Dependent Color Halftoning: Better Quality with Less Ink

Sasan Gooran

Dept. of Science and Technology, Linköping University, Campus Norrköping, Norrköping, Sweden

When a color image is halftoned its color channels are normally halftoned independently. The dots in different channels are placed independent of each other and consequently the final result may not be of high quality even if a well performing monochromatic halftoning method has been used. In this article we propose a method that halftones the channels of the color image in a context dependent manner. Since the yellow ink on a white paper is hardly visible, only cyan and magenta separations need to be halftoned dependently. We also show that dependent color halftoning not only increases the halftone image quality but also decreases the amount of ink needed to reproduce different colors.

Journal of Imaging Science and Technology 48: 354–362 (2004)

Introduction

Due to the rapid increase of computer power many complicated but efficient halftoning methods have been proposed in the last few decades.[1–6] Halftoning a color image introduces new challenges and difficulties. Since color printers and printing presses normally use three or four colors the halftoning method should be applied to three or four images, each representing a color channel of the original color image. Normally these channels are halftoned independently. Even if the dots in each channel are placed homogeneously there is no guarantee that the dots in different channels are placed homogeneously in relation to each other.

In literature a number of interesting papers have been published in recent years that propose new methods that look at the color channels together. In Ref. [7] the authors discuss extensions of error diffusion algorithms in which they look at the color separations together and select the closest output color. This results in more accurate color reproduction and less ink consumption. The final results however suffer from the typical artifacts related to error diffusion methods due to the directional propagation of the method. In Ref. [8] the authors propose a method based on the vector error diffusion method. The basic idea behind this method is similar to the previous one. Lee and Allebach propose a halftoning method based on Direct Binary Search (DBS) that controls the quality of each colorant texture separately along with the total dot distribution.[9] In this method the dot-on-dot printing is avoided as much as possible which results in color images of higher quality. The optimization problem is solved by swap-only DBS, which is normally time demanding. The authors however do not mention anything about the speed of the algorithm. Something that is not taken up in their article,[9] and will be discussed in this article, is the fact that some color shifts will always occur when the dot-off-dot printing strategy is used, compared to the case when the color channels are halftoned independently.

In this article we propose another dependent color halftoning method which also controls the dot placement over the entire color image and avoid the dot-on-dot printing as much as possible. In the following section we give a description of this method. Then we show that dot-off-dot printing needs less ink to produce the same color as when the dots are randomly overlapped, which is actually the case when the color channels are halftoned independently. We subsequently show some results of halftoning color images by the proposed method to demonstrate that the proposed dependent method results in images of higher quality, compared to the case where the channels are halftoned independently. We also show some results of halftoning a color image by Vector Error Diffusion. We also give a short description about how to extend the method to be used for CMYK images. Finally our future work is discussed, and we provide some discussion and final conclusions.

Our Dependent Color Halftoning

Before describing the method let us mention a number of assumptions. In this method we assume that “0” and “1” represent no dot (or white dot) and full tone dot, respectively. The original image is scaled between 0 and

Original manuscript received December 9, 2002

e-mail: sasgo@itn.liu.se; phone: +46 - (0)11 - 36 32 77; fax: +46 - (0)11 - 36 32 70

Supplemental Materials—Figures 6 through 9 are available in color on the IS&T website, www.imaging.org, for a period of no less than 2 years from the date of publication.

1. The color image is supposed to consist of three channels, Cyan, Magenta and Yellow, which will be denoted by C, M and Y. This algorithm can be applied to four color CMYK, by letting, at a given pixel, a black dot replace cyan, magenta and yellow dots if and only if all three are present there, see below.

Monochromatic Halftoning 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 grayscale image can easily be 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. Simply, the original image and the final halftoned image should have the same spatially averaged mean value. Therefore, 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. By using this criterion we actually minimize the dc component of the quantization noise, i.e., the Fourier transform of the difference between the final halftoned image and the original at frequency (0, 0).

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 addressed as the feedback process in this article. 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 are placed and then the final halftoned image is achieved. The low-pass filter (human visual system) we use is a Gaussian filter with standard deviation 1.3 truncated to 11 × 11.

The algorithm is summarized in Fig. 1. Note that in Fig. 1 the original image is first filtered by f, which is actually the same 11 × 11 filter. By changing this filter, we can change the sharpness of the final image. However, using the same filter as the one that is used within the method makes the final image be as sharp as the original one, see Figs. 2 and 3. In Fig. 1 we see that some noise has been added to the image. Since the program is based on searching for a maximum, in a constant image the first or the last pixel will be chosen as the maxima and this can cause the final image to become highly structured. To avoid that we add a very small amount of noise to the image to make the pixels have slightly different values even if the original pixels hold the same value. However, the amount of added noise must be so small that it does not at all have any impact on the original image.

According to our experiments the results are satisfying for most images but by a further investigation we saw that the gray tones are not reproduced well for some

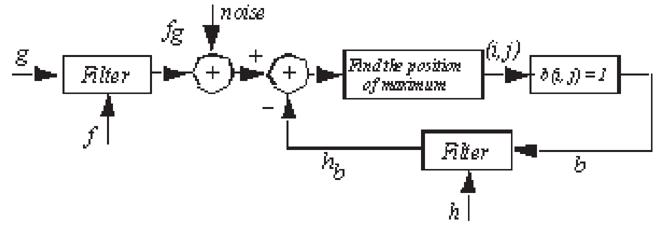


Figure 1. The proposed monochromatic halftoning method for grayscale images.

images, especially in 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, but not 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 another 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.

Now, the method works quite well for almost all kinds of images. However, the dots in the extreme highlights are not placed as homogeneously as one would expect. The reason is that the 11 × 11 filter is not big enough for these regions. Assume that we have a $n \times n$ constant image with a pixel value of p $(0 < p < 1)$. The number of dots to be placed is therefore pn^2. Assume further that the average distance between the dots is a. To have the dots placed as homogeneously as possible we should have

$$a^2 pn^2 = n^2 \tag{1}$$

which gives us the average distance between the dots as

$$a = \sqrt{1/p} \tag{2}$$

Therefore, the size of the filter in the very light parts of the image should be calculated by Eq. (2). Now, if for example $p = 0.04$ then we have $a = 5$ which gives us an 11 × 11 filter, which is exactly the filter we have used; $p = 0.01$ needs a 21 × 21 filter. In our method we thus use an 11 × 11 filter in areas where the gray tonal level is above 0.04 (and less than 0.96). For the lighter tones we use Eq. (2) to find the distance between the dots, i.e., a, and consequently the size of the filter, which is $(2a + 1 \times 2a + 1)$. Note that a in Eq. (2) has to be rounded to the nearest integer.

Figures 2 and 3 show two grayscale images halftoned by the proposed method. In Fig. 2 the filter f is 11 × 11 and thus the result is as sharp as the original. In Fig. 3, the filter f is 7 × 7, i.e., a smaller low-pass filter. As can be seen the image in Fig. 3 is sharper than the one in Fig. 2. In both images one can specially observe that the dots in the highlights of the image are homogeneously placed.

Figure 2. The image is halftoned with the proposed monochromatic halftoning method. The filter f is 11 × 11.

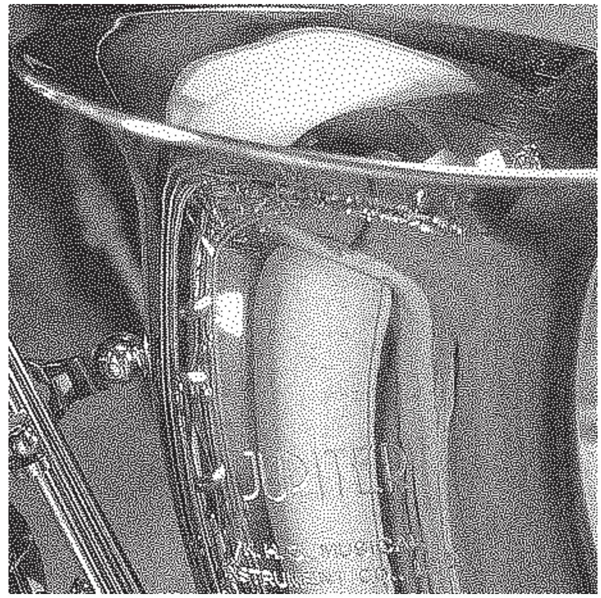


Figure 3. The image is halftoned with the proposed monochromatic halftoning method. The filter f is 7 × 7.

The Color Halftoning Method

Due to the characteristics of this algorithm it is easily extended to a dependent color halftoning method. Since the yellow ink on a white paper is much less visible than cyan and magenta we choose to halftone the Y channel independent of the two others without having a noticeable change in the result, see also our proposal for future work, below, for further discussion.

The strategy here is to avoid the dot-on-dot printing as much as possible and place the C and M dots homogeneously over the entire color image. Let c and m denote the coverage in the C and the M color channels, respectively. We start with the case $(c + m) \leq 1$, which means that dot-on-dot printing can completely be avoided.

As in the monochromatic case the numbers of dots to be placed in different tone regions of the C and M separations are calculated in advance. Then, the algorithm starts with finding the maximum pixel value in the C and M channels. Suppose that the maximum was found in the C channel. Now the feedback process is performed as before, but this time to both the C and M channels. Since the maximum was found in the C channel the filter for the C channel is chosen exactly as in the monochrome case. Finding the filter that should be used in the M channel needs more investigation. Suppose that $c > 0.2$, then, according to Eq. (2), the average distance between the C dots is less than 2.23 (rounded to 2). This means that the distance between a C dot and a M dot should be unity. Therefore, it is enough if only a 1 × 1 filter is used in the M channel. For $c < 0.2$ we use Eq. (2) to find a and use $a/2$ to decide the size of the filter that should be used in the M channel. The same strategy is used if the maximum is found in the M channel. After these feedback processes are performed the algorithm seeks for the next largest value in the modified C and M channels and places the next dot at the corresponding position of the corresponding channel. The feedback process is performed again. This process continues until the pre-determined numbers of dots are placed in the 22 tonal regions of each channel.

When $(c + m) > 1$ we cannot completely avoid dot-on-dot printing, therefore there will be a number of blue dots. The coverage of blue dots is actually $b = c + m - 1$. In this case instead of halftoning C and M we halftone C' and M' with coverage $c' = c - b = 1 - m$ and $m' = m - b = 1 - c$, respectively. Since $(c' + m') = (1 - b) < 1$, C' and M' can be halftoned as described before. A portion, $b \times 100\%$, of the resulting image is now empty and is accordingly filled with blue. Let us give an example. Assume that $c = 70\%$ and $m = 60\%$. We will have 70 + 60 – 100 = 30% blue. In this case we halftone C' and M' with 40% and 30% coverage respectively. Then 30% of the result is empty (white) and will therefore be filled with blue.

For a real image with the channels C and M the approach is as follows. $C[m]$ and $M[m]$ denote the amount of cyan and magenta in position m of the image, respectively.

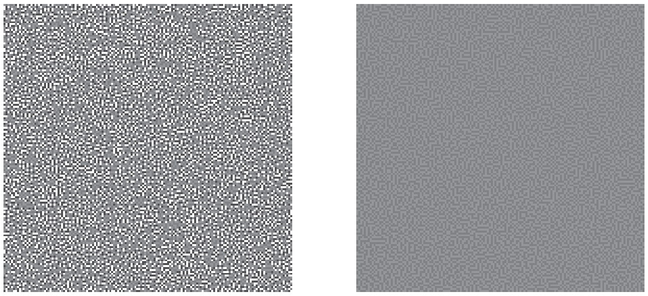
$$b[m] = C[m] + M[m] - 1 \tag{3}$$

$$C'[m] = \begin{cases} C[m], & \text{where,} \quad b[m] \leq 0 \\ 1 - M[m], & \text{otherwise,} \end{cases} \tag{4}$$

$$M'[m] = \begin{cases} M[m], & \text{where,} \quad b[m] \leq 0 \\ 1 - C[m], & \text{otherwise,} \end{cases} \tag{5}$$

Then C' and M' are halftoned to C'_h and M'_h. The final results, C_h and M_h, will then be,

$$C_h[m] = \begin{cases} 1, & \text{where,} \quad b[m] > 0 \;\&\; C'_h[m] = 0 \;\&\; M'_h[m] = 0 \\ C'_h[m], & \text{otherwise} \end{cases} \tag{6}$$



(a) (b)

Figure 4. A color image with 50% coverage in its cyan and magenta channels is halftoned. In (a), the channels are halftoned independently. In (b), the dot-off-dot printing is used.

$$M_h[m] = \begin{cases} 1, & \text{where,} \quad b[m] > 0 \;\&\; C'_h[m] = 0 \;\&\; M'_h[m] = 0 \\ M'_h[m] & \text{otherwise} \end{cases} \tag{7}$$

Using the dot-off-dot printing strategy will however cause a color shift compared to the case when the C and M separations are halftoned independently, see Fig. 4. An image with 50% cyan and magenta has been halftoned. In Fig. 4(a) the color channels are halftoned independently with the monochromatic algorithm presented above. In Fig. 4(b) the color channels of the original image are halftoned dependently as described in this section. As can be seen two different colors are perceived from these two images. In the next section we will show that by using the dot-off-dot strategy we can achieve the same color as the independent case with less ink.

Dependent Color Halftoning Needs Less Ink

As mentioned in the previous section in the proposed dependent color halftoning method dot-on-dot printing is avoided as much as possible. We also mentioned that we only halftone the C and M channels dependently. The Y channel is halftoned independent of the other two channels. When the C and M channels of a color image with a coverage of c and m are halftoned independently the resulting tristimulus value X can be approximated by using Demichel's[10] and Neugebauer's[11] equations:

$$c(1-m)X_c + m(1-c)X_m + cmX_{cm} + (1-c)(1-m)X_p \tag{8}$$

where X_c, X_m, X_{cm} and X_p denote the X value for Cyan, Magenta, Blue (Cyan on Magenta) and the paper or substrate. Assume that c_d and m_d denote the coverage of the C and M channels if we want to achieve the same color as in Eq. (8) with the dependent method. We have now two cases, $c_d + m_d \le 1$ and $c_d + m_d > 1$, which will be discussed in the following two subsections.

Where $c_d + m_d \le 1$

If the C and M separations with a coverage of c_d and m_d respectively are halftoned dependently as described above the resulting tristimulus value X is approximated by:

$$c_d X_c + m_d X_m + (1 - c_d - m_d) X_p = X \tag{9}$$

Note that Eq. (9) is valid if and only if $c_d + m_d \le 1$ (or 100%). If this sum is greater than unity then dot-on-dot cannot completely be avoided and Eq. (9) is not valid anymore; see below. Similar equations can be written for the Y and Z values. If we want the dependent halftoned image to have the same color as the independent one, then we have to find c_d and m_d by solving the following equation system. This equation system is achieved by letting Eqs. (8) and (9) and their corresponding equations for Y and Z values be equal.

$$\begin{cases} (c_d - c)a_x + (m_d - m)b_x = cmk_x \\ (c_d - c)a_y + (m_d - m)b_y = cmk_y \\ (c_d - c)a_z + (m_d - m)b_z = cmk_z \end{cases} \tag{10}$$

where,

$$a_x = X_p - X_c,\ \ b_x = X_p - X_m,\ \ k_x = X_c - X_m - X_{cm} - X_p$$

$$a_x = Y_p - Y_c,\ \ b_y = Y_p - Y_m,\ \ k_y = Y_c - Y_m - Y_{cm} - Y_p$$

$$a_z = Z_p - Z_c,\ \ b_z = Z_p - Z_m,\ \ k_z = Z_c - Z_m - Z_{cm} - Z_p$$

As mentioned above only the solutions whose sum is less than or equal to unity are accepted. Note that Eq. (10) actually consists of three equations and only two unknowns, i.e., c_d and m_d. These kinds of equation systems are not guaranteed to have solutions. To see whether or not this equation system has a set of solutions we solve its two first equations and we get

$$c_d - c = \frac{b_y k_x - b_x k_y}{a_x b_y - a_y b_x} cm \tag{11}$$

$$m_d - m = \frac{-a_y k_x + a_x k_y}{a_x b_y - a_y b_x} cm \tag{12}$$

These results are accepted if and only if they satisfy the third equation. We put the solutions shown in Eqs. (11) and (12) into the third equation of Eq. (10). These solutions are accepted if:

$$cm\left(\frac{(b_y k_x - b_x k_y)a_z + (-a_y k_x + a_x k_y)b_z}{a_x b_y - a_y b_x} - k_z\right) = 0 \tag{13}$$

As can be seen Eq. (13) is dependent on the tristimulus values for cyan, magenta, blue and the white paper. These data depend on the colors that are used. In Table I we show these data for the ink jet printer at our department (Deskjet 970Cxi).

Since we solved Eq. system (10) for X and Y values then we know that we are going to have exactly the same X and Y values in both independent and dependent cases. By using the data shown in Table I in the left hand side of Eq. (13) we can find the difference between the Z values for these two cases. The absolute value of the difference is 1.9677 cm for the data shown in Table I. Note that in Eq. system (10) we only accept the results that satisfy $c_d + m_d \le 1$. To have the results satisfying this condition the sum of c and m should be less than almost 1.2 and therefore cm is at most around 0.36. Thus, the difference in the Z-value is not more than $1.9677 \times 0.36 \approx 0.7$, which is negligible. The reason why this result is not

TABLE I. The X, Y and Z Values of Cyan, Magenta, Blue (Cyan on Magenta) and the White Paper for Inkjet Printer (Deskjet 970Cxi).

	X	Y	Z
Cyan	52.36	76.30	105.227
Magenta	64.83	34.04	98.85
Blue	36.56	45.16	98.53
Paper	95.05	100	108.89

exactly equal to zero is that Neugebauer's equations are approximations, which actually work quite well in this case. However, if in some cases this difference is big and consequently not negligible then we add yellow into the left hand side of Eq. system (10). By doing so, we get an equation system consisting of three equations and three unknowns. This new equation system has one set of solutions, unless the determinant of the coefficients is zero, which is highly improbable.

Now we combine Eqs. (11) and (12) together to see how much ink is needed.

$$(c_d + m_d) - (c + m) = \frac{b_y k_x - b_x k_y - a_y k_x + a_x k_y}{a_x b_y - a_y b_x} cm \approx -0.5cm \tag{14}$$

Since both c and m are non-negative numbers the right hand side of Eq. (14) is always zero or negative. That means $c_d + m_d$ is always less than or equal to $c + m$. Therefore the amount of ink needed to reproduce the same color is less for the dot-off-dot printing, which will be shown below by simulation as well.

Where $c_d + m_d > 1$

In this case we have cyan, magenta and blue. The X-value is therefore approximated by Eq. (15).

$$(1 - m_d)X_c + (1 - c_d)X_m + (c_d + m_d - 1)X_{cm} = X \tag{15}$$

The Y and Z values are calculated correspondingly.

If we want the dependent halftoned image to have the same color as the independent one, then we have to find c_d and m_d by solving the following equation system. This equation system is achieved by letting Eqs. (8) and (15) and their corresponding equations for Y and Z values be equal.

$$\begin{cases} c_d a_x + m_d b_x = ck_x + ml_x + (cm+1)n_x \\ c_d a_y + m_d b_y = ck_y + ml_y + (cm+1)n_y \\ c_d a_z + m_d b_z = ck_z + ml_z + (cm+1)n_z \end{cases} \tag{16}$$

where,

$$a_x = X_{cm} - X_m,\ b_x = X_{cm} - X_c,\ k_x = X_c - X_p,$$
$$l_x = X_m - X_p \text{ and } n_x = X_{cm} - X_p - X_c - X_m \tag{17}$$

The other coefficients in Eq. system (16) can be found by just replacing x by y or z in Eq. (17). Solving this equation system is unfortunately not as straightforward as Eq. system (10).

Simulations

Since Eq. system (16) was difficult to solve analytically we chose to simulate all possible cases in order to show that dependent halftoning needs less ink. We do actually the simulations for both cases discussed above.

We vary therefore c and m from 0 to 1 with a step of 0.01, then we solve the two first equations in Eq. system (10) and find c_d and m_d. If $c_d + m_d \leq 1$ the results are accepted and we put these results into the third equation in Eq. system (10) and calculate the absolute value of the difference between the Z-values for the dependent and independent case. But if $c_d + m_d > 1$, we solve Eq. system (16) and find c_d and m_d. We put these solutions in the third equation in Eq. system (16) and find the absolute value of the difference between the Z values again. In Fig. 5(a) these differences are plotted for all possible cases. As can bee seen the biggest difference is 0.672. To show that this difference is actually negligible it is more accurate to calculate

$$\Delta E_{Lab} = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2},$$

(the color difference in the uniform color system CIELab) for all cases. Since the X and Y values are equal in all cases then the L and a values are also equal. However, we have calculated ΔE_{Lab} for all cases and plotted the result in Fig. 5(b). As can bee seen the largest ΔE_{lab} is 0.43, which is actually much less than the JND (Just Noticeable Difference) according to Ref. [12] which defines a ΔE_{lab} of around 2.3 as correspondsing to a JND.

To show that dependent halftoning needs less ink to produce the same color as the independent case we run the same simulation as before and plot $(c + m) - (c_d + m_d)$, see Fig. 5(c). From the plot we can see that this difference is non-negative in all cases, which means that to produce the same color we always need more ink in the independent case.

These three plots are actually quite similar. They are equal to zero where either c or m is equal to 0% or 100%, which means that the two cases are exactly the same. All the local maximums are for those c and d's that result in $c_d + m_d = 1$. This is the case when in the dependent case, we have only cyan and magenta (no paper and no blue). This can also be proven mathematically (geometrically) by studying the color gamut of the utilized printer in the XY-plane. However, the absolute (global) maximum occurs when $c = 58\%$ and $m = 59\%$, which results in $c_d = 60\%$ and $m_d = 40\%$.

Results

In Fig. 4 we showed two images with 50% coverage in their cyan and magenta channels. In Figs. 4(a) and 4(b) the channels were halftoned independently and dependently, respectively. The images are perceived to have two different colors. In order to make them have the same color we solved Eq. system (10) for $c = 0.5$ and $m = 0.5$. The solution is $c_d = 0.5113$ and $m_d = 0.3648$ (Note that $c_d + m_d < 1$). Then an image with 51.13% and 36.48% coverage in its C and M separations was halftoned by the proposed dependent method, see Fig. 6(b). First of all we can see that the dots are more homogeneously placed in the image shown in Fig. 6(b). Second, we can see that they both have the same color. Note that for a correct color reproduction these images should be printed using the printer for which these calculations were done, i.e., Deskjet 970Cxi. Third we can observe that the amount of ink used to reproduce the image in Fig. 6(b) is less. It is actually

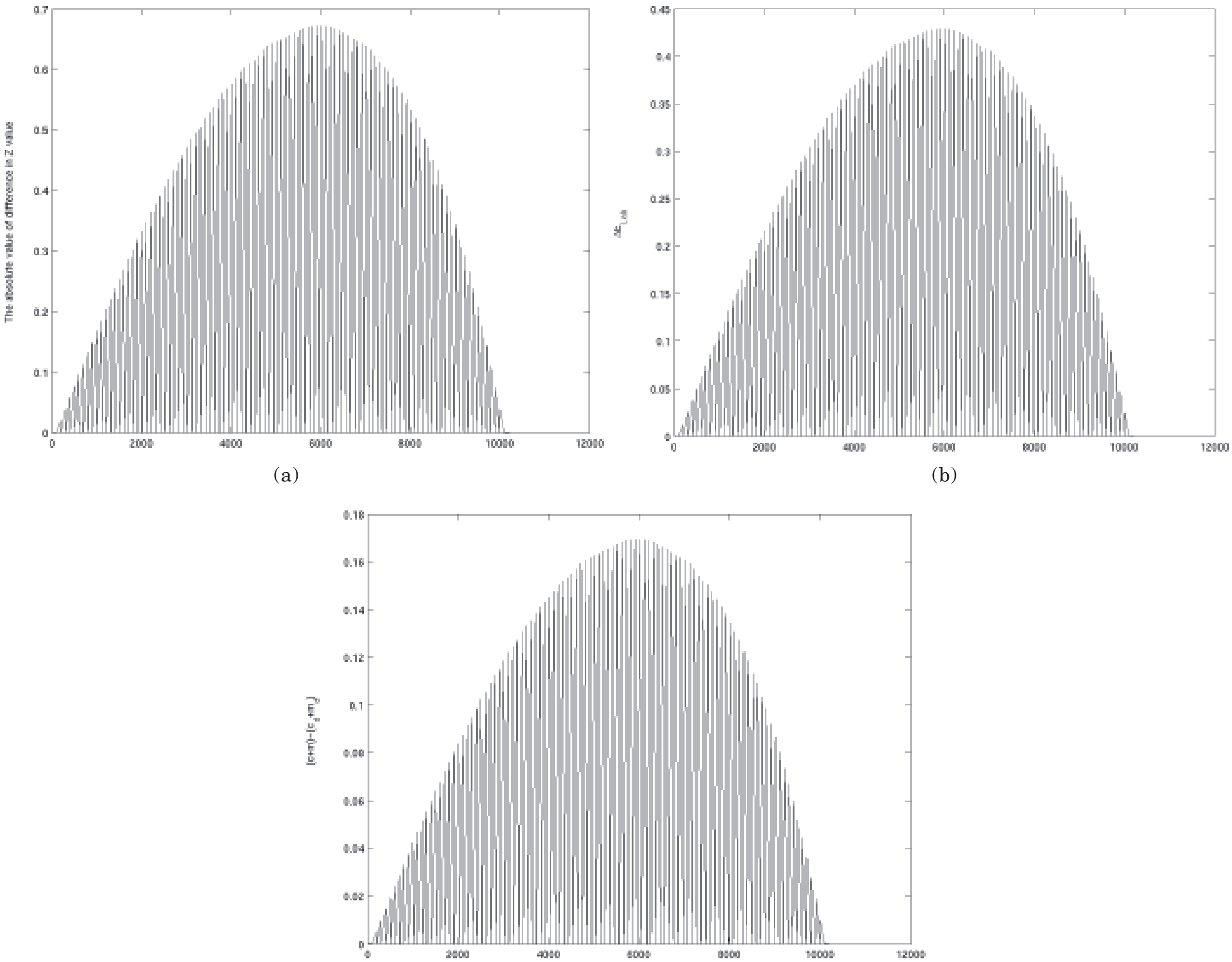


Figure 5. Comparison between the strategy where the color channels are halftoned independently and the strategy when dot-on-dot printing is avoided as much as possible, as discussed in the text. The simulation was run for all c and m varying from 0 to 1 with a step of 0.01. In (a), the absolute values of the difference in Z-values are shown. In (b), ΔE_{Lab} is shown for all cases. In (c), the difference between the amount of ink that is needed for these two different strategies is shown.

$$\frac{(0.5+0.5)-(0.5113+0.3648)}{0.5+0.5}\times 100 \approx 12\%$$

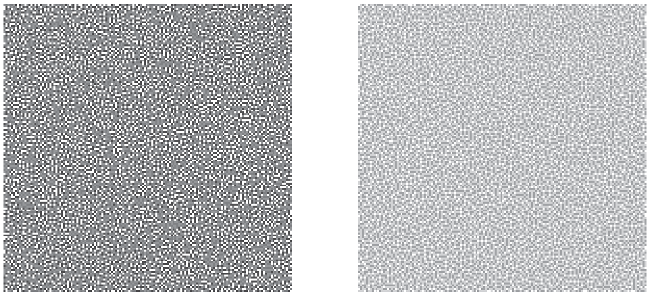
less for dependent halftoning.

Another example is shown in Fig. 7. A color image with 5% coverage in its both cyan and magenta channel has been halftoned. In Fig. 7(a) the channels are halftoned independently with the proposed monochromatic halftoning method. In Fig. 7(b) the image is halftoned with Vector Error Diffusion method, where the colors are treated by vectors in a XYZ color space. The filter is the Floyd and Steinberg filter with four weightings.[6] In Fig. 7(c) the image is halftoned by the proposed dependent color halftoning method. As can be seen the dots in the image shown in Fig. 7(c) are more homogeneously placed.

In Fig. 8 we show a real image halftoned by the three methods mentioned above. We can see that the image shown in Fig. 8(b) (halftoned with Vector Error Diffusion) is more homogeneous than the one in Fig. 8(a) (where the channels are halftoned independently). The image in Fig. 8(c) that has been halftoned by the proposed dependent method is in turn more homogeneous than the one in Fig. 8(b), see especially the banana and the apple. Error diffusion methods generally have a tendency towards high-pass filtering the original.[13] To be able to compare this method with ours we chose to use a 7×7 filter f in order to sharpen our images. Using a 7×7 filter actually results in better images in print, although the halftoned digital image is less similar to the original (it is a bit sharper). Something worth mentioning here is that for the images in Fig. 8(b) and 8(c) the original image has been transformed before being halftoned in order to match the colors with the colors of the image shown in Fig. 8(a). This transformation has been performed for the data shown in Table I. The images shown in Fig. 8(b) and 8(c) actually consume 8% less ink than the one in Fig. 8(a).

CMYK

In this article we have so far assumed that the color images are represented by their cyan, magenta and yel-



(a) (b)

Figure 6. The original image is a color image with 50% coverage in its Cyan and Magenta channels. In (a), the channels are halftoned independently. In (b), the original image is first transformed to an image with 51.13% and 36.48% coverage in its C and M channels and the result is then halftoned with the proposed dependent color halftoning method. The transformation is performed in order to achieve the same color as that in Fig. 6(a). *Figure 6 is available in color as Supplemental Material on the IS&T website, www.imaging.org, for a period of no less than 2 years from the date of publication.*

low channels. Since we normally have our images in RGB format they are easily converted to CMY by

$$C = 1 - R$$

$$M = 1 - G$$

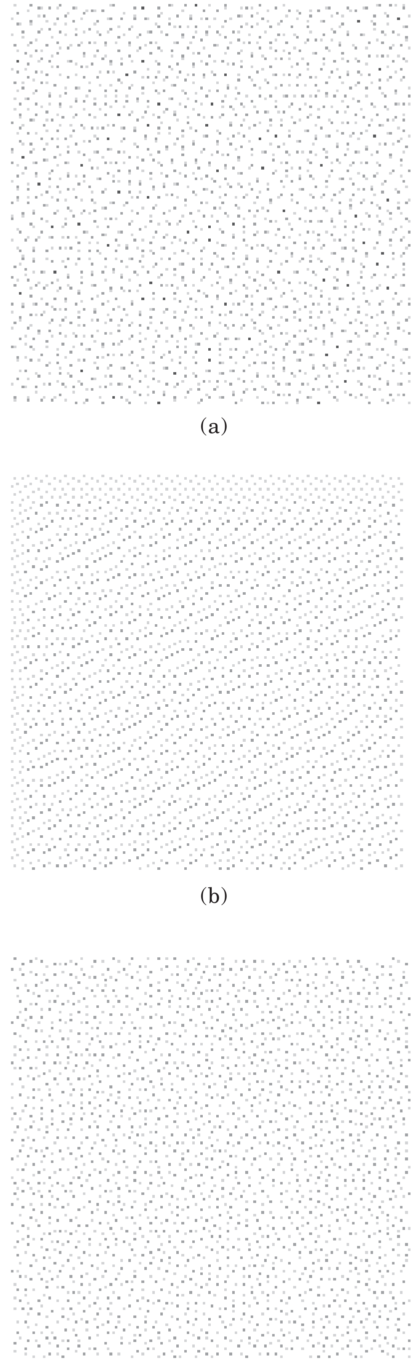
$$Y = 1 - B$$

where R, G and B are assumed to be scaled between 0 and 1. When the C, M and Y channels have been halftoned to C_h, M_h and Y_h it is possible to get the K_h channel by just putting a “1” in the positions of K_h where C_h, M_h and Y_h all three are equal to “1”.

However, in the case where we only have access to the CMYK representation of the original image, it is still possible to use the proposed method. In this case, it is probably better to halftone C, M and K dependently, and Y independently. The algorithm could be described as before. Look for the maximum in these three channels; the strategy for choosing the filter is then as before for the channel where the maximum is found. For the other two channels we should first find the distance between the dots for this particular pixel value, a, or the filter size $(2a + 1) \times (2a + 1)$ by Eq. (2). When the filter size is established we choose a filter of one-third this size for the other two channels, it was half-size when only C and M were halftoned dependently.

Future Work

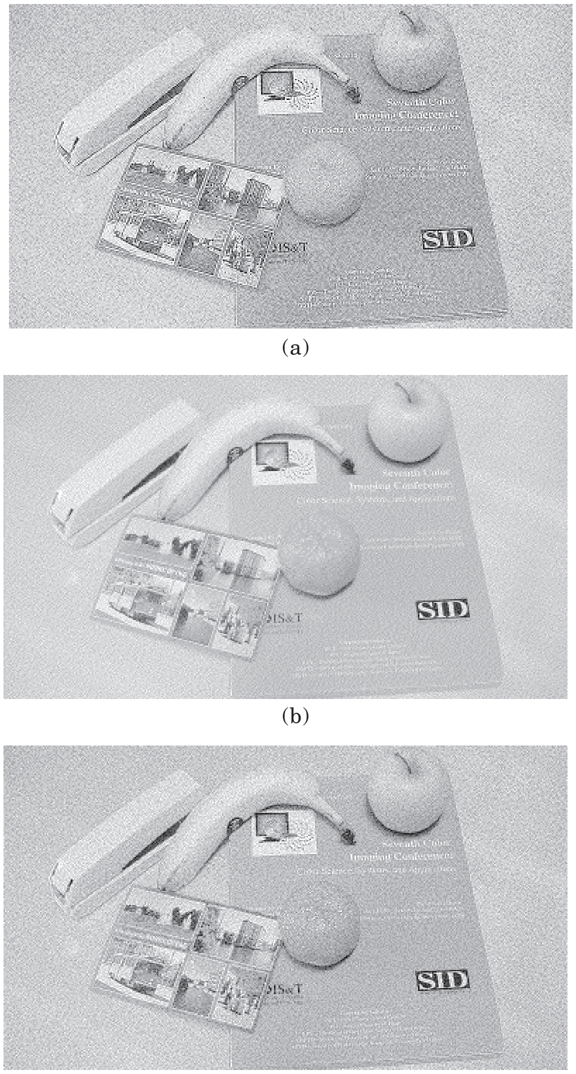
In this article we have halftoned the yellow channel independently. The reason was that the yellow color is much less visible than the other two when printed on a white paper. Therefore it is better to try to place the C and M dots as homogeneously as possible. That is also what other authors have also proposed in their algorithms.[9] This is however not always appropriate according to our experiments. We noticed that when a yellow dot is printed on a blue one, which results in a black dot, it is no longer as invisible as a simple yellow dot on white paper. Therefore we believe that if the yellow dots are prevented from being placed on blue dots as much as possible the result will be more homogeneous in the darker parts of color images, see Fig. 9. The image in



(a) (b) (c)

Figure 7. A color image with 5% coverage in its both C and M channels has been halftoned. In (a), the channels are halftoned independently by the proposed monochromatic halftoning method. In (b), the image is halftoned with Vector Error Diffusion method. In (c), the image is halftoned with the proposed dependent color halftoning method. The images are printed at 100 dpi. *Figure 7 is available in color as Supplemental Material on the IS&T website, www.imaging.org, for a period of no less than 2 years from the date of publication.*

Fig. 9(a) is halftoned with the dependent color halftoning method proposed in this article where the yellow channel is halftoned independently. The image shown in Fig. 9(b) is halftoned by the same method but the yellow dots

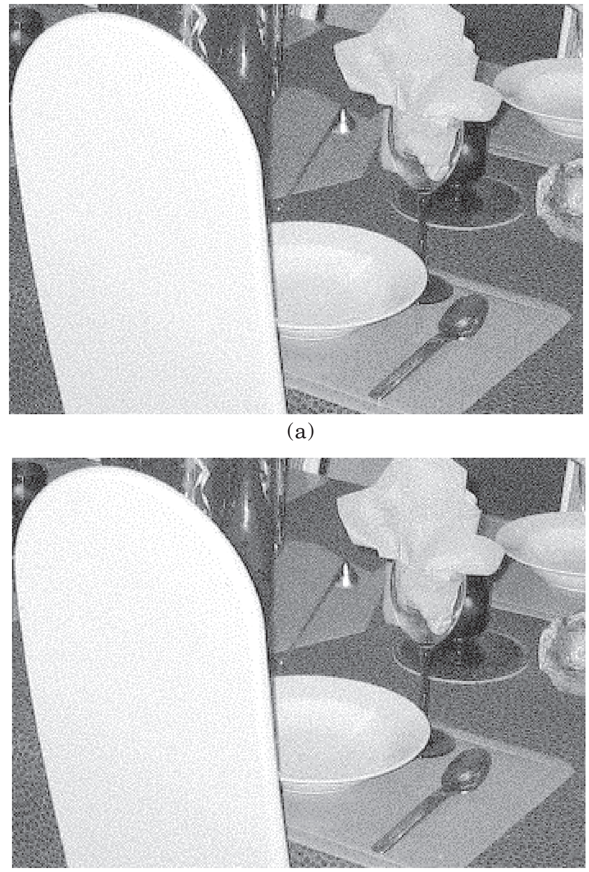


(a)

(b)

(c)

Figure 8. A real color image has been halftoned. In (a), the channels are halftoned independently by the proposed monochromatic halftoning method (the filter f is 7×7). In (b), the image is halftoned with Vector Error Diffusion method. In (c), the image is halftoned with the proposed dependent color halftoning method (the filter f is 7×7). In (b) and (c) the original image has been transformed before being halftoned in order to match the colors with the colors of the image shown in (a). This transformation has been performed for the data shown in Table I. The images are printed at 150 dpi. *Figure 8 is available in color as Supplemental Material on the IS&T website, www.imaging.org, for a period of no less than 2 years from the date of publication.*



(a)

(b)

Figure 9. A real color image has been halftoned with the proposed dependent color halftoning method. In (a), the yellow channel is halftoned independently. In (b), the yellow dots are prevented from being placed on top of the blue dots as much as possible, see the text. The images are printed at 150 dpi. *Figure 9 is available in color as Supplemental Material on the IS&T website, www.imaging.org, for a period of no less than 2 years from the date of publication.*

have been prevented from being placed on top of the blue dots insofar as possible. From these images we can see that the latter case results in a more homogeneous image; see especially the tablemat under the plate, an area with more or less uniform color.

The proposed method is actually very easily extended as described. After the C and M channels are halftoned we exactly know in which positions (pixels) we have a blue dot. We put a negative number at these positions in the yellow channel before halftoning it. The pixels with negative values will not be found as a maximum unless there is no place left, in order to avoid black dots (yellow on blue). In this method, however, it can easily be shown that the black dots can be avoided as long as $c + m + y < 2$. Something that will closely be investigated in near future is that how this modification can affect the color following a similar argument to that presented above.

Conclusions and Discussion

It is always very difficult to judge the quality of a halftoning method. Some methods work very well for certain types of images but not that well for others. The methods presented here have been tested for all possible types of images and always resulted in good and accepted images. One of the advantages of this method is the fact that it is very flexible and can easily be modified to face new challenges. Other iterative halftoning methods, especially methods based on Direct Binary Search,[3,9] probably offer at least the same image quality. But to our knowledge, they are often slower than the proposed method.[14] However, the latest version of the proposed method for grayscale images need 1.7, 8.5, 66 seconds for 512×512, 1024×1024 and 2048×2048 pixel images, respectively. The proposed color halftoning method is thus about four-fold slower. The computer on which the programs were run uses Windows XP with 256 MB RAM and 1.7 GHz CPU. The program is written in C++ and compiled and run in Matlab. We continue to work on finding more efficient ways to make the program run faster.

It has been shown in this article that dot-off-dot printing needs less ink to produce the same color as when the color channels are halftoned independently. This is actually valid for all methods that avoid dot-on-dot printing as much as possible. The idea of halftoning the yellow channel completely independently should probably be considered for the darker parts of images, as discussed in the preceding section.

References

1. K. M. Kang, S. W. Kang and C.W. Kim, A modified error diffusion scheme based on the human visual model, in *Recent Progress in Digital Halftoning II*, IS&T, Springfield, VA, 1999, p. 30.
2. R. Eschbach, Error diffusion with homogeneous highlight and shadow response, in *Recent Progress in Digital Halftoning II*, IS&T, Springfield, VA, 1999, p. 173.
3. M. Analoui and J. P. Allebach, Model-based halftoning using direct binary search, *Proc. SPIE/IS&T*, **1666,** 96 (1992).
4. S. Gooran, *High Quality Frequency Modulated Halftoning*, Thesis No. **668**, Linköping Studies in Science and Technology, Linköping, Sweden. 2001, ISBN 91-7219-913-X.
5. H. R. Kang, *Digital Color Halftoning*, SPIE/IEEE series on Imaging Science & Engineering, SPIE Press, Bellingham, WA, 1999.
6. R. A. Ulichney, *Digital Halftoning*, MIT Press, Cambridge, MA, 1987.
7. U. Meyer-Gruhl and C. Steger, Improvements for color dithering, *Recent Progress in Digital Halftoning II*, IS&T, Springfield, VA, 1999, p. 81.
8. T. Kawaguchi, N. Tsumura, H. Haneishi, M. Kouzaki, and Y. Miyake, Vector error diffusion method for spectral color reproduction, *Proc. PICS 1999,* IS&T, Springfield, VA, 1999, p. 394.
9. J. H. Lee and J. P. Allebach, CMYK halftoning algorithm based on direct binary search, *Proc. IS&T/SID* 9*th* *Color Imaging Conference*, IS&T, Springfield, VA, 2001 p. 199.
10. J. A. C. Yule, *Principles of color reproduction*, John Wiley and Sons, New York, 1967.
11. H. Neugebauer, Die theoretischen Grundlagen des mehrfarbenbuchdrucks, *Z. Tech. Phys.* **36,** 75 (1931).
12. G. Sharma, Ed., *Digital Color Imaging (Handbook)*, Xerox Corporation, Webster, New York, 2003.
13. K. T. Knox, Error image in error diffusion, *Proc. SPIE* **1657,** 268 (1992).
14. http://sipi.usc.edu/%7Eortega/icip2001/