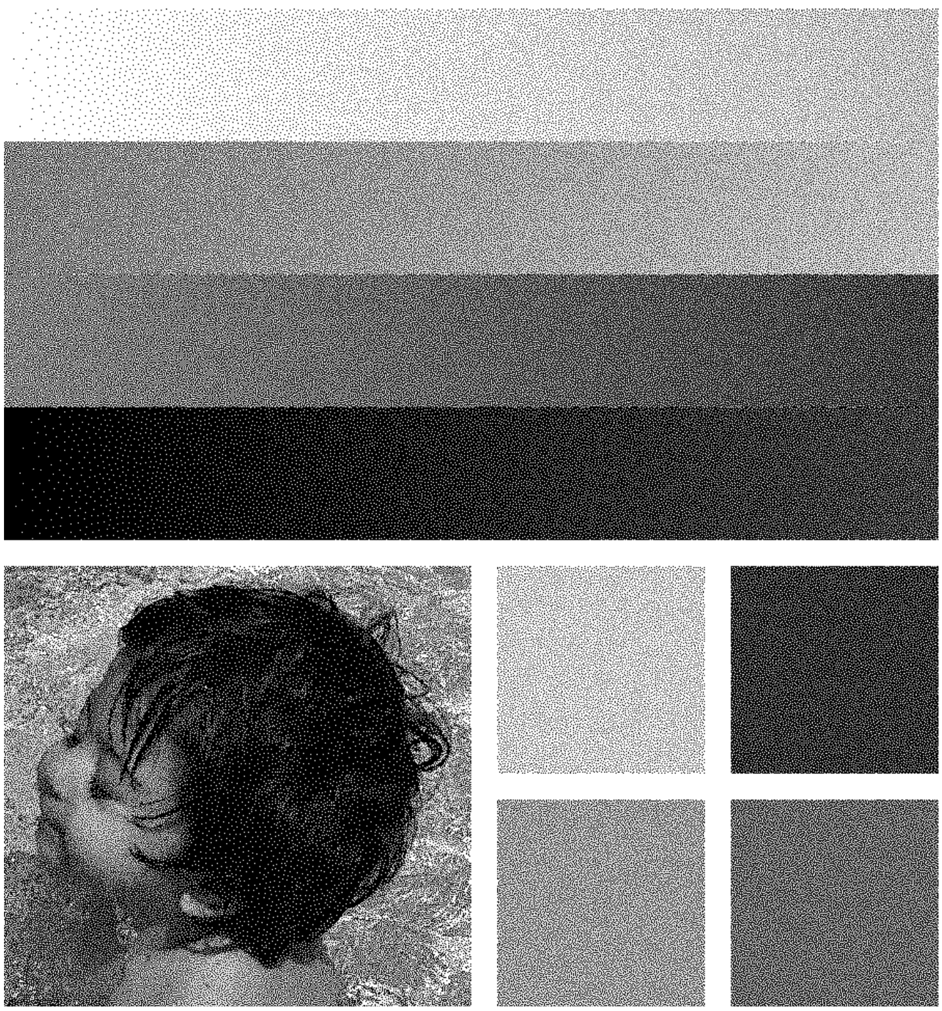

Fig. 20 The test images, a ramp, a regular image, and patches at 20%, 80%, 40%, and 60% are halftoned by first-order FM generated threshold matrix using a variable sigma.

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

18(b) show the filter in Eq. (7) using ($\sigma_1 = 3.0$, $\sigma_3 = 2.9$, and $l = 2$) and ($\sigma_1 = 2.0$, $\sigma_3 = 1.9$, and $l = 2$), respectively. Figures 19(a) and 19(b) show the gray-scale ramp being halftoned by the threshold matrices generated using the filters in Figures 18(a) and 18(b), respectively. Figure 19(c) shows the halftone using the filter in Figure 18(b) being rotated by 30 deg.

By changing the filters and/or adjusting the filter parameters, it is possible to achieve other halftone structures and dot shapes that might be useful for some applications or artistic reproductions.

5 Results

In order to study the results of the proposed approach to generate first-order and second-order FM threshold matrices, a number of test images were halftoned. The chosen test images are a gray-scale ramp, a regular image, and four images of constant gray levels 20%, 40%, 60%, and 80%. The gray-scale ramp is chosen to show how the generated matrices operate in different tonal ranges and how smooth the tonal transitions are. The regular image is chosen to show how they halftone regular images. The constant images are chosen to show the structures of the dots and how they cluster and also show how symmetrical they are in distributing black dots and "white" pixels. That is why the pair 20%, 80% and the pair 40%, 60% were chosen.

Figure 20 shows the test images halftoned by the first-order FM generated threshold matrix using a variable sigma that was described in Sec. 3.2.3. It can be seen that the dots are homogeneously placed in very light and dark tones of the halftoned ramp and the regular image. The tonal transitions are also very smooth, as is seen in the halftoned ramp. It can also be noticed both in the ramp and the constant images that the black dot and "white dot" distributions are symmetrical and similar in the two corresponding sides of the midtone.

Figures 21 and 22 show the test images halftoned by the second-order FM generated threshold matrix using Eq. (3) with ($\sigma_1 = 3.3$, $\sigma_2 = 0.5$) and ($\sigma_1 = 3.3$, $\sigma_2 = 1.0$), respectively. As discussed in Sec. 4, using a larger σ_2 results in bigger clustered dot areas, which are clearly seen by comparing the halftones in Fig. 21 with those in Fig. 22. It can be seen in both figures that the dots are homogeneously placed in very light and dark tones of the halftoned ramp and the regular image. The tonal transitions are very smooth in both halftoned ramps. It can also be noticed both in the

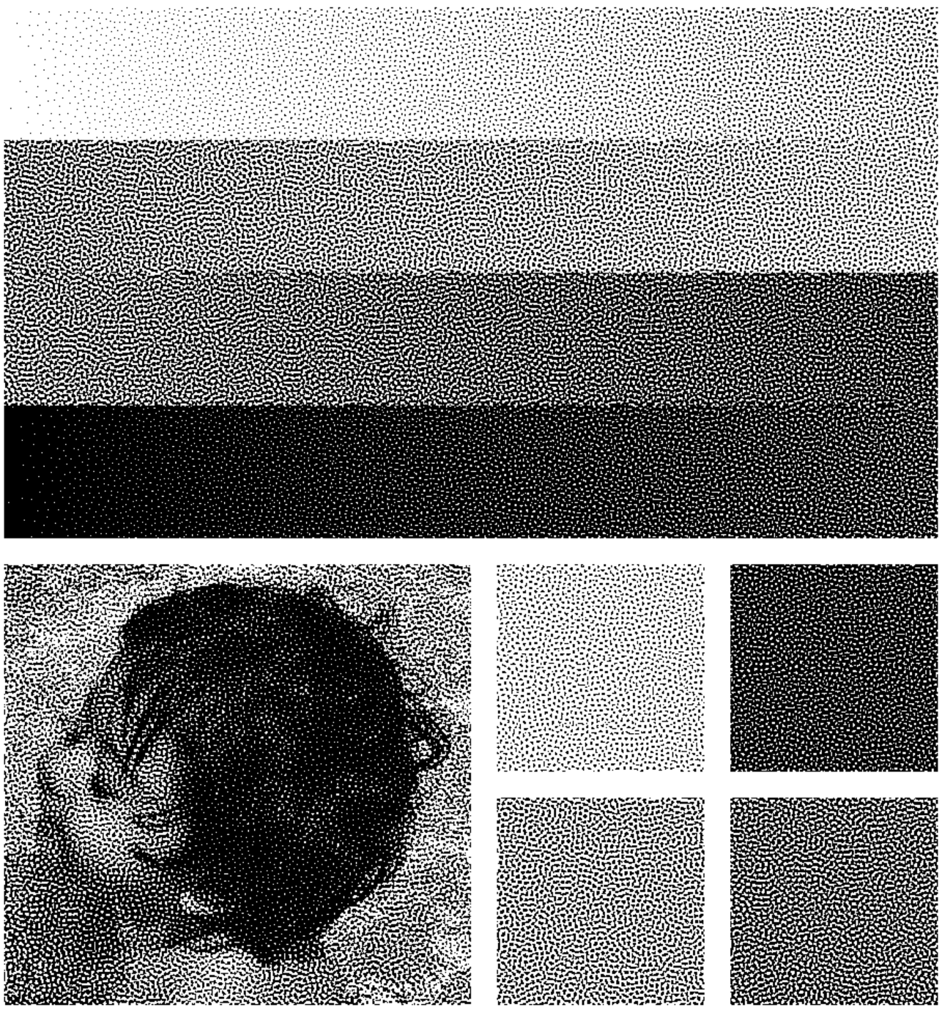

Fig. 21 The test images, a ramp, a regular image, and patches at 20%, 80%, 40%, and 60% are halftoned by second-order FM generated threshold matrix using $\sigma_1 = 3.3$ and $\sigma_2 = 0.5$.

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

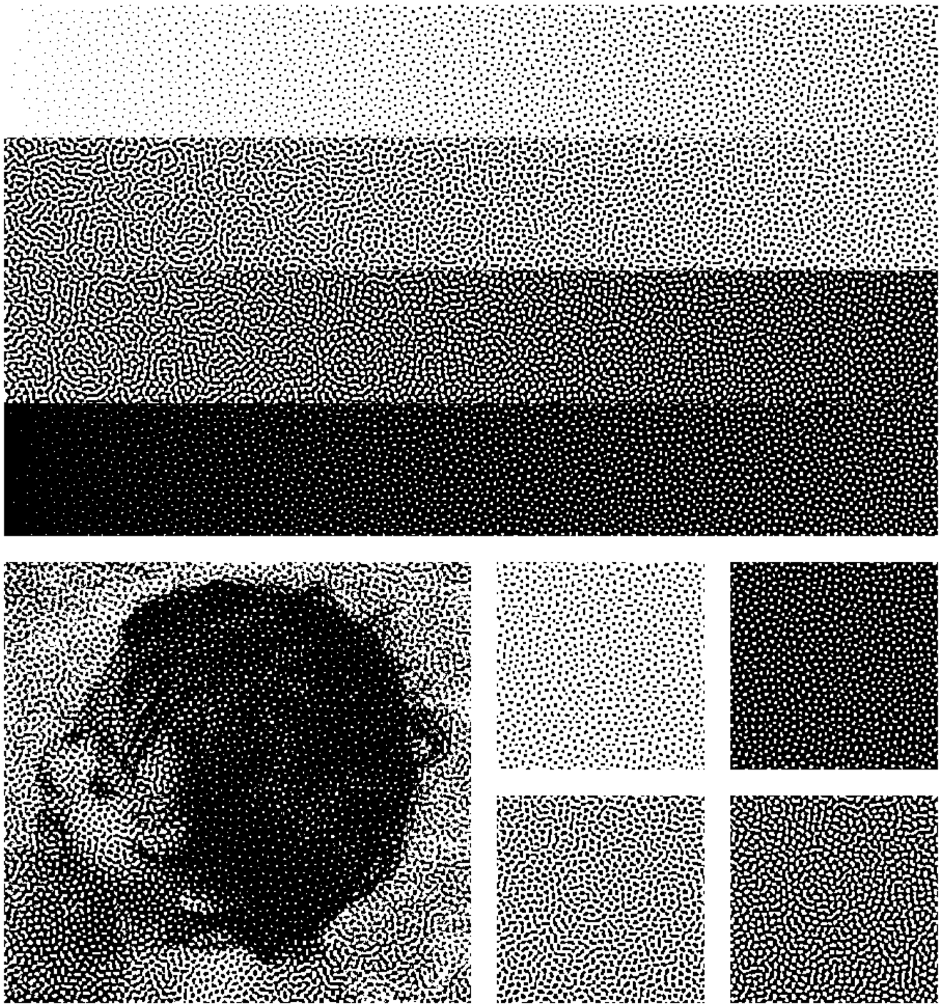

Fig. 22 The test images, a ramp, a regular image, and patches at 20%, 80%, 40%, and 60% are halftoned by second-order FM generated threshold matrix using $\sigma_1 = 3.3$ and $\sigma_2 = 1.0$.

halftoned ramps and the constant images that the black dot (clusters) and "white dot" shapes (voids) are symmetrical and similar in the two corresponding sides of the midtone.

6 Color and Dot-off-Dot Halftoning

As discussed in Sec. 1, periodic clustered halftones usually suffer from moiré. Second-order FM halftones provide a solution because of their stochastic nature of distributing the clustered dots. In this section, we explain how the proposed second-order FM halftoning can be used to halftone color images and how it can be expanded to utilize dot-off-dot structure.

Dot-off-dot structure means to avoid different colorant dots being placed on top of each other if possible. Therefore, if the sum of the coverages of the involved colorants is less than or equal to 100%, the dot overlap can be completely avoided. The advantage of dot-off-dot structure is that they produce smoother halftones and a larger gamut while using less ink compared to the case where the colorants are halftoned independently.[18,20] Another advantage is that the dot-off-dot screen is less sensitive to color shifts due to misregistration between the colorant channels.[18]

Let us first focus on two colorants, e.g., cyan and magenta. In the CMY print, the yellow channel is usually halftoned independent of the other two because of its low contrast.[20] If identical threshold matrices are used for C and M, i.e., $T_m = T_c$, then the dots in C and M channels will be placed precisely at the same positions producing a dot-on-dot structure. If two different threshold matrices T_c and T_m are generated and used for C and M, different colorant dots are placed independent of each other, although the same filters and parameters are used to generate both matrices. Furthermore, if one of the threshold matrices, e.g., T_c, is generated and the other one calculated by $T_m = 1 - T_c$, provided T_c is normalized between 0 and 1, then the overlap between the two colorants will not occur as long as the sum of their coverages does not exceed 100%. Note that in the operation $1 - T_c$, by 1 we mean a matrix of ones the same size as T_c. Figures 23(a) and 23(b) show two identical ramps representing cyan and magenta being halftoned according to the independent and dot-off-dot strategy explained above. The filter in Eq. (3) with $\sigma_1 = 3.3$, $\sigma_2 = 1.0$ has been used. In Fig. 23(b), dot-off-dot structures are maintained up to 50% area coverage per colorant. If there are three colorants involved, e.g., C, M, and Y, then in order

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

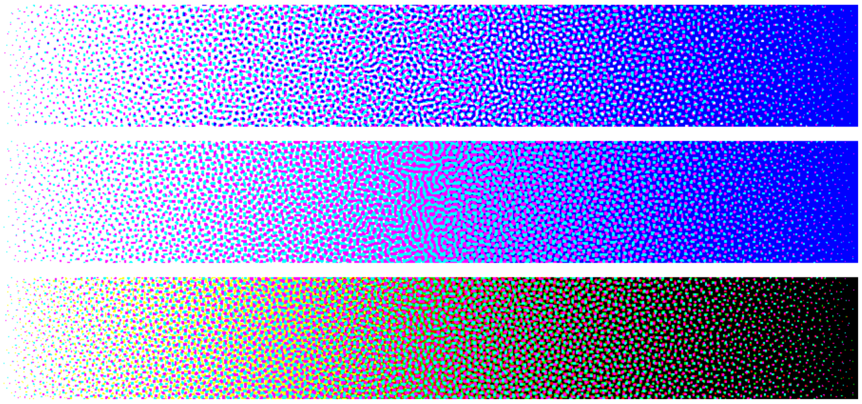

Fig. 23 (a) Identical cyan and magenta ramps are halftoned independently. (b) Identical cyan and magenta ramps are halftoned using dot-off-dot strategy. (c) Identical cyan, magenta, and yellow ramps are halftoned using dot-off-dot structure.

to achieve a dot-off-dot structure we use $T_m = 1 - T_c$ and $T_y = 2|(1/2) - T_c|$. Figure 23(c) shows three identical ramps representing cyan, magenta, and yellow being halftoned according to the dot-off-dot strategy discussed above. The dot-off-dot structures are, therefore, maintained up to 33% area coverage per colorant in this figure. Note that in this dot-off-dot strategy, we only need to generate one threshold matrix and the other two are found as explained. The cyan, magenta, and yellow separations can then be halftoned simultaneously by T_c, T_m, and T_y, respectively, and the colorant values do not need to be processed as a vector.

Although the proposed strategy guarantees dot-off-dot structures, in the highlights and shadows the dots in different channels are not necessarily placed as homogeneously as possible. However, in the proposed method, it is also possible to simultaneously generate T_c, T_m, T_y, etc., in order to both maintain dot-off-dot structure and make the dots in different colorants be placed as homogeneously as possible. The approach would be similar to what has been introduced in Ref. 20. The in-depth study of the latter approach is beyond the scope of the present paper and will be introduced in our future publications.

7 Conclusion

In this paper, we have proposed new methods to generate first-order and second-order FM threshold matrices. Utilizing threshold matrices makes the halftoning process operate very quickly and makes the proposed method feasible to be used in any printing application. It has been explained how the in-built Gaussian filter can be designed to generate well-formed first-order FM halftones. As it is very important to be able to control the size and the shape of the clusters in second-order FM halftones, our method gives the users the possibility to adjust the halftone structure, cluster dot size, shape, and alignment after their need.

The proposed first-order FM method, which was verified to produce well-formed halftones, can be used in print devices that can stably print isolated dots. The proposed second-order FM method can be used as an alternative to periodic clustered-dot screens, in print devices that cannot stably produce the isolated dots to overcome the problem of visible moiré caused by periodic interference. A dot-off-dot structure between three colorant channels is also possible to produce by generating one threshold matrix and calculating the other two matrices based on the first one.

References

1. P. Goyal et al., "Cost function analysis for stochastic clustered-dot halftoning based on direct binary search," *Proc. SPIE* **7866**, 78661A (2011).
2. R. Ulichney, *Digital Halftoning*, MIT Press, Cambridge, Massachusetts (1987).
3. R. Ulichney, "Dithering with blue noise," *Proc. IEEE* **76**, 56–79 (1988).
4. B. E. Cooper and D. L. Lau, "Evaluation of green noise masks for electrophotographic halftoning," *Proc. SPIE* **3959**, 613 (2000).
5. S. Gooran, "Hybrid Halftoning, a useful method for flexography," *J. Imaging Sci. Technol.* **49**(1), 85–95 (2005).
6. P. Zitinski, D. Nyström, and S. Gooran, "Multi-channel printing by orthogonal and non-orthogonal AM halftoning," in *AIC Colour 2013, Proceedings of the 12th Congress* (2013).
7. R. L. Levien, "Photographic image reproduction device using digital halftoning to screen images allowing adjustable coarseness," US Patent No. 5,055,942 (1991).
8. S. Wang, E. A. Bernal, and R. P. Loce, "2nd generation dot-off-dot stochastic halftone," US Patent No. 8,681,383 (2014).
9. D. L. Lau and G. R. Arce, "Method and apparatus for producing halftone images using green-noise masks having adjustable coarseness," US Patent No. 6,493,112 (2002).
10. H. Asai, "Halftone dots, halftone dot forming method and apparatus therefor," U.S. Patent No. 6,989,913 (2006).
11. J. Huang and A. Bhattacharjya, "Method and apparatus for generating dispersed cluster screen," U.S. Patent No. 7,239,429 (2007).
12. R. Levien, "Output dependent feedback in error diffusion halftoning," *Recent Prog. Digital Halftoning*, 106–109 (1994).
13. D. L. Lau, G. R. Arce, and N. C. Gallagher, "Green-noise digital halftoning," *Proc. IEEE* **86**(12), 2424–2444 (1998).
14. P. Li and J. P. Allebach, "Clustered-minority-pixel error diffusion," *J. Opt. Soc. Am. A* **21**(7), 1148–1160 (2004).
15. D. L. Lau, G. R. Arce, and N. C. Gallagher, "Digital halftoning by means of green-noise masks," *J. Opt. Soc. Am. A* **16**(7), 1575–1586 (1999).
16. N. Damera-Venkata and Q. Lin, "AM-FM screen design using donut filters," *Proc. SPIE* **5293**, 469–480 (2004).
17. M. Gupta et al., "Clustered-dot halftoning with direct binary search," *IEEE Trans. Image Proc.* **2**(2), 473–487 (2013).
18. E. A. Bernal et al., "Parametrically controlled stochastically seeded clustered halftones: design strategies and applications," *J. Electron. Imaging* **23**(1), 013010 (2014).
19. S. Wang and R. Loce, "Moiré-Free color halftoning using hexagonal geometry and spot functions," *J. Electron. Imaging* **21**(1), 013017 (2012).
20. S. Gooran, "Dependent color halftoning, better quality with less ink," *J. Imaging Sci. Technol.* **48**(4), 354–362 (2004).
21. Y. F. Liu and J. M. Guo, "New class tiling design for dot-diffused halftoning," *IEEE Trans. Image Process.* **22**(3), 1199–1208 (2013).
22. D. Kacker and J. P. Allebach, "Aperiodic micro screen design using DBS and training," *Proc. SPIE* **3300**, 386–397 (1998).

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms

Sasan Gooran is an associate professor at Linköping University. He received his MS degree in computer science and engineering from Linköping University in 1994 and his PhD degree in media technology from Linköping University in 2001. His current research interests include image and color reproduction, halftoning, digital printing and graphic arts, and multispectral printing.

Björn Kruse is a professor at Linköping University. He received his PhD degree in image processing from Linköping University in 1972. His research interests include color science, image reproduction, digital printing and graphic arts, and computer graphics.

Downloaded From: http://electronicimaging.spiedigitallibrary.org/ on 07/08/2015 Terms of Use: http://spiedl.org/terms