

Hybrid Halftoning—A Novel Algorithm for Using Multiple Halftoning Techniques

Sasan Gooran, Mats Österberg and Björn Kruse

Department of Electrical Engineering, Linköping University, Linköping, Sweden

Abstract

The rendering quality in halftoning is a critical issue. The quality aspects are more important in some images than in others. The quality of skin tone rendering of halftoned images generated by frequency modulated (FM) halftoning techniques differs from those generated by conventional halftoning techniques. Some judge the conventional halftoning techniques as superior in smoothly varying tones whereas frequency modulated halftoning techniques excel in heavily textured images. This paper describes an algorithm that can incorporate both technologies simultaneously. The technique is an iterative optimization of the binary halftone image with respect to the differences between the original and the halftoned images. The performance of the algorithm can be controlled by the nature of the original state of the iteration. The algorithm can in effect accommodate any type of halftone that can be described by a threshold matrix.

Introduction

The fact that most printing and display devices are restricted to black and white pixels, has led to an increased interest in halftoning techniques, which can be used to transform a graytone image into a binary (halftoned) one, which can be printed or displayed using a two-level device. The simplest technique one can think of is to quantize the graytone image with a fixed threshold, which in most cases doesn't lead to a satisfying result. Many different halftoning algorithms have been introduced in literature during the last half a century. All of the first introduced algorithms were operating in a pointwise manner, in which each output dot depends only on the input data value at the corresponding location. The error diffusion algorithm, introduced by Floyd and Steinberg in 1975,¹ on the contrary, requires neighbourhood operations and consequently has more computational complexity. There are also other types of halftoning algorithms which process globally, such as neural nets,² the iterative Fourier transform algorithm (IFTA)³ and the Direct Binary Search.⁴ Halftoning techniques using pre-computed threshold matrices have an advantage in their speed of operation while iterative techniques show more visually pleasing result but have very slow operation. A halftoning technique bridging the gap between these two techniques has recently been introduced.⁵

There are two main screening methods, AM (Amplitude modulated, also called conventional) and FM

(Frequency modulated, sometimes referred to as stochastic), which can be used to build a halftoned image. In the conventional method the size of the dots is variable while their spacing is constant. In the case of FM the size of the dots is constant whereas their spacing is variable.

The algorithm introduced in this paper, which is an iterative algorithm, can be classified as belonging to the last category, i.e. global processing algorithms, mentioned above. This algorithm requires a halftoning technique which can be described by a threshold matrix. The goal of this algorithm is to improve the "quality" of the first halftoned image, that is, to decrease the difference between the first halftoned image and the original one. With the first halftoned image we mean the graytone image thresholded with a threshold matrix, that describes the halftoning technique. Since the FM technology seems to be superior in heavily textured images and the conventional technology in images with smoothly varying tones we will, by choosing a convenient first threshold matrix, make the algorithm incorporate both conventional and FM halftoning methods, simultaneously. Later in this paper we will show that the implicit update rule of our algorithm has similarities to the Hopfield update rule.⁶ This allows us to identify a quality function describing the behaviour of the algorithm.

Previous Work

Since the optimization of the Hopfield net can be applied to halftoning⁷ the consideration of using Hopfield net has reached high attention during the last years. Specially it has been investigated by Tuttaß and Bryngdahl⁸ and recently by Österberg.¹⁰

If the weights t_{ij} in a Hopfield net fulfill the constraints

$$t_{ij} = t_{ji} \text{ and } t_{ii} = 0 \text{ for } i = 1, \dots, N \text{ and } j = 1, \dots, M$$

the Hopfield net minimizes the so-called energy function $E^{(n)}$, which is

$$E^{(n)} = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M y_i^{(n)} t_{ij} y_j^{(n)} - \sum_{i=1}^N y_i^{(n)} c_i \quad (1)$$

where y_i and c_i denote the outputs and the external values of the Hopfield net, respectively. It has been shown that if

$$t_{ij} = -\sum_{n=1}^N w_{i-n} w_{j-n} \text{ and } c_i = w^* x^* w_i \quad (2)$$

the relation $\sigma^2 = 2 \cdot E^{(n)} + const$ is obtained,⁸ where,

$$\sigma^2 = \sum_{k=1}^N |(X_k - Y_k)W_k|,$$

capital letters denote the discrete Fourier transform of functions, x and y are the graytone image and its binary version respectively, W_k is a transfer function of the visual system and finally $w*x$ means the convolution between w and x . Since the Hopfield net minimizes $E^{(n)}$ minimization of σ^2 , which is often used as a quality criterion for binarization, is obtained. According to Equation 2, the diagonal elements of the weights are

$$t_{ii} = -\sum_{n=1}^N w_{i-n}^2$$

and thus the constraint $t_{ii} = 0$ can only be fulfilled for complex valued elements w_{i-n} . Often w is real valued and this constraint is not fulfilled and therefore a change of the transfer function W is unavoidable and consequently the linear relationship between $E^{(n)}$ and σ^2 is destroyed and σ^2 does not longer decrease monotonously. Nevertheless, acceptable results can be achieved by terminating the calculation in a minimum of σ^2 and by varying available parameters (e.g. the width of the filter). A change in the update rule accounting for the effects of nonzero self-coupling connections is proposed in reference 10.

The Algorithm and the Effect of Involved Parameters

The Algorithm

Our algorithm as mentioned before is an iterative algorithm and can be used for any halftoning technique that can be described by a threshold matrix. The aim is to decrease the difference between the original image, i.e. the graytone image, and the first halftoned image in a low pass region by updating the threshold matrix successively. The algorithm is outlined in Figure 1.

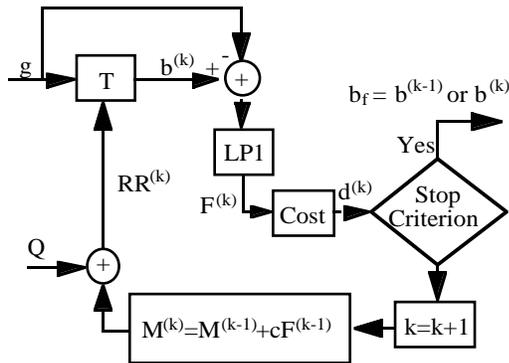


Figure 1. Iterative algorithm used for image binarization.

Before going into details some definitions must be introduced. In the following we will denote the graytone image as g (assume that it is scaled between 0 and 1), the halftoned image in iteration k as $b^{(k)}$, which is the original image thresholded with the threshold matrix $RR^{(k)}$, and the final halftoned image as b_f . Q denotes a

fix threshold matrix by which the halftoning technique can be described. This matrix is the same size as the original image and can be either dependent on or independent of it. At the start k is equal to 1 and $M^{(1)}$ is a matrix of zeros which causes $RR^{(1)}$ to be equal to Q . $F^{(k)}$ denotes the result of low-pass filtering the difference between $b^{(k)}$ and g .

The box, named Cost, calculates a cost function to be decreased. For instance $d^{(k)}$ can be the maximum of the absolute value of $F^{(k)}$, or the sum of the square of $F^{(k)}$'s elements. If the stop criterion is not fulfilled, then the threshold matrix is modified to produce a new halftoned image. Otherwise the algorithm is terminated and the final halftoned image is achieved. Whether the final halftoned image is the halftoned image in the last iteration or in the previous one depends on the stop criterion. An example for stop criterion can be $d^{(k)} > d^{(k-1)}$, that means if the cost in a given iteration k is larger than the cost in the previous iteration the stop criterion is fulfilled. If we use this stop criterion we should make $d^{(0)}$ be larger than $d^{(1)}$ before the stop criterion in the first iteration in order to make the algorithm fulfil its first iteration and the final halftoned image will be the halftoned image in the previous iteration. Finally c denotes a real number which can be constant for all iterations. How this number c for a decided Q should be chosen to give the best result is not clear and can vary from one image to another one. By modifying this algorithm, as we will see in section 3.3, we can decrease the effect of the choice of c . It must be mentioned that there is no guaranty that this algorithm minimizes the cost of the difference with all choices of involved parameters and cost function, but according to our experiments it can decrease the cost with appropriate choices of these parameters.

As mentioned before the implicit update rule of our algorithm has similarities to the Hopfield update rule and consequently we can show that there is a cost function which our algorithm minimizes. But due to the existence of non-zero self-coupling terms, i.e. t_{ii} mentioned in the Previous Work section, we have to modify our algorithm in order to find a quality function describing the behaviour of the modified algorithm. We present this quality function and the modified algorithm in the appendix.

Different Choices of Q

The behaviour of the final halftoned image varies with different choices of Q and thus Q plays an important role in our algorithm. In the following paragraph, we explain how we find a convenient Q describing the FM halftoning technique and later in the Examples section we will show some binary images produced by using the algorithm and this Q . In the section on Hybrid Halftoning, we explain how we can find a convenient Q in order to make the algorithm incorporate AM and FM halftoning techniques simultaneously. Two halftoned images produced by using this Q are also shown in the Examples section.

Our Choice of Q Describing the FM Halftoning Technique

Our aim of choosing a convenient first threshold matrix describing the FM halftoning technique is to

achieve a final halftoned image which has a good contrast by edges, i.e. in the high pass region. Therefore Q has to possess information about high frequency details of the original image. Figure 2 shows a diagram of how we choose such a convenient Q .

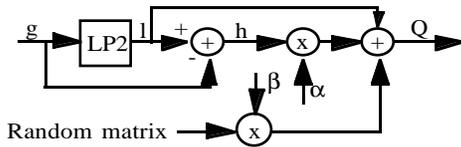


Figure 2. An example for choosing a convenient first threshold matrix describing the FM halftoning technique.

The LP filter, i.e. LP2 in Figure 2, can be the same filter as before, i.e. LP1 in Figure 1. h possesses information of the original image in high frequencies. Except the LP filter there are two other parameters, a and b , which can influence the properties of Q . The larger a is, the more the final image is affected of high frequency informations. The average of the elements of the random matrix, which is the same size as g and consists of random numbers between -0.5 and 0.5 , should be zero. The elements of this random matrix could for example have a blue noise distribution. At the last step l is added to the result in order to make Q has the same dc level as g . In the Examples section some halftoned images are shown in order to illustrate how different choices of α and β affect the behaviour of the final image.

Hybrid Halftoning

The quality of skin tone rendering of halftoned images generated by frequency modulated (FM) halftoning techniques differs from those generated by conventional halftoning techniques. Some judge the conventional halftoning techniques as superior in smoothly varying tones whereas frequency modulated halftoning techniques excel in heavily textured images.

Suppose that the threshold matrices by which we can describe the conventional and FM halftoning techniques are S and R , respectively. In order to make our algorithm incorporate both techniques simultaneously we choose Q to be equal to S in regions where

$$Q_{mn} = \begin{cases} R_{mn}, & \text{For those } m \text{ and } n \text{ for which } H_{mn} = 1 \\ S_{mn}, & \text{For those } m \text{ and } n \text{ for which } H_{mn} = 0 \end{cases} \quad (3)$$

the original image's, i.e. g 's, tones vary smoothly. In other regions of the original image Q is to be equal to R . Figure 3 shows an example to find a mask that is 0 in regions of g where the tones vary smoothly and 1 otherwise. The box named T denotes the threshold operation which thresholds all elements in h with the real constant d . By changing d we can change the properties of H , and consequently Q . An appropriate matrix describing the FM halftoning technique, i.e. the matrix R , can be that presented in the FM Halftoning Technique section. In Equation 3 the subscript mn for all matrices means the location (m, n) in the corresponding matrix.

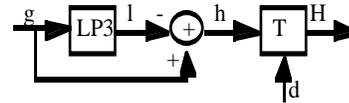


Figure 3. A method that generates a mask, which is 0 in regions of the original image where the tones vary smoothly and 1 otherwise.

Since H consists of only 0 and 1, Q has sharp edges in regions where its pixel values change from the pixel values of R to S and vice versa. To decrease this effect we first let H pass through a LP-filter and then by normalisation we make its value lie between 0 and 1. Suppose that the result after normalisation is denoted by N . Now we calculate Q by the following equation:

$$Q = R \cdot N + S \cdot (O - N) \quad (4)$$

where \cdot denotes pointwise multiplication of matrices and O is a matrix of ones with the same size as the other matrices in Equation 4. A halftoned image produced by using such a first threshold matrix, i.e. Q , is shown in the Examples section.

A Suggestion for Improvement of the Algorithm

Figure 4 shows a modified version of the algorithm shown in Figure 1. Besides the variables described before there are two new variables introduced, i and I . The variable i is an integer that is set to zero at the start. The variable I is also an integer, with the help of which we can stop our algorithm when a satisfying result is reached.

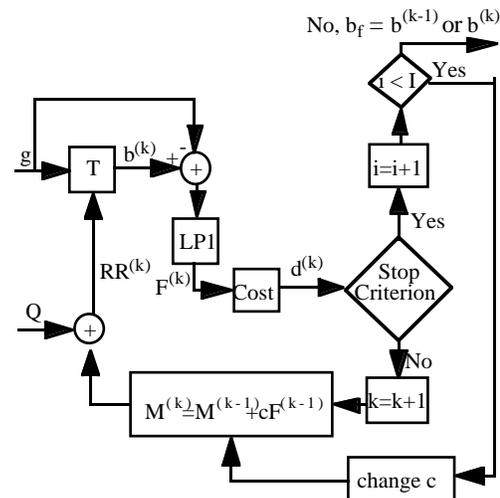
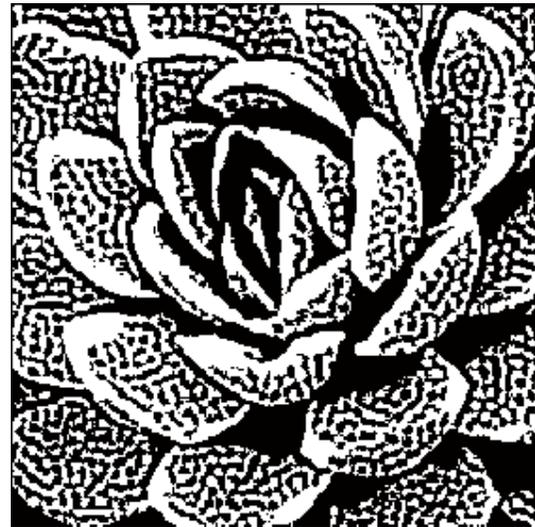


Figure 4. The Modified algorithm.

The advantage of the modified algorithm is that we can now change c after the best result with the previous choice of c is reached. After changing c , for example by multiplying it by a positive real number less than one, the algorithm continues as before until the best result for this c is reached and so on.



(a)

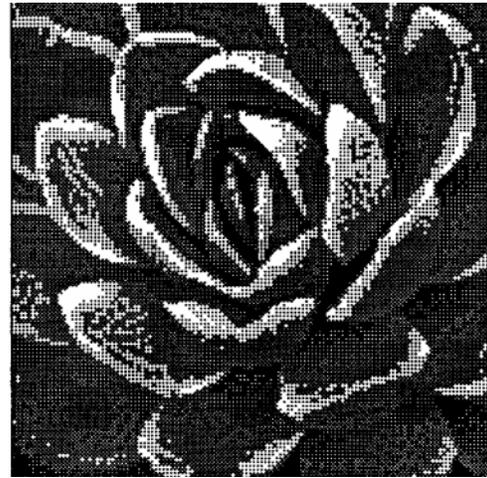


(b)

Figure 5. a) The first halftoned image (The original image thresholded with 0.5). b) The final halftoned image after 19 iterations.

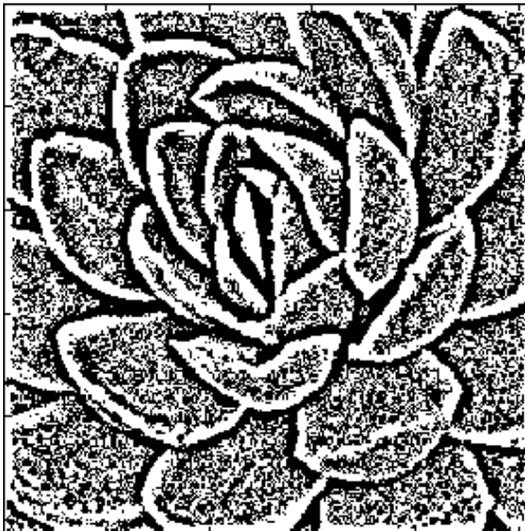


(a)

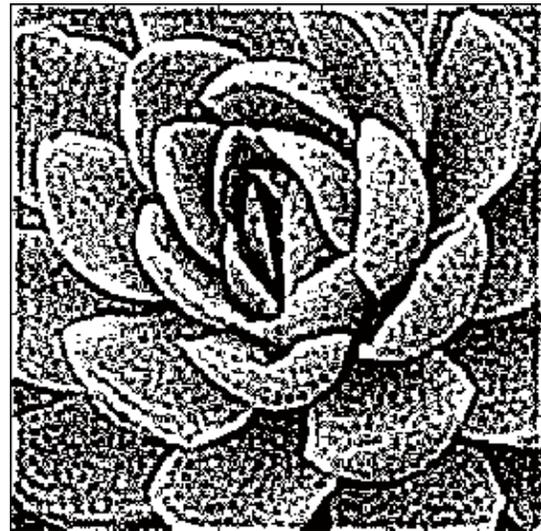


(b)

Figure 6. a) The first halftoned image (The original image thresholded with a threshold matrix generated by Recursive tessellation with 5 levels of gray). b) The final halftoned image after 10 iterations.

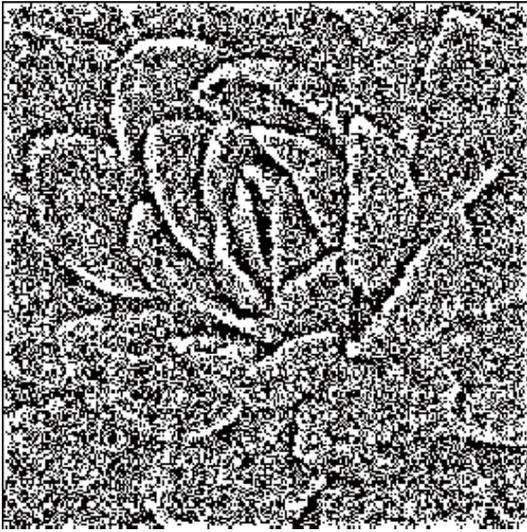


(a)

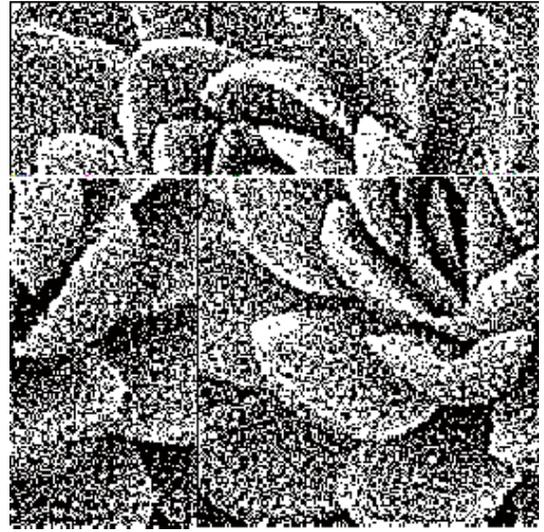


(b)

Figure 7. a) The threshold halftoned image. b) The final halftoned image. Q is computed according to the diagram in figure 2 with $a = 1$ and $b = 0.18$.



(a)



(b)

Figure 8. a) The first halftoned image. b) The final halftoned image. Q is computed according to the diagram in figure 2 with $\alpha = 0.18$ and $\beta = 0.5$.

Examples

In this section we will show some halftoned images produced by using our algorithm introduced in the Algorithm section with different choices of Q and other involved parameters. In all of these examples the original image is a graytone image of 256^2 pixels, the stop criterion is $d^{(k)} > d^{(k-1)}$ and the cost function is the maximum of the absolute value of $F^{(k)}$. The elements of random matrices are supposed to be uniformly distributed.

First, to show the efficiency of this algorithm two examples are given. In the first example we choose Q to be a matrix of 0.5. Figures 5a and 5b show the first and the final halftoned image generated by using this Q , respectively. In the second example we choose Q to be a threshold matrix generated by recursive tessellation with 5 levels of gray.⁹ The first and the final halftoned image produced by using this Q are shown in Figures 6a and 6b, respectively.

Second, to show how our choice of Q , describing the FM halftoning technique, and the different choices of α and β can affect the first and consequently the final halftoned image two examples are introduced as follows. In the first example Q is computed by taking α and β to be 1 and 0.18, respectively. The first and the final halftoned images generated by this choice of parameters are shown in Figures 7a and 7b, respectively. In the second example the value of α is decreased to 0.18 while the value of β is increased to 0.5. The generated halftoned images using these values are shown in Figure 8. By comparing figures 7b and 8b we can see how different values of α and β can change the appearance of the final halftoned image.

Finally to show how our choice of Q presented in subsection 3.2.2 can affect the final halftoned image two figures are shown below. In Figure 9, Q is a matrix only describing the AM halftoning technique and in Figure

10, Q is computed according to the method illustrated in Figure 3 with $d = 0.001$ and Equation 4. The matrix describing the FM halftoning technique, which is a part of Q used in Figure 10, is exactly the same as that used in Figure 8.

Appendix:

According to the algorithm outlined in figure 1 we can write the following statements in iteration $k + 1$:

$$b^{(k+1)} = st(g - RR^{(k+1)}) \quad (5)$$

where st is the threshold function, i.e. $st(x) = 1$ if $x > 0$, and $st(x) = 0$ if $x \leq 0$. Since

$$RR^{(k+1)} = Q + M^{(k+1)}, \quad M^{(k+1)} = c \sum_{i=1}^k F^{(i)}$$

(note that $M^{(1)}$ is a matrix of zeros) and $F^{(i)} = CONV(b^{(i)} - g, filter)$ (that is $F^{(i)}$ is $b^{(i)} - g$ convoluted with LPI's filter kernel, named *filter*, and is the same size as $b^{(i)} - g$) we can rewrite Equation 5 as follows:

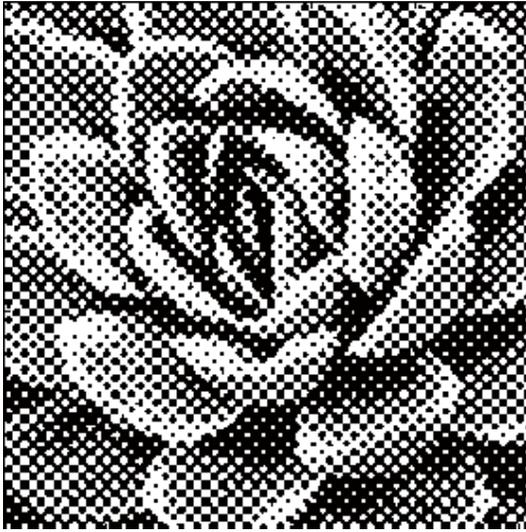
$$b^{(k+1)} = st\left(g - Q - c \sum_{i=1}^k CONV(b^{(i)} - g, filter)\right) \quad (6)$$

By expanding the summation in Equation 6, $b^{(k+1)}$ can be rewritten as:

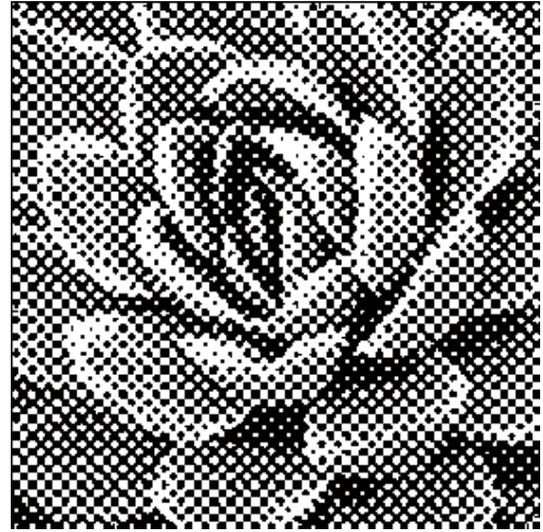
$$b^{(k+1)} = st(CONV(b^{(k)}, -c \cdot filter) + S) \quad (7)$$

where S is independent of $b^{(k)}$ and can be written as:

$$\begin{aligned} S &= g - Q + c \cdot CONV(g, filter) \\ &\quad - c \sum_{i=1}^{k-1} CONV(b^{(i)} - g, filter) \\ &= CONV(CONV(g - Q, invfilter), c \cdot filter) \end{aligned}$$



(a)



(b)

Figure 9. a) The first halftoned image. b) The final halftoned image. Q is a matrix describing the AM halftoning technique.

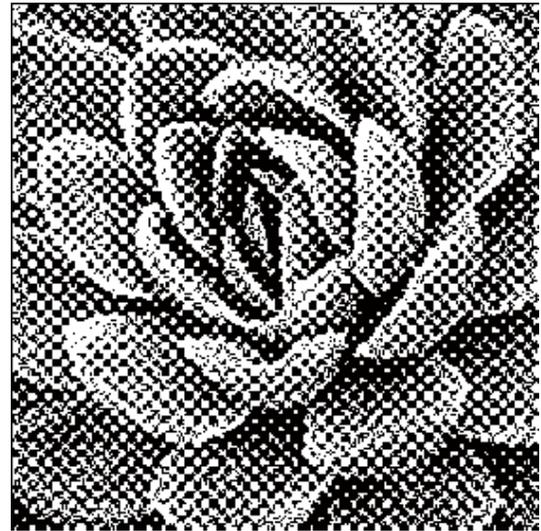
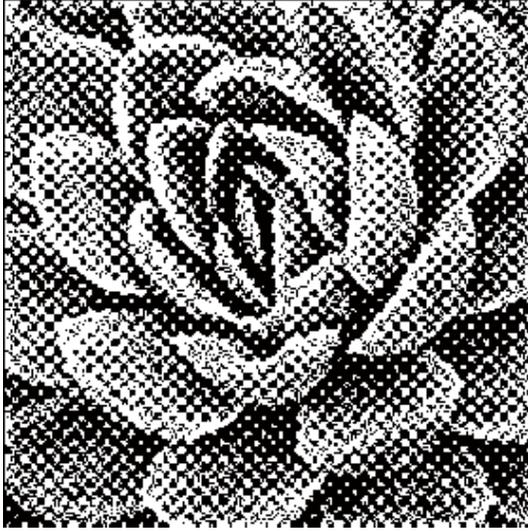


Figure 10. a) The first halftoned image. b) The final halftoned image. Q is computed according to the method illustrated in figure 3 with $d = 0.001$ and Equation 4.

$$+ \text{CONV}(g, c \cdot \text{filter}) - \sum_{i=1}^{k-1} \text{CONV}(b^{(i)} - g, c \cdot \text{filter})$$

$$= \text{CONV} \left(\text{CONV}(g - Q, \text{invfilter}) + g - \sum_{i=1}^{k-1} (b^{(i)} - g, c \cdot \text{filter}) \right) \quad (8)$$

$$= \text{CONV}([\dots], c \cdot \text{filter}) \quad (9)$$

where we have suppressed the expression inside the square brackets, used the distributive property of the convolution and expressed $g - Q$ as:

$g - Q = \text{CONV}(\text{CONV}(g - Q, \text{invfilter}), c \cdot \text{filter})$, with circular convolution and the filter kernel invfilter defined by:

$$\text{invfilter} = F^{-1} \left(\frac{1}{F(c \cdot \text{filter})} \right) \cdot F \text{ and } F^{-1}$$

and denote the Fourier transform and the inverse Fourier transform, respectively. It will be shown that for pixels in the interior of the image and with appropriate connection weights and constant term, the implicit update rule (Equation 7) has indeed the same form as the standard Hopfield rule.¹⁰ With the necessary changes due to the existence of nonzero self-coupling terms, we will below identify the following quality function describing the behaviour of the modified algorithm (see Figure 11) in the k^{th} iteration:

$$\begin{aligned} \text{Qual}(b^{(k)}) = & \sum_{x=f_x}^{N_x-f_x} \sum_{y=f_y}^{N_y-f_y} \left(\text{CONV}(b^{(k)}, H)_{(x,y)} \right. \\ & - \text{CONV} \left(\text{CONV}(g - Q, \text{invfilter}) \right. \\ & \left. \left. + g - \sum_{i=0}^{k-1} (b^{(i)} - g), H \right)_{(x,y)} \right)^2 \quad (10) \end{aligned}$$

For $Qual(b^{(k)})$ to be a valid quality function the following connection between the filter kernel $filter$ and the filter kernel H in Equation 10 must be established:

$$c \cdot filter = CONV(H, \tilde{H}, "full") \quad (11)$$

where \tilde{H} is the 180° clockwise rotation of H , or equivalent: $\tilde{H}(a,b) = H(-a, -b)$. Equation 11 implies: $filter(a,b) = filter(-a, -b)$. Furthermore the sizes must conform, i.e. if the size of H is $2 \cdot f_x + 1 \times 2 \cdot f_y + 1$, where we denote the upper-left most, the central and the lower-right most entry by $H(-f_x, -f_y)$, $H(0, 0)$ and (f_x, f_y) respectively, then the size of $filter$ must be $4 \cdot f_x + 1 \times 4 \cdot f_y + 1$, with the upper-left most, the central and the lower-right most entry denoted by $filter(-2f_x, -2f_y)$, $filter(0, 0)$ and $filter(2f_x, 2f_y)$ respectively. This is exactly what "full" in Equation 11 indicates. It is also assumed that the size of g and consequently $b^{(k)}$ is $N_x \times N_y$.

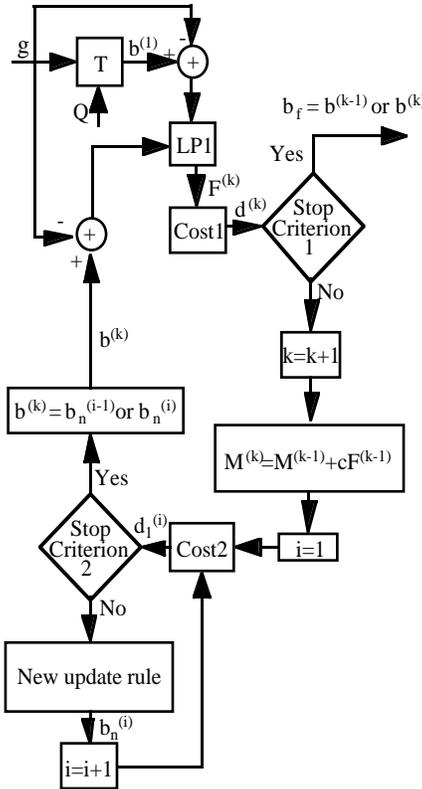


Figure 11. The modified algorithm.

Now we rewrite Equation 10 as:

$$Qual(b^{(k)}) = \sum_{x=f_x}^{N_x-f_x} \sum_{y=f_y}^{N_y-f_y} (CONV(b^{(k)}, H)_{(x,y)})^2 - 2 \sum_{x=f_x}^{N_x-f_x} \sum_{y=f_y}^{N_y-f_y} CONV(b^{(k)}, H)_{(x,y)} \cdot CONV([\dots], H)_{(x,y)} + \sum_{(x-f_x)=(y-f_y)}^{(N_x-f_x, N_y-f_y)} (CONV([\dots], H)_{(x,y)})$$

where we again have suppressed the expression in the square brackets. The suppressed expression is independent of $b^{(k)}$ and thus the third term is constant. The second term can be rewritten as:

$$-2 \sum_{m=f_x}^{N_x-f_x} \sum_{n=f_y}^{N_y-f_y} b^{(k)}(m,n) \cdot \sum_{(x,y)=(\max(m-f_x, f_x), \max(n-f_y, f_y))}^{(\min(m+f_x, N_x-f_x), \min(n+f_y, N_y-f_y))} (CONV([\dots], H)_{(x,y)} \cdot H_{(x-m, y-n)}) \quad (13)$$

$$-2 \sum_{m=f_x}^{N_x-f_x} \sum_{n=f_y}^{N_y-f_y} b^{(k)}(m,n) \cdot (CONV([\dots], CONV(H, \tilde{H}, "full"))_{(m,n)}) = -2 \sum_{m=f_x}^{N_x-f_x} \sum_{n=f_y}^{N_y-f_y} W_{mn,ij} b^{(k)}_{(m,n)} \cdot S_{(m,n)} \quad (14)$$

where the last equality is given by Equation 9 and Equation 11. The local field $\tilde{h}_m^{(k)}$, for pixels (m, n) in the interior of the image and including the self-coupling term is given by:¹⁰

$$\tilde{h}_{mn}^{(k)} = \sum_{(i,j)=(f_x, f_y)}^{(N_x-f_x, N_y-f_y)} W_{mn,ij} b_{(m,n)}^{(k)} + S_{(m,n)} = CONV(b^{(k)}, \tilde{W}) + S_{(m,n)} \quad (15)$$

where $W_{mn,ij}$ denote the connection weights. Now if we have $W = -CONV(H, \tilde{H}, "full") = -c \cdot filter$, and thus $\tilde{W} = W$, then Equation 15 can be written as: $\tilde{h}_{mn}^{(k)} = CONV(b^{(k)}, -c \cdot filter) + S_{(m,n)}$. Comparing this equation with the implicit update rule in Equation 7 we will have:

$$b_{(m,n)}^{(k+1)} = st(2\tilde{h}_{mn}^{(k)}) \quad (16)$$

which for pixels in the interior of the image, has the same form as the standard Hopfield update rule. Due to the existence of nonzero self-coupling connections, this update rule will not minimize the quality function in Equation 10. Since $W_{mn, mn} = W_{0,0} = -c \cdot filter(0,0)$ the local field acting in the (m, n) pixel without the effect of self-coupling connection is given by:

$$h_{mn}^{(k)} = \tilde{h}_m^{(k)} + c \cdot filter(0,0) \cdot b_{mn}^{(k)} \quad (17)$$

The new update rule is:¹⁰

$$b_{(m,n)}^{(k+1)} = st(2h_{mn}^{(k)} + W_{mn, mn}) = st(2h_{mn}^{(k)} - (c \cdot filter(0,0))) \quad (18)$$

Using Equation 17 and Equation 18 the new update rule can be rewritten as:

$$b_{(m,n)}^{(k+1)} = st(2\tilde{h}_{mn}^{(k)} + 2c \cdot filter(0,0) \cdot b_{(m,n)}^{(k)} - (c \cdot filter(0,0))) \quad (19)$$

Using Equation 5 we can see that $g - RR^{(k+1)} = g - Q - M^{(k+1)}$ is equal to $\tilde{h}_{mn}^{(k)}$. This leads to the following update rule:

$$b^{(k+1)} = st(2g + 2c \cdot filter(0,0) \cdot b^{(k)} - 2Q - 2M^{(k+1)} - (c \cdot filter(0,0))) \quad (20)$$

The new image update rule implicates some changes in the algorithm. These changes are outlined in Figure 11. In the modified algorithm we also include the changes

needed in order to update the pixels more than once before we change quality function. The function Cost2 is used to terminate the pixel updating and $d_1^{(i)}$ can for example be a function of the quality function presented in Equation 10. The function Cost1 and the stop criterion 1 can be as before. The box called “New update rule” produces a halftoned image according to Equation 20.

References

1. R. W. Floyd and L. Steinberg, *Proc. SID* **17**, 75 (1976).
 2. D. Anastassiou and S. Kollas, *Visual Commun. Image Proc.* **1001**, 1062 (1988).
 3. M. Broja, F. Wyrowski and O. Bryngdahl, *Optics Comm.* **69**, 205 (1989).
 4. M. Analoui and J. P. Allebach, 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.
 5. F. Nilsson, Precomputed frequency modulated halftoning maps that meets the continuity criterion, *Proceedings of the IS&T International Conference on Digital Printing Technologies (NIP12)*, Oct. 1996, p. 72; **see pg. 176, this publication.**
 6. J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA* **79**:2554–2558, 1982.
 7. D. Anastassiou, Error diffusion coding for A/D conversion, *IEEE Transactions on Circuits and Systems*, Vol. **36** (9): 1175–1186, 1989.
 8. T. Tuttaß and O. Bryngdahl, Image halftoning: Fourier transform/neural net iteration. *Optics Communications*, Vol. **99** (1,2): 25–30, 1993.
 9. R. A. Ulichney, *Digital Halftoning* (MIT Press, 1987).
 10. M. Österberg, Two Optimization Approaches to Digital Halftoning. Technical report, Linköping University, ISY, S-58183 Linköping, 1996.
- * Previously published in *NIP12*, pp. 79–86, 1996.

