Journal of Imaging Science and Technology® 50(2): 157–167, 2006.

A Novel Hybrid Amplitude Modulated/Frequency Modulated Halftoning Based on Multilevel Halftoning

Sasan Gooran
Department of Science and Technology, Campus Norrköping, Linköping University,
601 74 Norrköping, Sweden
E-mail: sasgo@itn.liu.se

Abstract. *A digital gray scale image generally consists of 256 different gray tones. Printers and image setters normally generate much fewer levels and mostly only two levels. Therefore, in order to be able to display a digital cone-tone image by a multilevel device it has to be transformed into an image with fewer levels. The technique doing this transformation is called multi-level halftoning, and in the case of bilevel devices it is simply called halftoning. In this paper we propose a novel (bilevel) halftoning technique that is based on multilevel halftoning. The proposed method can also be categorized as belonging to hybrid amplitude modulated (AM)/ frequency modulated (FM) techniques. In this method the original digital image is firstly halftoned by a multilevel FM halftoning. Each level in the multilevel halftoned image is then replaced by a halftone table (microcell). An approach for extending any bilevel FM halftoning method to a multilevel method is also presented in this paper. The performance of the proposed method is examined by a number of illustrations where nonmodified error diffusion and our FM method are used. The problem with maze-like artifacts that occur when our FM halftoning, or similar methods such as DBS, are used as the macroscreen is discussed and a simple solution is introduced. An approach for extending the proposed method to be used in situations where the halftone dots cannot be produced smaller than a specific size is also proposed and examined. This modified version of the method can be useful for flexography, where the dots normally cannot be produced smaller than a critical size.* © *2006 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.ImagingSci.Technol.(2006)50:2(157)]

INTRODUCTION

The number of levels a printing device can generate is generally much less than the number of gray tones (or colors) the original image contains. The original digital image should therefore be quantized to fewer levels in order to be displayed by a printing device. This quantization of the original image to fewer levels is called multilevel halftoning. Although printing technologies that can generate more than two levels are becoming more common most of printing devices are still bilevel. Therefore gray scale images are mostly quantized (or halftoned) to two levels and the transformation is simply referred to as halftoning. The most commonly used digital halftoning techniques have been, and probably still are, the so-called amplitude modulated (AM) techniques, where the size of the dots varies while their spacing (frequency) is constant. The darker the tone the bigger the halftone dot. There is also another type of halftoning methods, called frequency modulated (FM), which is the opposite of AM. In these techniques the size of the microdots is constant while their frequency varies. The darker the tone the more the number of microdots. One of the first real FM techniques was invented in the 1970s by Floyd and Steinberg.[1] Due to the fast increase of computer power and introduction of Computer to Plate (CtP) the FM techniques have become more popular and today they are commonly used in many printing devices, e.g., ink jet printers. If the print device is able to produce the distinct dots properly the FM methods are superior in most cases. However, the FM methods generally sever much more from dot gain and also give a grainy impression in the midtones. Therefore, research has been carried out to overcome this problem by introducing hybrid AM/FM halftoning, which uses a FM method in highlight and shadows area and an AM method in the midtones. In flexography, for example, hybrid AM/FM halftoning is a necessity since the method cannot produce the dots sufficiently small.[2] Many AM/FM hybrid halftoning methods proposed in literature have therefore their main focus on its application in flexography.[2–6] The main idea behind all these methods is to use a FM method in the highlights (and shadows) with the smallest producible dot (or hole) and an AM method in the rest of the image. Many approaches and methods for combining these two methods in order to improve the print quality in flexography have been proposed. Some suggest the so-called FM classic halftoning, which is not actually a mixture of AM and FM halftoning. Instead, it is AM halftoning, where part of the highlight dots is left away depending on the dot percentage.[3] Another method to combine AM and FM is to use AM halftoning until the certain critical size is reached. The FM halftoning is then made of screen dots with this critical size. To avoid a clear transition between AM and FM parts both of them are mixed in an intermediate density range.[3] There are also methods that are similar to hybrid halftoning described above without having any mixture of AM and FM in the intermediate density range.[2]

There are also other types of hybrid AM/FM halftoning that produce dot clusters which vary in both their size and frequency depending on the tone.[7] Recently Lin and Allebach introduced a hybrid screen method that produces a stochastic dispersed-dot texture in highlights, and a periodic clustered-dot pattern in midtones.[8] Their experimental results show a very smooth transition from dispersed-dot tex-

Received Nov. 15, 2004; accepted for publication May 11, 2005.
1062-3701/2006/50(2)/157/11/$20.00.

ture to cluster-dot pattern. This method is most suitable for electrographic devices that produce a very noisy rendering of midtone dispersed-dot texture.[8] Very recently He and Bouman proposed a method for optimizing AM/FM halftoning for specific printers.[9] Their approach is based on regularized optimization of print quality as measured from actual printed and scanned halftones. Their test results also show that the AM/FM halftoning produces high quality halftone images on electrophotographic printers with pulse width modulation capability. Creo's Staccato screening method is also a so-called second-order FM, or AM/FM halftoning method, where both the dot size and spacing are not fixed.[10] In FM halftoning (also called first-order FM halftoning), as discussed above, dot size is fixed while spacing is variable. However, the details of Staccato algorithm are not revealed but the test results show an obvious improvement of image quality.

The method that is proposed in this paper can also be classified as being hybrid AM/FM halftoning and the main goal is to present a fast and at the same time satisfying method that can be used for flexography. The basic idea behind this method is very simple and already known.[11] Start with halftoning your digital image to an image with fewer levels by multilevel halftoning. Replace each level with a halftone table that corresponds to that level. For example, using 2×2 halftone tables means that our original image should be multilevel halftoned into five levels, i.e., 0, 0.25, 0.5, 0.75, and 1. When it is done each pixel is replaced by its corresponding halftone table. This is actually what is done when using macroscreens and microscreens in supercell approach, which is also referred to as pulse-surface-area modulation.[12] First, in this paper we present this idea and show some results using a multilevel error diffusion method. In order to use our bilevel FM method in this hybrid algorithm we need to extend it to a multilevel halftoning. Therefore, in this paper we present an approach to extend any bilevel halftoning to a multilevel halftoning. We then use error diffusion and our FM method and expand them to multilevel halftoning methods and illustrate the results. In the next step, we show how this approach can be modified to be suitable for flexography where the dots cannot be smaller than a specific size.

In Ref. 8 the authors show that this way of halftoning, i.e., the supercell approach, will cause a visible maze-like artifact at gray tones close to 8/256 thought the blue noise mask DBS screen is used as the macroscreen. By changing the arrangement of the microdots in the highlight core and simultaneously optimizing the macroscreen and microscreen this problem is overcome. Finally, in the present paper we also discuss this problem and show how the maze-like artifact can be reduced by a simple modification of our FM method and the microscreen. Since in Ref. 8 the authors optimize the dot placement our results are probably not as good as shown in Ref. 8 but our suggested modification does not have any impact on the speed of the process. Since the print resolution is high in flexography we need to deal with quite large images and therefore we need a fast method. On the other hand since the critical dot size normally consists of about 4×4 pixels the maze-like artifacts in the highlight are not as visible as being illustrated in Ref. 8 and the present paper. Therefore, the simple modification suggested in this paper fulfills our purpose.

In this paper it is assumed that gray scale images are scaled between 0 and 1, which represent white and black, respectively. Most of the images are printed at 300 dpi, otherwise the actual print resolution is mentioned.

AM/FM HALFTONING BASED ON MULTILEVEL ERROR DIFFUSION

Error diffusion was invented in the middle of the 1970s by Floyd and Steinberg.[1] At each pixel in the original image a decision is made based on the value of the pixel. The decision of choosing a 1 or 0 is made by thresholding the pixel value with a threshold, which is normally fixed at 0.5. The difference between the pixel value and the output value at this pixel, i.e., the error, is diffused to a number of neighboring pixels that yet have not been processed. To what pixels and how the error is diffused is decided by the error filter. Nonmodified error diffusion halftoning is simple and result in images of fairly good quality. The method, however, suffers from correlated artifacts and directional hysteresis.[12] Figure 1(a) shows our test image being halftoned by nonmodified error diffusion using the Floyd-Steinberg error filter. The correlated artifacts are best seen in the midtones of the image and the directional hysteresis in the highlight and shadow areas. Error diffusion is easily extended to a multilevel halftoning method.[13,14] Instead of thresholding the pixel value with 0.5 and put either a 1 or 0 at the corresponding position in the output one can simply choose the closest level among the available levels. For example assume that only five gray levels are available. At each position one can choose one of the five possibilities, i.e. 0, 0.25, 0.5, 0.75, and 1, that is closest to the pixel value and put it at the same position in the output. The difference between the input and the output at this position is calculated and diffused exactly as before. Now when the image is multilevel halftoned each level can be replaced by a halftone table that represents the same gray tone. To show how this method works we multilevel halftoned our test image by error diffusion to ten levels, i.e., $0, 1/9, \ldots, 8/9$, and 1. Each level is then replaced by its corresponding 3×3 table. The halftone dot within the tables grows as a spiral in our example. Figure 1(b) shows the resulting image. The test image is also halftoned by a conventional table halftoning using the same halftone tables, see Fig. 1(c). This image only consists of ten levels of gray and consequently has a poor quality. Let us point out a number of differences between the images shown in Figs. 1(a) and 1(b). Since the original image is not compensated for dot gain in these illustrations the image shown in Fig. 1(a) looks darker. The reason is that dot gain is bigger in the midtones when using a (first-order) FM halftoning than the case when a hybrid AM/FM halftoning is used. Another difference is that the test image being halftoned in Fig. 1(a) is actually three times bigger in each direction, i.e., nine times bigger in area (pixel $\times$ pixel), than the image being multilevel half-

Figure 1. (a) The test image is halftoned by nonmodified error diffusion using Floyd-Steinberg filer. b) The result after table halftoning the multilevel halftoned image produced by multilevel error diffusion (ten levels, and thus 3×3 halftone tables). (c) The test image is table halftoned by 3×3 halftone tables, thus only ten levels of gray.

toned to make the image shown in Fig. 1(b). Since each level in the multilevel halftoned image has then been replaced by a 3×3 halftone table the images shown in Figs. 1(a) and 1(b) are the same size. This difference can actually mean a lot when it comes to the process speed, especially when iterative halftoning methods, such as DBS or our FM method, are used. Therefore, this type of halftoning means a dramatic increase of the process speed. When it comes to the quality the image shown in Fig. 1(b) suffers much less from the artifacts visible in Fig. 1(a). However, in the highlight and shadow areas of the image shown in Fig. 1(b) the trace of the directional hysteresis are visible. To overcome this problem it is probably better to use a better FM halftoning as the macroscreen. In the next section we will show how any FM method can be used to multilevel halftone gray scale images. This approach is then used to extend our FM method, which is briefly described in this paper, to a multilevel halftoning method.

FROM BILEVEL FM HALFTONING TO MULTILEVEL

As discussed in previous section using a better FM screen than error diffusion as the macroscreen might increase the image quality. Our aim is to use our FM method as the macroscreen (or similarly the multilevel halftoning). The approach to extend our FM method to a multilevel halftoning can actually be utilized for extending any FM method to a multilevel method. First of all let us see the problem of bilevel FM halftoning as the problem of placing a number of black dots (or 1's) on a white background (or 0's). Furthermore, let us describe the approach with a simple example. Assume that we want to have five levels, i.e., 0, 0.25, 0.5, 0.75, and 1. Assume further that we have an image with gray tones less than or equal to 0.25. Hence, it is enough to only use the levels 0 and 0.25 to represent this image and this is what we do in this approach. This can be described as placing 0.25's on a white background. This is similar to our definition of bilevel FM halftoning, the only difference is that here we place 0.25's instead of 1's. If we now multiply all pixel values in this image by 4 and then halftone this image by a bilevel FM halftoning method and then replace the black dots (the 1's) in the resulting image with 0.25's we are done with multilevel halftoning of this image. Assume now that we have an image with gray tones between 0.25 and 0.5. Here we only need the levels 0.25 and 0.5. Halftoning this image can also be described as placing 0.5's on a background containing 0.25's, or vice versa. If we scale the pixel values in this image so that 0.25 and 0.5 become 0 and 1, respectively, problems might occur. If for example we have a gray scale ramp with pixel values varying from 0 to 0.5, then the interval [0–0.25] becomes [0–1], and the interval [0.25–0.5] also becomes [0–1]. This means that at the middle of this ramp our scaled image jumps from 1 to 0, which can cause the bilevel halftoning method not to perform very well in this area. To overcome this problem we simply scale the image so that [0.25–0.5] becomes [1–0]. This means that for gray tones between 0.25 and 0.5 we place 0.25's on a background containing 0.5's. When the bilevel halftoning is performed 0's become 0.5 and 1's become 0.25. With similar reasoning

the interval [0.5–0.75] and [0.75–1] become [0–1] and [1–0], respectively. This approach can be extended to any number of levels. The following pseudocode, which is very simple when written in *MATLAB*, explains this preprocessing approach. By preprocessing we mean the scaling of the image before being halftoned by a bilevel FM halftoning

```
for m=0:1:(n-1) (m varies from 0 to n-1 with a
step of 1)
      if m is even
            where (in>m/n) and [in<=(m+1)/n]
                        out=(n*in-m);
      else (if m is odd)
            where (in>m/n) and [in<=(m+1)/n]
                        out=(m+1-n*in);
      end
end.
```

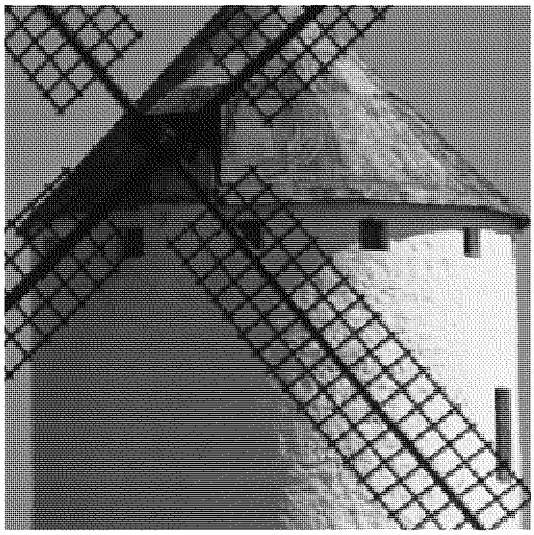
In the code above "*in*" is the original image, "*out*" (the result after preprocessing) is the rescaled version of '*in*' and n is the number of levels minus 1. In order to explain the approach summarized above let the number of levels be 5, i.e., $n=4$, as it was the case in the example above. m varies from 0 to 3 with a step of 1. At the first step when m is 0, and consequently even, in the regions of *in* where the tones vary from $m/n=0$ to $(m+1)/n=0.25$, *out* becomes $n^*in-m=4^*in$, which means that [0–0.25] becomes [0–1]. When m increases to 1, which is odd, then in the regions of *in* where the tones vary from $m/n=0.25$ to $(m+1)/n=0.5$, *out* becomes $m+1-n^*in=2-4^*in$, which means that [0.25–0.5] becomes [1–0]. For $m=2$ the interval [0.5–0.75] becomes [0–1] and finally for $m=3$ the interval [0.75–1] becomes [1–0]. By changing n in the code above the original image can be preprocessed for any number of levels. Now when the preprocessing part is done the scaled image, here *out*, can be halftoned by any bilevel FM halftoning method. After the bilevel halftoning is performed the result, called *half*, should be postprocessed to become a multilevel halftoned image. This is done by the following pseudocode:

```
for m=0:1:(n-1) (m varies from 0 to n-1 with a
step of 1)
      if m is even
            where (in>m/n) and [in<=(m+1)/n]
                  where half==0
                        multi=m/n;
                  where half==1
                        multi=(m+1)/n;
      else (if m is odd)
            where (in>m/n) and [in<=(m+1)/n]
                  where half==1
                        multi=m/n;
                  where half==0
                        multi=(m+1)/n;
      end
end.
```

In the code above *in* and n are the same as before, "*half*" is the result after the preprocessed image *out* has been halftoned by the bilevel FM halftoning method and *multi* is the result after the postprocessing is performed, i.e., the multilevel halftoned image. Let us go back to the previous example and explain the postprocessing code. n is still 4 and m varies from 0 to 3. At the first step when m is 0, and consequently even, in the parts of the original image *in* where the tones vary between $m/n=0$ and $(m+1)/n=0.25$ the 0's in *half* become $m/n=0$'s and the 1's in *half* become $(m+1)/n=0.25$'s in "*multi*." When m is increased to 1, and is consequently odd, in the parts of the original image *in* where the tones vary between $m/n=0.25$ and $(m+1)/n=0.5$ the 0's in *half* become $(m+1)/n=0.5$'s and the 1's in *half* become $m/n=0.25$'s in *multi*. Hence, the code performs exactly the way we explained earlier in this section. To show how this approach works we preprocessed our test image for $n=9$, i.e., ten levels, $0, 1/9, 2/9, \ldots, 8/9$, and 1. Then we halftoned the resulting image by nonmodified error diffusion and our FM method. Then we postprocessed the results in order to get the multilevel halftoned images and then replaced each level by its corresponding 3×3 halftone table. Figures 2(a) and 2(b) show the results for which the bilevel halftoning methods (the macroscreens) are nonmodified error diffusion and our FM method, respectively. The image in Fig. 2(a) is similar to the one already shown in Fig. 1(b). As discussed earlier the trace of the directional hysteresis is visible in the highlights, close to the right wall of the mill, and the shadows, the intersection between the wings. Although the image shown in Fig. 2(b) does not suffer from the same line structure visible in Fig. 2(a) it shows a visible maze-like artifact in the highlight and shadow areas. This problem and how the maze-like artifact can be reduced are discussed later in this paper. The image shown in Fig. 2(c) is the original image being directly halftoned by our FM method. As discussed earlier, the original image used for making the image shown in Fig. 2(c) is nine times bigger in area (pixel × pixel) than the one used for making the image in Fig. 2(b). This means that the number of pixels is also nine times more and therefore the process for making this image is at least nine times slower. Since normally the original images are bigger than our test image an increase of the number of pixels by nine actually makes the process more than nine times slower since the memory access also has a great impact on the process speed. The time for doing the preprocessing, postprocessing, and the final table halftoning is relatively short and consequently negligible.

HYBRID METHOD WITH CRITICAL DOT SIZE

As been discussed earlier, in some printing technologies, such as flexography, the dots cannot be produced smaller than a specific size, i.e., the critical dot size. Here we describe how our proposed approach can be modified to handle this problem. Let us start describing the modified approach with an example. Assume that the microscreen is representing 17 levels, thus halftone tables are 4×4. Assume further that the dots cannot be smaller than 2×2, corresponding to the gray level $4/16=0.25$. In the preprocessing approach described in the previous section we would have transformed the interval



(a)

(b)

(c)

Figure 2. In a and b the proposed preprocessing and postprocessing approaches are applied. The number of levels is 10 and therefore each level is replaced by its corresponding 3×3 table. (a) The nonmodified error diffusion is used as the macroscreen. (b) Our FM method is used as the macroscreen. In (c), the image is halftoned by our original bilevel FM method.

[0, 1/16]–[0, 1]. But in the present case since the dots cannot be smaller than 2×2 we should instead transform [0, 4/16]–[0, 1]. This is simply done by multiplying all the tones in this range by $16/4=4$. In general case it means

```
Where (in<=f/n)
   out=(n/f) * in;
end,
```

where *out*, *n*, and *in* are defined as before and *f* is the number of black microdots in the critical dot. In our example here $n=16$ and $f=4$. For levels darker than 4/16 the preprocessing is almost as before, see the following code:

```
for m=f:1:(n-1) (m varies from f to n-1 with a step of 1)
    If (m is even AND f is odd) OR (m is odd AND f is even)
        where (in>m/n) and [in <=(m+1)/n]
                    out=(n*in-m);
    end
    If (m is odd AND f is odd) OR (m is even AND f is even)
        where (in>m/n) and [in<=(m+1)/n]
                    out=(m+1-n*in);
    end
end.
```

In this code our loop starts from *f*, in our example 4. Depending on whether *f* is odd or even the order of doing the process is changed. For example, in our example, *f* is even and *m*, which starts from *f*, is also even at the first step. That means both *m* and *f* are even at the first step. Therefore, the interval [4/16–5/16] will be transformed by $(m+1-n^{*}in=5-16^{*}in)$ to [1–0], which is exactly what we want since [0, 0.25] became [0, 1]. At the next step $m=5$, and is consequently odd, but *f* is of course still even. That means *m* is odd and *f* is even. Therefore the interval [5/16, 6/16] is transformed by $(n^{*}in-m=16^{*}in-5)$ to [0, 1], which is again exactly what we want since [4/16, 5/16] became [1, 0]. If the critical dot contains an odd number of black microdots the code written above will still work. When the image now is preprocessed it can be halftoned by a bilevel halftoning precisely as before. The post-process should, however, be modified as well.

In the first step in the post-processing all the black dots should be replaced by f/n (in our case 0.25) in the regions where *in* is smaller than or equal to f/n,

```
where (in<=f/n)
   where half==1
      multi =f/n
   where half==0
      multi=0
end,
```

where *multi* is, as before, the image after postprocessing. In our example it means that all black dots in *half* are replaced by 0.25 in regions where *in* is smaller than 0.25. This will

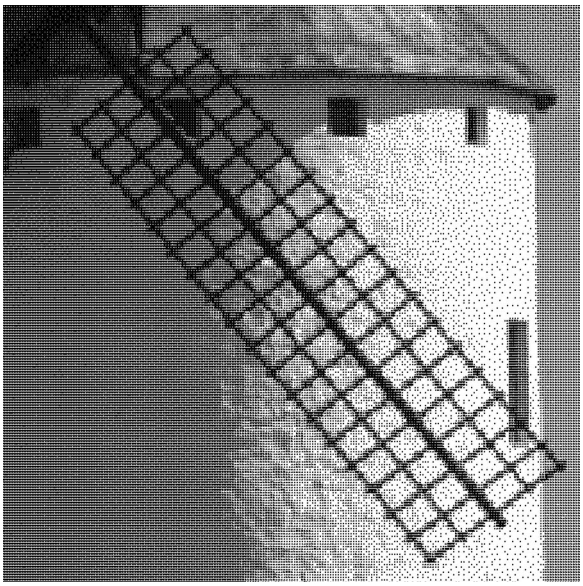


Figure 3. The test image is halftoned by our proposed method. The number of levels is 17. The smallest possible producible halftone dot is 2 $\times$ 2. Only part of the image is shown.

mean that the smallest non-zero level in *multi* is 0.25, which will guarantee that when *multi* is table halftoned later on the halftone dots will not be smaller than 2×2. For the regions where *in* is bigger than 0.25 (or f/n) the postprocessing is modified as shown in the following pseudocode. Here again depending on whether f is odd or even the order of doing the post-processing is changed, exactly as described above for preprocessing

```
for m=f:1:(n-1) (m varies from f to n-1 with a step
of 1)
    If (m is even AND f is odd) OR (m is odd
    AND f is even)
        where (in>m/n) and in<=(m+1)/n
            where half==0
                                multi=m/n;
            where half==1
                                multi=(m+1)/n;
If (m is odd AND f is odd) OR (m is even
AND f is even)
        where (in>m/n) and in<=(m+1)/n
            where half==1
                                multi=m/n;
            where half==0
                                multi=(m+1)/n;
    end
end.
```

Now we use our approach to halftone our test image. We assume that n is 16 and f is 4, which means that the smallest producible dot is assumed to be 2×2. Figure 3 shows our test image being halftoned by using the modified preprocessed and postprocessing approaches. None of the dots in the highlight regions of this image is smaller than 2×2. This approach can easily be extended to be used for the shadows, if there are some restrictions for the holes in the shadows not to be smaller than a specific size. However, in Fig. 3 only the dot size in the highlight areas has been restricted not to be smaller than 2×2. This modified approach can probably be useful for flexography, where AM/FM hybrid halftoning have widely been used in order to overcome the problems in the highlight and shadow regions.[2–6] Observe that the proposed method can handle any critical dot size and dot shape. The dot size can be changed by changing f in the codes above. The shape can be changed by choosing an appropriate halftone table for the level f/n that forms the desirable dot shape.

OUR FM METHOD

In this section we give a short description of the FM method that is used in the proposed hybrid AM/FM halftoning technique. This method is thoroughly described in Ref. 15. In this method the initial binary image is supposed to be empty, that is there is no dot in the initial halftoned image. The problem of halftoning a gray scale image is described as placing a number of dots on this empty initial image so that the final result "resembles" the original image. As the overall impression of lightness/darkness of the image is very important the number of dots to be placed can actually be determined in advance. The sum of the pixel values in the original image rounded to the nearest integer gives us the number of dots that should be placed. Now when we know the number of dots to be placed the question is where to place the "1"s, i.e., the black dots. In this method the dots are placed iteratively in order to decrease the difference between the original image and the halftoned image. Therefore, the first dot is placed at the position of the darkest pixel in the original image. By the darkest pixel we mean the pixel that holds the largest value, i.e., the position of the maximum in the original image. Since the human eye acts as a low-pass filter the difference of the low-pass versions of the two images should actually be decreased. Therefore, the original image should be low-pass filtered prior to the method. Once the first dot is placed the filtered version of the current halftoned image is subtracted from the filtered version of the original image. This process will be referred to as the feedback process in the following. Then the position of the maximum in this modified image is found and the second dot is placed there and the feedback process is performed again. This will continue until the predetermined number of dots is placed and then the final halftoned image is achieved. If we only control the number of dots over the entire image according to our experiments the gray tones are not reproduced well for some images, especially in the highlight and shadow regions. To overcome this problem we choose to control the number of dots to be placed in a number of gray tone regions, and not only over the entire image. Since the highlights and shadows are more sensitive to changes in gray tones we use more control regions in these areas. We choose seven control regions between gray tones 0 and 0.1, i.e., [0, 0.01], [0.01, 0.02], [0.02, 0.03], [0.03, 0.04], [0.04, 0.06], [0.06, 0.08], [0.08, 0.1]. We choose an-

other eight control regions between 0.1 and 0.9, i.e., [0.1, 0.2], [0.2, 0.3], and so on. We choose another seven control regions between 0.9 and 1, [0.9, 0.92], [0.92, 0.94], [0.94, 0.96], [0.96, 0.97], [0.97, 0.98], [0.98, 0.99], [0.99, 1]. Totally we have 22 control regions. Now the numbers of dots to be placed in each of these control regions are determined in advance by the sum of the pixel values in the corresponding region of the original image. The algorithm is now terminated when the predetermined numbers of dots are placed in all these 22 gray tone regions.

When it comes to the size of the filter for the feedback process our experiments showed that an 11×11 Gaussian filter results in halftone patterns with good blue noise characteristic for gray tones between 0.1 and 0.9. However, the dots in the very highlights (and shadows) are not placed as homogeneously as one would expect if an 11×11 filter is used. Therefore, the size of the filter should be chosen dependent on the gray tone. For lighter (tones <0.1) and darker tones (tones >0.9) the size of the filter is therefore gradually increased in order to preserve the blue noise characteristic.[15] In Ref. 15 we describe how the size of the filter should be chosen in different gray tone regions. This FM method has been proved to perform very well for all kinds of images, and especially in the highlights and the shadows.

REDUCING THE MAZE-LIKE ARTIFACT

As mentioned earlier in this paper and also in Ref. 8 our FM halftoning and similar methods cause a visible maze-like artifact in the highlight and shadow areas of the image, see for example Fig. 2(b). As also being pointed out in Ref. 8 when the number of levels is 17 this artifact will occur for gray tones close to $8/256=1/32$. 1/32 is actually half of 1/16 (the lowest nonzero level) and therefore in the preprocessing process is scaled to 0.5. Figure 4(b) illustrates the maze-like artifact for the generated halftone pattern at 8/256. Figure 4(a) shows the halftone pattern after the bilevel halftoning (macroscreen) has been performed and Fig. 4(b) shows the final image. Figs. 4(c) and 4(d) show the spectrum for the images shown in Figs. 4(a) and 4(b), respectively. As can be seen in these images, and also being discussed in Ref. 8, the halftone pattern generated using our macroscreen has a blue noise characteristic while the final halftone pattern does not.[16] There are two reasons for illustrated maze-like artifact. One reason is that having a blue noise characteristic at 0.5 means that the black dots (and the white dots) build a maze structure, see Fig. 4(a). The other reason is that each black dot in Fig. 4(a) has always been replaced by the 4×4 halftone table up to the left in Fig. 4(f). In order to reduce the maze-like artifact we firstly need to change the blue noise characteristic of the pattern shown in Fig. 4(a). Second, we should avoid using the same table for all black dots in Fig. 4(a). In Ref. 8, this artifact is reduced by changing the arrangement of the microdots in the highlight core and simultaneously optimizing the macroscreen and microscreen. This type of optimization is probably time demanding and will slow down the process. Here we try to overcome this problem without doing any complicated and time demanding

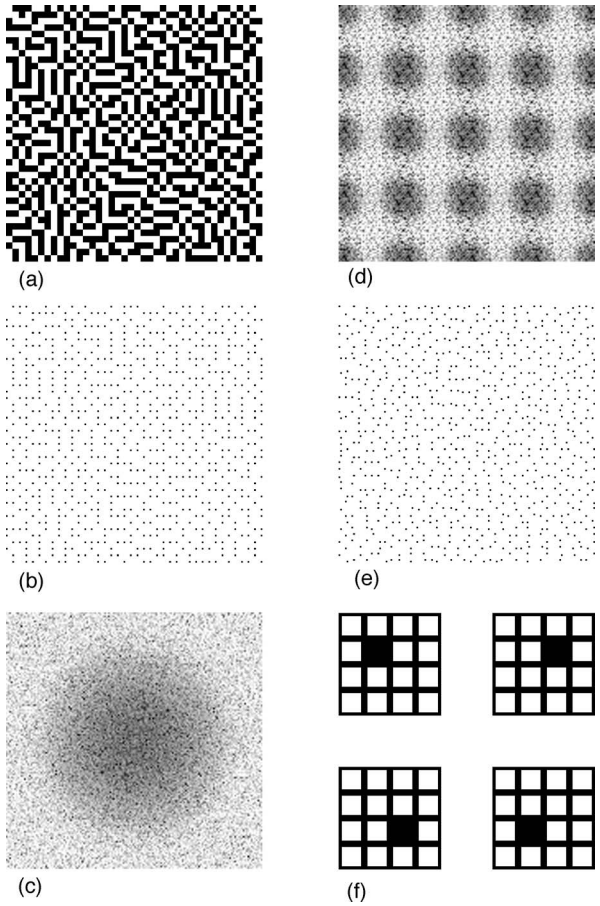


Figure 4. A halftone tint at 8/256 is halftoned. (a) The halftone pattern of the macroscreen (our FM method). (b) The final halftone pattern after table halftoning. Only the halftone table up to the left shown in f is used. The image shows a visible maze like artifact. (c) The spectrum for the image shown in a, which shows a blue noise characteristic. (d) The spectrum for the image shown in b, showing that this halftone pattern does not have a blue noise characteristic. (e) One of the four halftone tables shown in f has randomly been chosen for each pixel. The maze-like artifact is still visible. (f) The four possible halftone tables representing the gray level 1/16. Images are printed at 75 dpi.

computations. The first thinkable and probably simplest way of reducing this artifact would be to randomly choose one of the halftone tables shown in Fig. 4(f) to replace each black dot in Fig. 4(a). This way of choosing the halftone table will result in the image shown in Fig. 4(e). The maze-like artifact is still visible. Observe that the replacement of a level by a halftone table can simply be done by thresholding, see Fig. 5. In our illustrations in this section we assume that the dots will grow as a spiral starting at 1 in the matrix shown in Fig. 5 and ending at 16. If we use this matrix as our threshold matrix (observe that all values are divided by the number of levels, i.e., 17) we can see how each level can easily be replaced by the corresponding table, see again Fig. 5. When we want to choose the halftone table randomly we can build a big mesh of threshold matrices, which consists of these four matrices shown in Fig. 4(f), where they are randomly chosen and tiled together. This big mesh of threshold matrices, or parts of it, can be used whenever needed. That means that this process just needs to be done once and therefore does

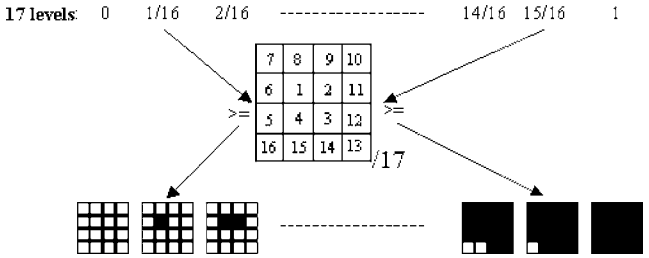


Figure 5. This matrix can be used as the threshold matrix. The black microdots grow as a spiral from 1 to 16 in this matrix. Each level is compared with the values in the matrix. If bigger, a 1 is set at the corresponding position in the result. If smaller, a 0 is set there.

not affect the speed of the halftoning process. However, as being shown in Fig. 4(e), choosing one of the four possible halftone tables randomly did not reduce the maze-like artifact noticeably. As we already discussed we also need to change the blue noise characteristic of the pattern. In Ref. 8 the authors also show that when the problem is solved the halftone pattern of the macroscreen does not any longer have the isotropic blue noise characteristic. As been discussed in the previous section, in our FM method an 11 $\times$11 filter is used for gray tones between 0.1 and 0.9. This choice of filter size will make the final image having a blue noise characteristic.[15] Now if a smaller filter is used we will not have the isotropic blue noise characteristic anymore. Therefore, in the region around 0.5 we use a smaller filter. Our experiments show that using a 5×5 filter in this region will satisfyingly reduce the maze-like artifact. Figure 6(a) shows the halftone pattern after bilevel halftoning using our FM method with a 5×5 filter. The spectrum illustrated in Fig. 6(c) shows that this halftone pattern does not have the isotropic blue noise characteristic. Now we randomly choose one of the halftone tables shown in Fig. 4(f) to replace each black dot in Fig. 6(a). The result and its spectrum are shown in Figs. 6(b) and 6(d), respectively. The maze-like artifact is considerably reduced in the halftone pattern shown in Fig. 6(b) and its spectrum has a fairly good blue noise characteristic.

Before testing our method on a gray scale ramp and the test image we have to discuss how the size of the filter is changed in the modified version of our method. We showed that a 5×5 filter fulfills our purpose quite well for the tint at 8/256, or a tint at 0.5 after preprocessing. Since we mostly use an 11×11 filter in our method a sudden change of its size to 5×5 at gray tones around 0.5 will cause a visible distinction in the final image. Therefore, the size of the filter should gradually be decreased to 5×5 as the tone grows up (or is reduced) to 0.5. Our experiments on grayscale ramp show that the following choices of filter size will reduce the visible artifacts. In the interval [0, 0.2] and [0.8, 1] the filter size is exactly as in our original FM method.[15] In the intervals [0.2, 0.25] and [0.75, 0.8] the filter size is decreased to 9×9. In the intervals [0.25, 0.35] and [0.65, 0.75], a 7×7 filer is used. Finally, in the interval [0.35, 0.65] a 5×5 filter is used, as discussed before. It has to be pointed out that the

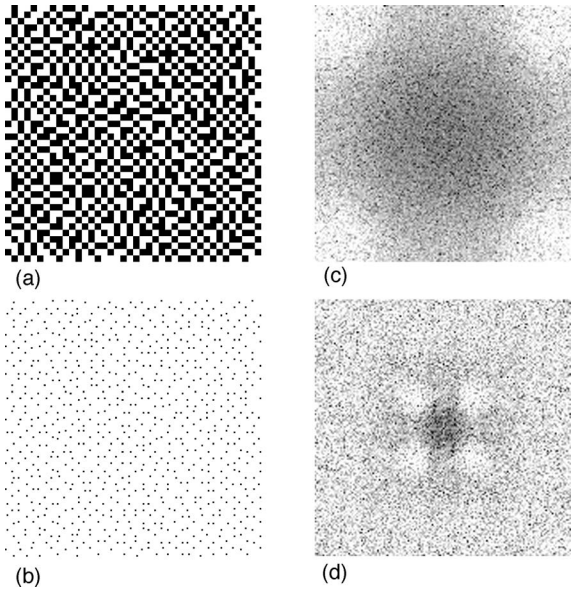


Figure 6. A halftone tint at 8/256 is halftoned by the modified method. (a) The halftone pattern of the macroscreen (our modified FM method). (b) The final halftone pattern after table halftoning. One of the four halftone tables shown in Fig. 4(f) is randomly chosen for each pixel. The image shows an improvement and the maze like artifact is reduced. (c) The spectrum for the image shown in a, which does not have an isotropic blue noise characteristic. (d) The spectrum for the image shown in b, which has a fairly good blue noise characteristic. Images are printed at 75 dpi.

result of the modified method is not so sensitive to how these intervals are chosen. As long as we choose a 5×5 filter in the midtones and then gradually increase the filter size as the tones become lighter or darker it will result in acceptable images. Something important to be mentioned here is that this modification of the method is only used where the original image has tones less than 1/16 in our example. For the gray tones bigger than 1/16 the filter size is exactly the same as in the original FM method although these parts are also scaled between 0 and 1. Let us clarify this by giving an example. Assume again that our original image is called *in*. We again call the image after performing the preprocessing *out*. Now, in those parts of *out* where $in<=1/16$ and $0.35<out<0.65$ we use a 5×5 filter. In those parts where $in>1/16$ we still use our original FM method no matter what *out* is. Figure 7(a) shows the gray scale ramp first being multilevel halftoned by our original FM method to 17 levels and then table halftoned (thresholded) to a bilevel halftoned image. The gray tone of the ramp increases linearly from 0 (bottom right) to 1/16 (bottom left), from 1/16 (middle left) to 0.25 (middle right), and from 0.25 (top right) to 0.5 (top left). As can be seen in this ramp the maze-like artifact is visible in the regions with gray tones close to $8/256 = 1/32$ (in the middle of the ramp in the bottom segment). Figure 7(b) shows the same ramp first being multilevel halftoned by our modified FM method to 17 levels. Then in the regions where the gray tones are less than 1/16 (bottom segment) one of the four tables shown in Fig. 4(f) is randomly chosen for each pixel, as described before. As can be seen the maze-

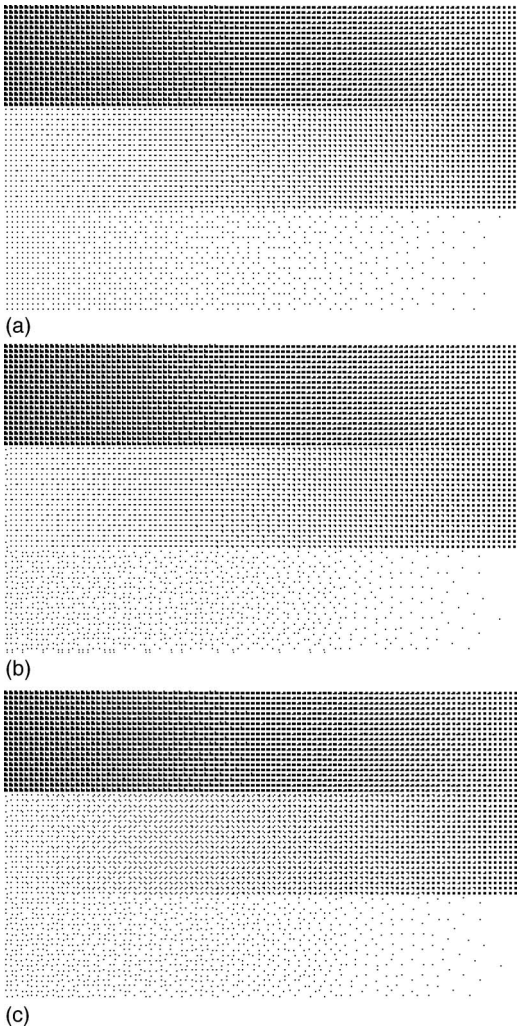


Figure 7. A gray scale ramp is halftoned and printed at 150 dpi. The gray tone increases linearly from 0 (bottom right) to 1/16 (bottom left), from 1/16 (middle left) to 0.25 (middle right), and from 0.25 (top right) to 0.5 (top left). (a) The proposed method, without modification, is used. This result shows visible maze-like artifact in the middle of the bottom segment and visible dot-withdrawal pattern about one fourth of the way from the left side of the bottom segment. (b) The ramp is halftoned using the modified method. Only the bottom segment (gray tones <=1/16) is changed. The transition between stochastic and cluster screen at 1/16 is very smooth. The artifacts visible in a are reduced. (c) The bottom segment is the same as shown in b. In the middle segment (gray tones between 1/16 and 0.25) one of the threshold matrices shown in Fig. 9(a) is randomly chosen for each pixel. The transition from stochastic to cluster pattern at 0.25 is very smooth.

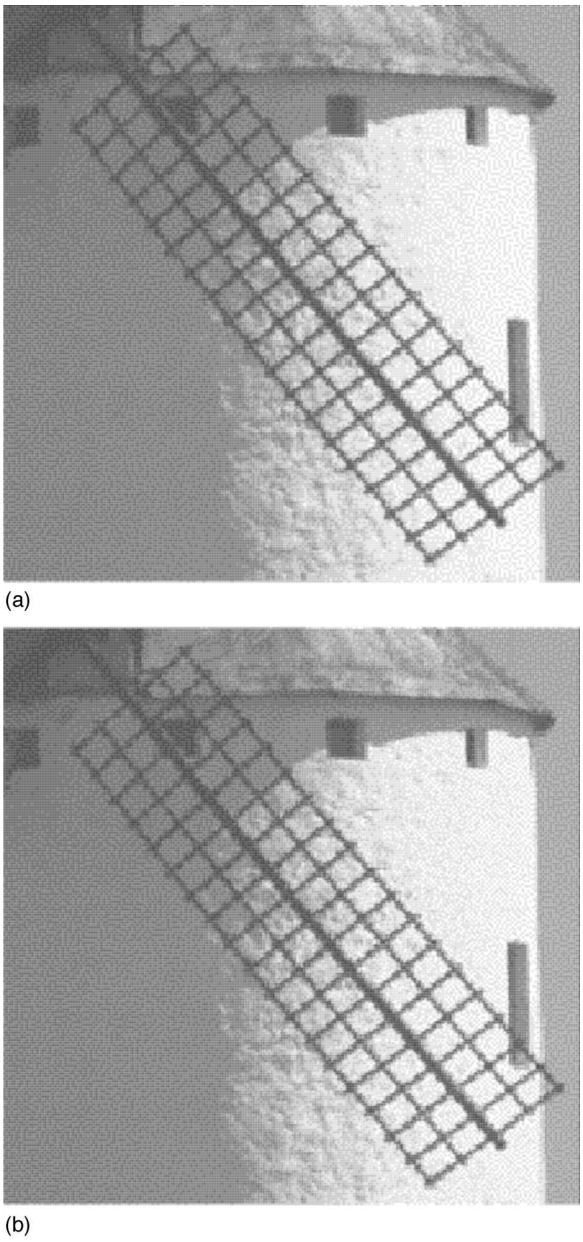


Figure 8. A lightened version of our test image is halftoned by the proposed methods. (a) The original FM method is used. The result shows a visible maze-like artifact in the highlight regions. (b) The modified method for reducing maze-like artifact is used. Maze-like artifact is considerably reduced in the highlight areas. Only parts of the images are shown.

like artifact visible in the middle of the bottom segment in Fig. 7(a) is considerably reduced in Fig. 7(b). The dot-withdrawal pattern seen in Fig. 7(a) about one fourth of the way from the left side of the bottom segment is also less visible in Fig. 7(b). It has to be pointed out here that the transition from stochastic to cluster pattern at 1/16 in this ramp is very smooth. To see how this method works for our test image and in order to be able to illustrate the artifacts in a real image we had to make our test image lighter. Figure 8(a) shows the lightened test image being multilevel halftoned by our original FM method and then table halftoned. Figure 8(b) shows the same image being halftoned by the modified method as described above. As can been seen the maze-like artifact visible in the highlight area (close to the right wall of the mill) in Fig. 8(a) is greatly reduced in Fig. 8(b).

In the ramp shown in Fig. 7(b), as mentioned earlier, the transition from the stochastic texture to a periodic AM pattern was at 1/16. Now we describe how this transition level can be changed to another one, for example 0.25. In our case 0.25 means a 2×2 cluster. In this case, however, we multilevel halftone the image precisely as we did for the ramp shown in Fig. 7(b). For gray tones between 1/16 and

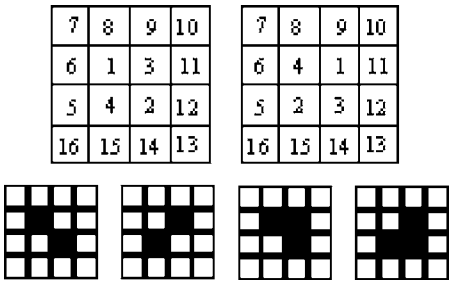


Figure 9. (a) Two threshold matrices that are randomly chosen in the interval [1/16–0.25]. (b) The two possible halftone tables for gray level 2/16. (c) The two possible halftone tables for the gray level 3/16.

0.25 (the middle segment of the ramp) instead of only using the threshold matrix shown in Fig. 5 we randomly choose one of the two threshold matrices (all values should be divided by 17) shown in Fig. 9(a) for each pixel. The gray level 2/16 is therefore represented by one of the tables shown in Fig. 9(b) and the gray level 3/16 is represented by one of the tables shown in Fig. 9(c), see the middle segment in Fig. 7(c). The cluster in the halftone table grows as a spiral for the tones darker than 0.25, precisely like before. The transition from stochastic screen to cluster screen at 0.25 is very smooth.

In order to describe this approach more clearly let us describe this thresholding step by step by following *MATLAB* pseudo code, (only for highlights).

$$mask1 = (in <= 1/16);$$
$$mask2 = (in > 1/16 \& in <= 0.25);$$
$$mask3 = in > 0.25,$$

$$MASK1 = imresize(mask1, 4);$$
$$MASK2 = imresize(mask2, 4);$$
$$MASK3 = imresize(mask3, 4).$$

First of all in this example three different masks have been built to distinguish between the three different gray tone intervals. For example the first row in the code above means "*mask1*" becomes 1 in regions where the original image *in* has gray tones less than or equal to 1/16 and 0 elsewhere. Since the final image is four times bigger in each direction than the original image the masks should become four times bigger, that means each pixel in *mask1*, *mask2*, and *mask3* should be repeated four times in each direction. The function "imresize" does this job. By using these masks we can now build the threshold matrix by

$$tr = MASK1 .* tr1 + MASK2 .* tr2 + MASK3 .* tr3,$$

which means that *tr* is equal to *tr1* where *MASK1* is 1, and equal to *tr2* where *MASK2* is 1 and finally equal to *tr3* where *MASK3* is 1. Observe that in each position only of one these three masks can be 1. *tr1* is the threshold matrix that we want to use for $in <= 1/16$. Therefore *tr1* is a matrix the same size as *MASK1*, *MASK2*, and *MASK3*, and consists of four threshold matrices corresponding to the tables shown in Fig. 4(f), randomly chosen and tiled together. *tr2* is a threshold matrix the same size as *tr1* and consists of the two threshold matrices shown in Figure 9(a), randomly chosen and tiled together. *tr3* is also the same size and only consists of one threshold matrix shown in Fig. 5, being periodically repeated. As can be observed these three threshold matrices are independent of the original image. Therefore, these threshold matrices do not need to be built every time we will halftone an image. Bigger threshold matrices can be built once and then parts of them that are the same size as *MASK1* (and the other two masks) can be selected and used in the above code line. When the threshold matrix *tr* is built then the final image is achieved by the following code:

$$final = imresize(multi, 4) >= tr,$$

which first repeats each pixel value in *multi* four times in each direction and then perform a pixel-by-pixel comparison between the pixel values in this resized image and the threshold matrix.

Finally, it has to be mentioned that the shadow regions are exactly treated as the highlights, the dot texture is simply transformed to a hole texture by taking the complement of the dot textures in the highlights.

DISCUSSION AND CONCLUSIONS

A novel hybrid AM/FM halftoning method that uses our FM method as the macroscreen has been proposed in this paper. A preprocessing and a postprocessing approach have been presented in order to make any bilevel halftoning be used as a multilevel halftoning. These approaches have been used to multilevel halftone a number of images by nonmodified error diffusion and our FM method. The multilevel halftoned image has then been transformed to a bilevel halftoned one by table halftoning or thresholding (microscreen). Using FM methods similar to ours reduce textural artifact that normally are generated when a Bayer screen is used as the macroscreen.[8] However, using our FM method, or DBS screen, will cause maze-like artifact visible in the tones close to 8/256 when the microscreen is 4×4, i.e., 17 levels of gray. In this paper we have shown how this artifact can be reduced by slightly modifying our FM method and randomly selecting the halftone tables in those regions. We have also shown how the preprocessing and postprocessing approaches can be modified to make our method work for applications like flexography, where the halftone dots cannot be produced smaller than a specific size. The illustrations show that the proposed methods result in high quality images. One of the advantages of this AM/FM method over our original FM method is that this method is much faster. Suppose that we want to use our method for a 600 dpi printer and we would like the screen frequency to be 150 lpi. Assume that we want our printed image to be the same size as an A4 paper, i.e., about 8×11 inches. If we now want to use our original halftoning method then we have to halftone a 4800×6600

pixels image, which will take long time even though the latest version of our FM method is used. The latest version of our FM method, which is relatively fast for being an iterative halftoning method, run on Pentium (R) 4 CPU, 3.00 GHz and 1.00 GB random access memory (RAM), takes about 90 s for a 4096×4096 image.[17] But the image mentioned in this example is much bigger and probably will take about 3 min, if the RAM capacity allows for performing our halftoning method on such a big image. But if the image is being halftoned by the proposed hybrid AM/FM method the image that should be FM halftoned is only 1200×1650, which will only take about 6 s using the same computer. The preprocessing and postprocessing are usually very fast. The table halftoning, or the thresholding, will not either take any considerable time if the threshold matrices are built in advance, as was discussed before. Therefore, halftoning this image will not take more than about 7–8 s, which is a very reasonable time for such a halftoning method with fairy good and acceptable results.

Our future goal is to study the possibility of using this method for flexography. Compared to our previous hybrid method presented in Ref. 2 this method is much faster. However, it remains to test this method for real prints in flexography. Another future goal is to study how this method can be extended to be used for color images. If this method is applied to the color channels of the color image independently, in the highlights (and the shadows) the color dots in different channels will be placed randomly in relation to each other. In other parts, the color halftone dots will be printed on top of each other. Therefore, a similar technique as presented in Ref. 15, which halftones the color channels dependently, will probably increase the print quality. Another possibility for achieving better results might be to choose appropriate and different halftone tables for each channel. It is also very interesting to investigate how the extended method can handle the problem with Moiré. Since the hybrid method behaves like a AM method in the midtones we probably need to use different angles for different colors in those regions.

ACKNOWLEDGMENT

The author would like to thank Professor Björn Kruse for his valuable comments.

REFERENCES

[1] R. W. Floyd and L. Steinberg, "Adaptive algorithm for spatial grey scale", *Proc. International Symposium Digest of Technology* (1975) p. 36.
[2] S. Gooran, "Hybrid halftoning, a useful method for flexography", J. Imaging Sci. Technol. (to be published).
[3] Barco Graphics, SambaFlex Screens, *User Manual*, Partcode: 4913710.
[4] R. Bartels, "Reducing patterns in the FM part of tile-based hybrid screens", *Proc. IS&T/SID's Fourth Color Imaging Conference* (IS&T, Springfield, VA, 2002) pp. 238–245.
[5] D. Blondal, "The lithographic impact of microdot halftone screening", *Proc. TAGA* 607 (2003).
[6] R. Bartels, "Hybrid screening, what it will do", *Proc. TAGA* 647 (2003).
[7] D. L. Lau and R. A. Gonzalo, *Modern Digital Halftoning* (Marcel Dekker, New York, 2001).
[8] G. Lin and J. P. Allebach, "Generating stochastic dispersed and periodic clustered textures using a composite hybrid screen", Proc. SPIE **5293**, 413 (2004).
[9] Z. He and Ch. A. Bouman, "AM/FM halftoning: digital halftoning through simultaneous modulation of dot size and dot density", J. Electron. Imaging **13**, 286 (2004).
[10] http://www.visiongraphics-inc.com/technologies/staccato.html.
[11] E. A. Sandler, D. A. Gusev, and G. Y. Milman, "Hybrid algorithms for digital halftoning and their application to medical imaging", Comput. Graph. **21**, 69–78 (1997).
[12] R. A. Ulichney, *Digital Halftoning* (MIT Press, Cambridge, MA, 1987).
[13] M. Broja, K. Michalowski, and O. Bryngdahl, "Error-diffusion concept for multilevel quantization", Opt. Commun. **79**, 280 (1990).
[14] J. R. Goldschneider, E. A. Riskin, and P. W. Wong, "Embedded multilevel error diffusion", IEEE Trans. Image Process. **6**, 956 (1997).
[15] S. Gooran, "Dependent color halftoning, better quality with less ink", J. Imaging Sci. Technol. **48**(4), 352 (2004).
[16] R. A. Ulichney, "Dithering using blue noise, selected papers on digital halftoning", SPIE Milestone Series **MS 154**, 37 (1988).
[17] D. Byström, "Implementing an iterative halftoning algorithm, Diploma Work LITH-ITN-MT-EX—04/046-SE", Linköping University, Sweden, (2004).