Context Dependent Color Halftoning with Color Matching

Sasan Gooran and Björn Kruse*

Keywords: Halftoning, Color, Color Noise, Color Matching

Abstract: An iterative context dependent color halftoning algorithm is presented in this paper. Unlike the normal approach of halftoning a color image, in which the color separations of the original image are halftoned independently, the original color image is halftoned in a context dependent manner. The method is based on a monochromatic halftoning method that was presented earlier. The number of dots to be placed in each color separation of the final halftone image or in different parts of it is decided in advance. There is no dot in the initial halftoned image. In each iteration a new dot is placed at some particular position. The position of this dot depends on where the previous dots were placed. The final halftoned image is obtained after the predetermined numbers of dots have been placed in the color separations of the halftoned image. The strategy to reduce color noise and gain control over color gamut is to prevent dot-on-dot printing as much as possible. The color shifts that might occur because of this dot-off-dot printing strategy have to be compensated before halftoning. This transformation is dependent on some data for the printer with which the halftoned color image is supposed to be printed. The experiments verify that the color noise is notably smaller in the images that are halftoned by the proposed method compared to the images halftoned using the normal approach of halftoning color images.

Introduction

Many methods for halftoning gray-scale images have been proposed in literature (Ulichney, 1987), (Floyd, 1976), (Analoui, 1992), (Nilsson, 1996), (Gooran, 1996), (Gooran, 1998). These methods can directly be used for halftoning color images by independently applying them to the color separations of the original image. Since the color perception is very much dependent on how the separations

*Department of Science and Technology, Linköping University, Sweden

behave in relation to each other, having a good structure in each separation by itself doesn't guarantee a good perception of the final color image. In this paper we will show that halftoning color images in a context dependent manner can notably reduce the color noise of the final halftoned color image. The strategy to reduce color noise is to prevent dot-on-dot printing as much as possible. The color shifts that might occur are handled by transforming the original color image before halftoning. This transformation is performed by using Neugebauer's and Demichel's Equations.

Neugebauer's equations are used to find the resulting color when several small different colored areas are averaged together by the human eye (Neugebauer, 1931):

$$\begin{bmatrix} X_{tot} \\ Y_{tot} \\ Z_{tot} \end{bmatrix} = \sum_i a_i \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \quad (1)$$

with,

(X_i, Y_i, Z_i):	the tristimulus values for the ink color i,
$(X_{tot}, Y_{tot}, Z_{tot})$:	the tristimulus values for the resulting color,
a_i:	the fractional area covered by the ink color i.

and

$$\sum_i a_i = 1 \quad (2)$$

For example when we print with four inks (CMYK) it is possible to create 16 different colors (of which nine are in effect black), so the sum in Eq. 1 will have 16 terms.

Demichel's equations are used to calculate the approximated fractional coverage for each printing color when a semi-stochastic overlap behavior is assumed (Yule, 1967). Let c, m, y and k be the fractional area covered by cyan, magenta, yellow and black ink, respectively. The fractional area not covered by a particular ink is then $1 - x$, where x represents the fractional area covered by the colored ink x. For example, the fractional area covered by cyan and magenta, but not yellow or black (blue color) is:

$$a_{cm} = cm(1-y)(1-k) \tag{3}$$

and the fractional area covered by only magenta is:

$$a_m = (1-c)m(1-y)(1-k) \tag{4}$$

In the transformation mentioned earlier in this section, the tristimulus values should be measured for primary, secondary and black colors when they are printed with the printer which is supposed to be used for printing the final halftoned image. For more detail about how to perform this transformation see the following sections.

In the next section the halftoning method for monochromatic images is described. This is followed by a presentation of the color halftoning method, which is based on the monochromatic halftoning method, and a description of the color matching transformation that should be performed due to the dot-off-dot printing strategy used in the method. A number of examples are given to show how the proposed methods can increase the quality of the halftoned images. The color illustrations are shown in (Gooran 2001a, 2001b), where more examples are also given.

Halftoning Method, Monochromatic

Suppose that we have a $n \times n$ gray scale image to be halftoned. There are $2^{(n \times n)}$ binary images of this size, and only a limited number of them actually resemble the original one. It is clear that most of these images do not represent the original one and therefore can be skipped. For example the final binary image should have the same mean as the original one. Therefore we can decide how many black dots the final image should have before performing the transformation. This number is decided by integrating the gray values that all pixels in the original image hold. The closest integer to this result, say k, can be a good approximation for the number of black dots the final binary image should include, assuming that 1 and 0 represent black and white respectively. The problem of halftoning a $n \times n$ gray scale image with the average close to $k/(n \times n)$ can now be seen as the problem of finding a way of placing k black dots on a totally empty image of the same size so that the final image "resembles" the original one. There exist $\binom{n \times n}{k}$, $n \times n$ binary images with k black dots. For instance for a 16×16 image with an average of 0.4 there are $\binom{256}{102} \approx 29 \times 10^{72}$ such images. We can see here that even for such a small image it is impossible to compare all the possible halftoned images with the original one. However, in the presented method we try to place a certain number of black dots, decided by the mean value of the original image, on a totally white image so that the final image gives a good perception of the original. The gray scale image is assumed to be scaled between 0 and 1. Suppose that we want to minimize the difference between the original gray scale image, g, and the binary image b. A measure that could be used for the difference is,

$$e = \sum_{i,j} (g(i,j) - b(i,j))^2 \tag{5}$$

with,

$x(i, j)$: the pixel value of image x at position (i, j).

Where should we now place the first dot in order to minimize e? Of course in the position where g holds the largest density value (or the maximum). After placing the first dot there, the second dot should be placed where g has its second largest density value, and so on until the decided number of dots are placed. The final result is of course nothing but the gray scale image thresholded with a fixed threshold, which represents only 2 levels of gray and consequently is not a good representation of the original image. The reason is that the measure in Eq. 5 is not a good measure of the difference of these images. Since the human eye acts as a low pass filter, a much better measure would be the difference of the low pass version of the images,

$$e = \sum_{i,j} (f_g(i,j) - f_b(i,j))^2 \tag{6}$$

with,

f_x: the filtered version of image x filtered with filter f.

Where should we place the first dot? The answer is unfortunately not as simple as before. It has been shown that designing filter f by a fast decreasing function can decrease the error in Eq. 6 when the dot is placed at the position of the largest density value (Gooran 2001a). For having a homogeneous placement of dots the filter should also be circularly symmetric. In our experiment we therefore choose to use a Gaussian filter,

$$f(x, y) = ke^{-(x^2 + y^2)/(2\sigma^2)} \tag{7}$$

with,

k:	normalization factor used to make the weights of the filter have a sum of 1,
σ:	the standard deviation.

For having good structure of the placed dots we sometimes need to change the size of the filter and therefore we are not exactly going to decrease the error in Eq. 6 but the error,

$$e = \sum_{i,j} (f_g(i,j) - h_b(i,j))^2 \tag{8}$$

with,

f:	the filter the original image is filtered with,
h:	the filter used within the algorithm.

The sharpness of the final halftoned image can be changed by changing filter f (Gooran, 2001a).

The algorithm is now summarized in Fig. 1. First the original image, g, is filtered with filter f. Then the position of the largest density value in f_g is found and a dot is placed at the same position in b which is totally white to begin with. Then b is filtered with filter h and the difference f_g - h_b is made. The position of the largest density value in f_g - h_b is found again and the next dot is placed at this position and so on until a certain number of dots are placed in b. The number of dots to be placed is decided by the sum of the density values in g, as discussed before. Something worth mentioning here is that in constant images all pixels hold the same value and the program will return the first (or the last) pixel it meets as the position of the maximum. This can cause the resulting image to be highly

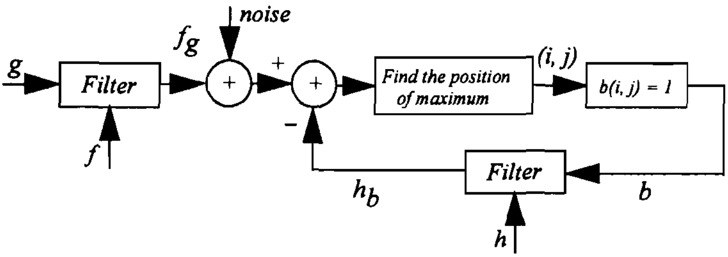
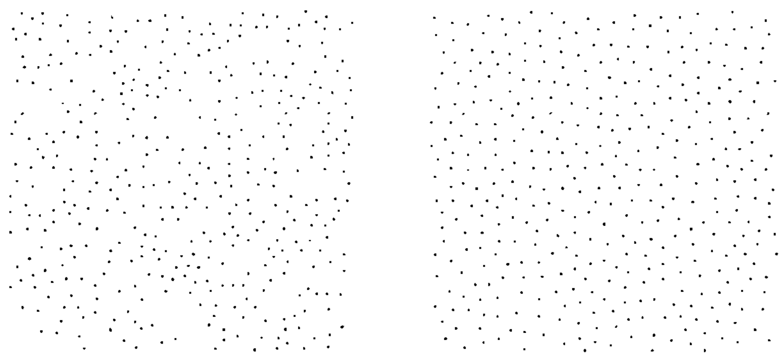


Figure 1: The algorithm.

structured when halftoning a constant image. By adding small quantities of random noise to the gray scale image this effect can be avoided.

According to our experiments an *11 x 11* filter gives rise to homogenous results for almost all constant images ranging from 10% up to 90%. For constant images outside of that range it doesn't result in homogenous halftone patterns. Therefore filters with different size should be used for those images. Let us now give an example. In Fig. 2a we see a constant image with a gray value of 0.01 being halftoned by this method using an *11 x 11* filter. We see that the resulting image is not perceived quite well. What we really would like to have here is that the dots are placed homogeneously and as far apart as possible, which is not the case in Fig. 2a. The reason is that an *11 x 11* filter is not big enough. Fig. 2b shows this constant image halftoned by the proposed algorithm using a *21 x 21* filter. It is obvious that the halftone pattern in Fig. 2b is more homogeneous than that in Fig. 2a.



a b

Figure 2: A constant image with a gray value of 0.01 is halftoned by the presented method using a) an 11 x 11 filter b) a 21 x 21 filter.

Therefore when halftoning a regular image a filter with different size should be used in the light and dark parts of the image. In other parts an *11 x 11* filter is used.

Halftoning Method, Color

All halftoning techniques designed for monochromatic images can directly be used for chromatic images by applying them to the color separations independently. When the separations of a chromatic image are halftoned independently, the dot placement in each separation is controlled only in that separation and doesn't affect the other separations. Since the color perception is very much dependent on how the separations behave in relation to each other, having a good structure of dot placement in each separation by itself doesn't guarantee a very good color perception of the halftoned color image. We are going to show that a detailed control of the dot placement can reduce the color noise of the final halftoned color image. Due to the nature of the method presented in the previous section, it can be extended to a method for chromatic images that halftones the separations dependently. Recall from the previous section that in the monochromatic method the dot placement in the halftoned image was controlled by the filter utilized within the algorithm. In this section we will see how chromatic images can be halftoned in a context dependent manner by using appropriate filters.

First of all we assume that the color image is divided into several separations and each separation is represented by a continuous-tone image with values between 0 and 1. For the sake of simplicity we also assume that the color image is represented by its C, M and Y separations (the image can be represented by C, M, Y and K or other separations as well). The algorithm is almost the same as for the monochromatic case. The separations of the halftoned image are supposed to be totally white to begin with.

The algorithm begins with finding the position of the largest density value, that is the largest density value in all three separations of the original color image. Then a dot is placed at the same position in the corresponding separation of the halftoned image. In the monochromatic case a filter was used to control the dot placement and the filtered version of the binary image was subtracted from the filtered version of the original one. Now besides that we do the same in this particular separation, we use another (or the same) filter to feedback the impact of this dot to the other separations, see Fig. 3. That means, when a dot is placed for example in the M separation, some neighborhood of this position in the M separation is affected, decided by filter h, and another neighborhood of this position in C & Y are also affected using the same or another filter.

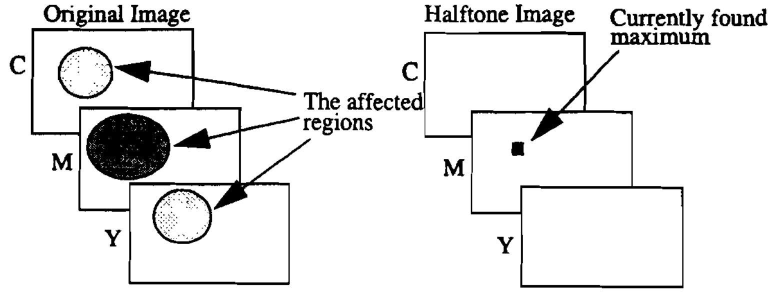


Figure 3: The algorithm for chromatic images. The placed dot impacts all separations of the original image.

Let us now give a simple example. Suppose that we have a color image that has 2%, 3% and 0% coverage in its C, M and Y separations, respectively. This image is first halftoned by applying the monochromatic halftoning method to the separations independently, see Fig. 4a. This image is also halftoned by the proposed color halftoning method, see Fig. 4b. Since in the first case the separations are halftoned independently, the filter used for each separation is designed only for that separation. For having a good structure of dot placement the filter used for the C and M separations should be *15 x 15* and *13 x 13* respectively (Gooran, 2001a). The size of the filter used for each separation is only dependent upon the coverage of that separation. But in the dependent case, we want the dots in both separations to be placed homogeneously over the entire halftoned color image. The filter size should therefore correspond to a coverage of 2 + 3 = 5% (Gooran, 2001a). The filter used for both separations is therefore *9 x 9* in this case.

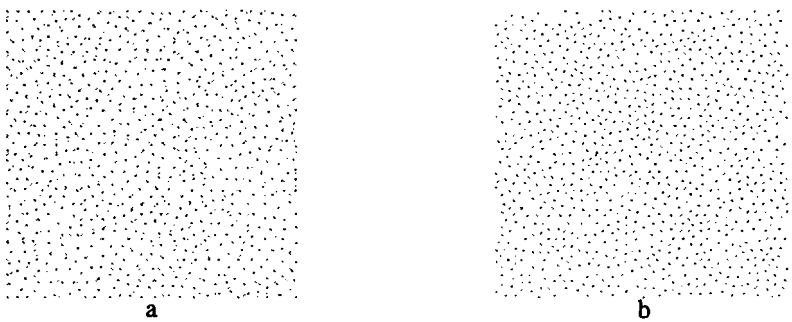
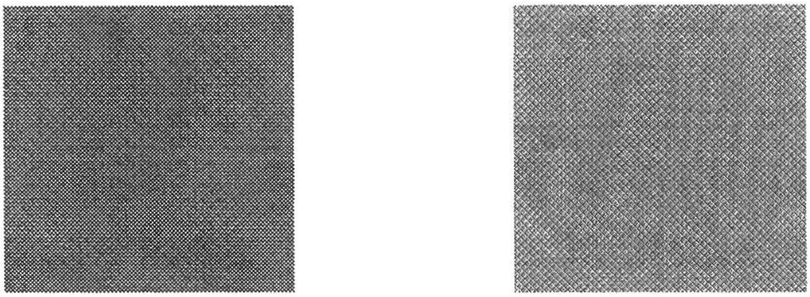


Figure 4: A color image with 2%, 3% and 0% coverage in its C, M and Y separations is halftoned with the proposed methods. In a, the separations are halftoned independently. In b, the separations are halftoned dependently by the proposed color halftoning method.



a b

Figure 5: A color image with 50% coverage in its Cyan and Magenta separations is halftoned. a) dot-on-dot printing strategy is used. b) dot-off-dot printing strategy is used.

Color Matching

In the previous section we have seen how we can halftone the color separations of a color image dependently and prevent the dots in different separations from being placed on top of each other as much as possible. The question we ask now is: If the resulting halftoned image is printed, will all the colors in the image be produced correctly? The answer is no. To illustrate the difference between the colors of printed halftoned images that are built by dot-on-dot and dot-off-dot strategies, we give an example. Suppose that we have an image that has 50% coverage in its Cyan and Magenta separations and no coverage in its Yellow separation. We halftone the image in two ways. In the first one we make every dot in the Cyan separation be placed on top of a dot in the Magenta separation. In the second one we prevent all dots in these two separations from being placed on top of each other, see Fig 5. We see that images in Fig. 5a and b produce two different colors. The reason is that these colors don't have the same tristimulus values, that is the X, Y and Z values for these colors are different. We measured the tristimulus values for black, primary and secondary colors when printed with the printer at our department, i. e. an inkjet printer (Deskjet 970CXi). The tristimulus values for the cases discussed above can be calculated by Neugebauer's equations:

$$\text{Dot on Dot}\begin{cases} X = 0.5X_{CM} + 0.5X_{paper} \\ Y = 0.5Y_{CM} + 0.5Y_{paper} \\ Z = 0.5Z_{CM} + 0.5Z_{paper} \end{cases} \tag{9}$$

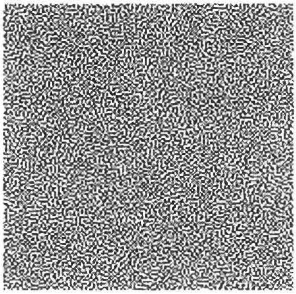
$$\text{Dot off Dot}\begin{cases} X = 0.5X_C + 0.5X_M \\ Y = 0.5Y_C + 0.5Y_M \\ Z = 0.5Z_C + 0.5Z_M \end{cases} \qquad (10)$$

with,

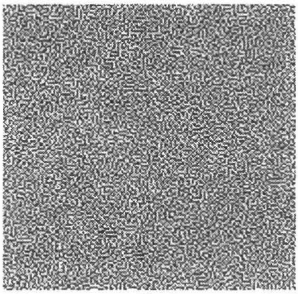
(X_i, Y_i, Z_i):	the tristimulus values for the ink color *i*,
(X_{ij}, Y_{ij}, Z_{ij}):	the tristimulus values for the ink color *i* printed on j,
$(X_{paper}, Y_{paper}, Z_{paper})$:	the tristimulus values for the bare paper (no ink).

In Fig. 6 the results of halftoning this color image with the proposed methods, i. e. independently and dependently, are shown. Obviously they are not the same color. The X, Y and Z values for the image to the right equal to the ones calculated for dot-off-dot case above because we know that in our algorithm no dot will be placed on top of another one before the whole image is filled. Now the question is whether we can make our algorithm produce all colors by avoiding dot-on-dot as much as possible. The answer is yes for some colors.

Suppose that we want our algorithm to produce a certain color (X_t, Y_t, Z_t). We assume that no dots in different separations are placed on top of each other. The proposed method can fulfill this demand if the sum of the newly calculated coverages of the separations does not exceed 100%. Of course a negative coverage is not accepted either (Gooran, 2001a). Therefore we will only have the primary colors and the bare paper. Now in order to find the coverage in the C, M



a



b

Figure 6: A color image with 50% coverage in its Cyan and Magenta separations is halftoned. a) by the monochromatic halftoning method being applied to the separations independently. b) by the proposed color halftoning method.

and Y separations the following equation system should be solved for the target color (X_t, Y_t, Z_t), i.e. the color we want our algorithm to produce.

$$\begin{cases} X_t = cX_c + mX_m + yX_y + (1-c-m-y)X_{paper} \\ Y_t = cY_c + mY_m + yY_y + (1-c-m-y)Y_{paper} \\ Z_t = cZ_c + mZ_m + yZ_y + (1-c-m-y)Z_{paper} \end{cases} \tag{11}$$

with,

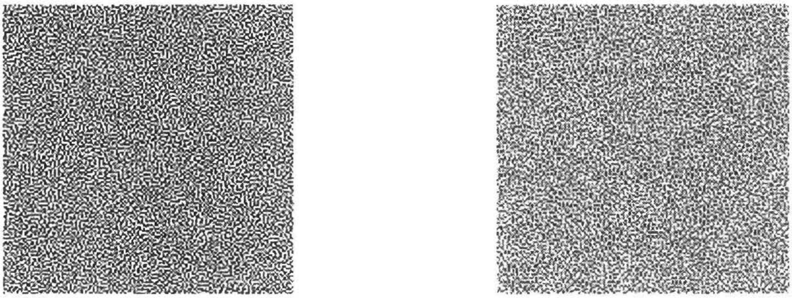
(X_t, Y_t, Z_t):	the tristimulus values for the target color,
c, m, y:	the coverage in the *C*, *M* and *Y* separations.

As long as Eq. 11 has acceptable solutions for any required color's X, Y and Z values the color can be produced by dot-off-dot printing strategy and by using the inkjet printer (Deskjet 970CXi).

Let us now make our algorithm produce an image whose color matches the one of the image in Fig. 6a. For finding the tristimulus values for this color Demichel's equations are used. For this example we will have $a_{cmy} = 0$, $a_c =$ 0.25, $a_m = 0.25$, $a_y = 0$, $a_{cm} = 0.25$, $a_{cy} = 0$, $a_{my} = 0$, and $a_{paper} = 0.25$. a_{xy} denotes the fractional area covered with only x and y color inks. Note that according to Demichel's equation, for instance the fractional area covered with cyan and magenta is calculated by $a_{cm} = cm(1-y)$. Now we calculate the color of the image in Fig. 6a and use this color as the target color. Then Eq. 11 is solved for this color and the new coverages are calculated. The calculated coverages of C, M and Y separations are 50.82%, 36.51% and 0.54%, respectively. That means, the image that should be halftoned by the proposed color halftoning method in order to produce this color has 50.82%, 36.51% and 0.54% in its C, M and Y separations respectively.

Fig. 7a shows again this image being halftoned by applying the monochromatic halftoning method to the color separations independently. Fig. 7b shows the transformed image being halftoned by the proposed color halftoning method. The transformation was performed in order for the algorithm to produce the same color as the one in Fig. 7a.

Now we are going to see how a regular color image should be halftoned with the proposed color halftoning method. As mentioned earlier the original image has to be transformed to a new one before the method is applied.



a b

Figure 7: The original image is a color image with 50% coverage in its Cyan and Magenta separations. a) This image is halftoned by the monochromatic halftoning method being applied to the separations independently. b) The original image is first transformed to an image with 50.82%, 36.51% and 0.54% coverage in its C, M and Y separations and the result is then halftoned with the proposed color halftoning method. The transformation is performed in order to achieve the same color as it would have been obtained if a semi-stochastic overlap behavior was assumed.

First the tristimulus values for each pixel's c, m and y values are calculated by Demichel's and Neugebauer's equations. Then Eq. 11 is solved in order to find the new c, m and y values. If any of these values is negative or their sum exceeds 1, they are not accepted and the c, m and y values remain unchanged. At the same time we solve these equations, we find out in which parts of the image the values are changed. A mask indicating which parts of the image are changed is thus produced. When the entire image is transformed the halftoning method can start processing the new image. The mask that was built is also used in the algorithm. For each processed pixel this mask indicates whether the pixel's c, m and y values are changed or not. If they are changed then the other color separations should be affected precisely as before. But if the c, m and y values are not changed the other separations should remain unaffected for that particular position. Hence the separations are halftoned dependently in those parts where the c, m and y values are changed and independently elsewhere.

More color images halftoned with the proposed color halftoning method are shown in (Gooran 2001a, 2001b).

Conclusions

An iterative halftoning method for gray scale images has been introduced. This method is extended to a color halftoning method that halftones color images in a context dependent manner. In this method the dots in different separations are prevented from being placed on top of each other as much as possible. This can cause some color shifts in the final halftoned image. The color image should therefore be transformed before halftoning. The given examples show that the proposed method results in halftoned images with notably smaller color noise compared to results of using the normal approach of halftoning color images, in which the separations are halftoned independently.

Acknowledgment

The authors gratefully acknowledge the financial support of the research program T2F.

References

Analoui, M., and Allebach, J. P.
1992 "Model-based Halftoning Using Direct Binary Search," Proceedings of the 1992 SPIE/IS&T Symposium on Electronic Imaging Science and Technology, Vol. 1666, San Jose, CA, 1992, pp. 96-108.

Floyd, R. W., Steinberg, L.
1976 "An adaptive algorithm for spatial grey scale," Proc. SID, Vol. 17/2, 1976, pp. 75-77.

Gooran, S., Österberg, M., Kruse, B.
1996 "Hybrid Halftoning - A Novel Algorithm for Using Multiple Halftoning Technologies," Proc. IS&T Int. Conf. on Digital Printing Technologies (NIP12), Oct. 1996, pp. 79-86.

Gooran, S.
1998 "Hybrid and Frequency Modulated Halftoning," Licentiate Thesis LIU-TEK-LIC-1998: 61, Linköping Studies in Science and Technology, 1998, ISBN: 91-7219-307-7.

2000 "Context Dependent Color Halftoning in Digital Printing," proceedings of IS & T's 2000 PICS Conference, March 2000, Portland, USA, pp. 242-246.

2001a "High Quality Frequency Modulated Halftoning," Ph.D. Thesis No. 668, Linköping Studies in Science and Technology, Jan. 2001, ISBN 91-7219-913-X.

2001b "www.itn.liu.se/~sasgo/TAGA2001".

Neugebauer, H.
1931 "Die theoretischen Grundlagen des Mehrfarbenbuchdrucks," Z.

Tech. Phys. 36, 1931, pp. 75-89.

Nilsson, F.
1996 "Precomputed frequency modulated halftoning maps that meet the continuity criterion," Proceedings of the IS&T International Conference on Digital Printing Technologies (NIP12), 1996, pp. 72-77.

Ulichney R. A.
1987 "Digital Halftoning" (MIT Press), 1987.

Yule, J. A. C.
1967 "Principle of color reproduction" (New York: John Wiley & Sons) 1967.