RC22849 (W0305-137) May 29, 2003
Electrical Engineering

IBM Research Report

A Unified Framework for Digital Halftoning and Dither Mask Construction: Variations on a Theme and Implementation Issues

Chai Wah Wu [1] **, *Gerhard Thompson*** [1] ***and Mikel Stanich*** [2]

[1] *IBM Research Div., P. O. Box 218, Yorktown Heights, NY 10598, USA*
[2] *IBM Printing Systems Div., 6300 Diagonal Highway, Boulder, CO 80301, USA*

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

A Unified Framework for Digital Halftoning and Dither Mask Construction: Variations on a Theme and Implementation Issues

Chai Wah Wu[1], *Gerhard Thompson*[1] *and Mikel Stanich*[2]
[1]*IBM Research Div., P. O. Box 218, Yorktown Heights, NY 10598, USA*
[2]*IBM Printing Systems Div., 6300 Diagonal Highway, Boulder, CO 80301, USA*

Abstract

We present a unified algorithmic framework for some classes of digital halftoning algorithms including Direct Binary Search (DBS) and dither mask generation algorithms such as Void and Cluster, BIPPSMA, and clustered dot with blue noise interpolation. Although these algorithms are different and used in different ways, e.g. Direct Binary Search is a global halftoning process, whereas dither masks are used in point halftoning processes, we show that they are all variations of a core algorithm. This makes it easier to compare the performance of these algorithms. Furthermore, by viewing these algorithms in the same framework, algorithmic extensions and implementation tricks and techniques among these algorithms can be more easily shared and their benefits exploited.

The core algorithm is essentially an optimization algorithm using pixel swapping where the cost function describes the perceptual difference between the halftone image and the color image when viewed at a distance. We compare various algorithms in the literature as they are cast in this framework. In particular, this framework allows us to derive a more efficient implementation of DBS.

1. Introduction

Most printers today can print in only a limited number of colors. Digital halftoning is a technique for converting a full color image into a halftone image which uses a limited number of colors. The halftone image is such that it appears to consist of many colors and resembles the full color image when viewed at a distance. For example, a picture of black and white dots can appear to display gray levels when viewed from some distance. In the rest of this paper we will refer to the full color image as the *input* image and the halftone image as the *output* image.

There are in general three classes of halftoning algorithms: *point processes* where each output pixel depends only on the value of the current input pixel, *neighborhood processes* such as error diffusion where each output pixel depends on the input and output pixels in a local neighborhood and *global processes* where each output pixel depends on all the input and output pixels. In fact, the direct halftoning algorithms considered in this paper are all global processes. The present paper is mainly concerned with halftoning algorithms for grayscale images where the output pixel is either black or white. We will represent a grayscale image as a matrix of real numbers between 0 and 1, with 0 denoting white and 1 denoting black. The color case can be dealt with by converting the color image into separate color channels (such as CMYK) and applying the grayscale algorithm to each channel. There are also methods for adapting bi-level output algorithms to the case of multiple output levels. In the sequel, a (halftone) *pattern* will be defined as a collection of black and white pixels on a rectangular grid, i.e. a matrix of 0's and 1's.

In point processes, the prevalent method is to use a dither mask to halftone an image and the goal is to design a dither mask which produces nice looking halftone patterns. The design of a dither mask is equivalent to designing N (usually 256) halftone patterns, one for each of the N possible gray values while satisfying the stacking constraint, i.e. the pixel locations of the black pixels in a pattern for a particular gray level must be a subset of those locations in a pattern for a darker gray[1]. Recently, there has been great interest in constructing blue noise or stochastic dither masks where the patterns have a dispersed random look. An excellent review of algorithms for constructing such dither masks can be found in [1]. If we view the construction of the dither mask as halftoning N different uniform gray images, then dither mask construction is itself a halftoning problem. This allows us to consider both the design of a dither mask and direct digital halftoning under the same framework.

[1]Although this constraint can be relaxed in lookup table type algorithms, care must be taken to produce correlated adjacent patterns to prevent sudden texture jumps. Correlated patterns are a natural consequence of the stacking constraint.

2. Image halftoning via restricted pixel swapping

2.1. Core algorithm

We present the core halftoning algorithm used to halftone an image directly or to construct a dither mask. The algorithm produces a halftone output image from a grayscale input image by swapping pixels of an initial output image. The set of pixels which can be swapped are restricted to satisfy some constraints. It can be viewed as an optimization algorithm where the goal is to minimize the perceptual difference between the halftone image and the grayscale image and pixels in the halftone image are swapped to reduce this difference. Pseudo-code of this algorithm is shown in Fig. 1. Given an image I, the algorithm attempts to generate a halftone pattern P such that the *error* or *cost* $E(P, I)$ is minimized. The swapping of pixels is accomplished by toggling a set of black pixels and a set of white pixels. The constraints are expressed by the set S_0, S_1 and S_2 and will be specified later depending on the application. In general, these sets depend on whether the algorithm is used to directly halftone an image or to create a dither mask, with further restrictions on S_1 and S_2 depending on the search heuristic used. As written in Fig. 1 the black pixels are first picked to be swapped with white pixels, although in some iterations, the roles of black and white pixels can be reversed, i.e. by interchanging the word "black" with "white" and the word "high" with "low".

1. Generate initial pattern P such that pixels in S_0 are black.
2. Calculate first filter error $L_1(P - I)$.
3. Select n black pixels in S_1 with high first filter error.
4. Toggle these n pixels resulting in pattern P'.
5. Calculate second filter error $L_2(P' - I)$.
6. Select n white pixels in S_2 with low first filter error.
7. Decide whether to swap or not. If so, toggle these n pixels in P' to obtain a new P. Otherwise, keep the old P.
8. Loop back to step 2 until stopping criteria are met.

Figure 1: *Pseudo-code of core halftoning algorithm.*

2.1.1. Image Halftoning

In this case, the image I is the grayscale image to be halftoned and S_0 is the empty set. The restricted sets S_1 and S_2 can be all the pixels in the image. However, to speed up implementation, the sets S_1 and S_2 are set to be significantly smaller (Sect. 3.3).

2.1.2. Dither mask construction

In this case, a collection of N uniform gray images I are halftoned sequentially[2], one for each graylevel $0 \leq g \leq 1$. The order in which these images are halftoned determines the sets S_1 and S_2. A possible order is to halftone the midtone gray first, and then successively halftone lighter and lighter grays. When that is finished, starting from the midtone gray, darker and darker grays are halftoned. Other ordering schemes such as binary tree partitioning are also possible [1]. To satisfy the stacking constraint, S_0, S_1 and S_2 for halftoning graylevel g are determined as follows. Let g_a and g_b be the graylevels closest to g which have already been halftoned such that $g_a < g < g_b$. Then S_0 are the black pixel locations of the pattern for g_a. The sets S_1 and S_2 are both defined as the pixel locations which are black in the pattern for g_b but white in the pattern for g_a. Depending on the search heuristic and the algorithm, additional restrictions on S_1 and S_2 are imposed (Sect. 3).

2.2. Detailed explanation of core algorithm

Initial pattern generation (Step 1 in Fig. 1)

To generate an initial pattern, first it is determined how many black pixels (say m) should be in the pattern in addition to the pixels in S_0. Generally this is determined to be such that the average gray level of the initial pattern is close to the average gray level of the image I, possibly taking into account the tone reproduction curve and dot gain considerations [2]. Two ways to put m black pixels in S_1, are either by randomly selecting m locations in S_1 or by running Fig. 1 m times with $n = 1$, each time adding one black pixel. When a pattern is initialized this second way it can be adequate enough to be used as a final pattern without running any iterations of Fig. 1 which will be referred to as a *greedy* approach.

Selecting pixels in S_1 and S_2 to swap

One possible way is to pick the n pixels with the highest filter error and swap them with the n pixels with the lowest second filter error. By not requiring them to have the highest (lowest) filter error, other heuristics for picking the n pixels in S_1 and S_2 are possible which potentially take less time and generate better patterns.

Calculating filter error

The choice of the cost function $E(P, I)$ is used to model the perceptual difference between the grayscale image I and the halftone pattern P and should take into account the human visual system (HVS). To make the problem tractable, usually it is assumed that the HVS acts as a linear shift-invariant low-pass filter. Some examples of such HVS based filters can be found in [3, 4]. Other non-HVS based filters such as Gaussian filter [5] or Butterworth filter [6] have also been used. In addition to HVS, the dot

[2]Threshold swapping methods to optimize several patterns at once have not been very successful [1].

gain characteristics and printer spot profile of the printer model can also be incorporated into the filter, e.g. $L_1 = L_{hvs} \circ L_{spot}$.

The filters L_1 and L_2 are used in Fig. 1 to minimize the cost function E. In general they can be chosen as the filters described above, and a priori they don't have to be the same as the filters used in defining E. When $E(P, I)$ is chosen as $\|L_{hvs}(P - I)\|_2$, there is a relationship between the filters L_{hvs} and L_1 and L_2 (Sect. 3.3) which allows E to be decreased at each iteration. The filter error $L(P - I)$ is the convolution between the impulse responses of L and $P - I$. For dither mask construction the convolution is circular to account for the fact that the dither mask is tiled periodically in the halftoning process. Since the FFT is directly suitable for performing circular convolution, algorithms such as BIPPSMA utilize FFT for this purpose. On the other hand, since the difference of the halftone patterns between iterations is at most $2n$ pixels, very efficient methods for computing the filter errors from the filter errors of the previous iteration are possible, provided the support of the filter impulse response is small. Furthermore, when $L_1 = L_2$, the difference between P and P' is n pixels and thus the second filter error $L_2(P' - I)$ can be obtained from the first filter error by adding (or subtracting) the impulse response centered at these pixels. Similarly the first filter response in the next iteration can be obtained from the second filter error in this way. Also note that for dither mask construction, I is a uniform gray image, and the search for the pixels with the highest (or lowest) filter error $L(P - I)$ is equivalent to the search for pixels with the highest (or lowest) $L(P)$, i.e. $L(P)$ can be used as the filter error.

Stopping criteria

The stopping criteria for exiting the loop in Fig. 1 can be when the maximum number of iterations are reached or when the change in error is small between successive iterations. A double loop can also be used where in the inner loop the number n is decreased at each iteration until it reaches 1 and in the outer loop the number n is reset to a specified value as is the case in BIPPSMA.

3. Comparison with other algorithms

Many algorithms for creating dither masks or for direct halftoning fall under the class of algorithms in Fig. 1.

3.1. Simulated annealing

In this approach [1], $n = 1$, and the decision to swap pixels (Step 7 in Fig. 1) is only accepted if there is a reduction in cost or if the increase in cost is below a random threshold, with the average magnitude of this threshold decreasing as time goes on. The idea is to initially accept some swaps which increase the cost in order to move out of local minima, with the number of such swaps decreasing as time goes on (and the system "cools").

3.2. Void and Cluster

The Void and Cluster algorithm [5] uses a Gaussian filter for L_1 and L_2 and uses the second way for initializing the pattern (Sect. 2.2). After the first pattern is generated with Fig. 1, the rest of the patterns are generated with the greedy approach. The pixel in S_1 with the highest filter error is swapped with the pixel in S_2 with the lowest filter error. For the first pattern, S_1 and S_2 are all the pixels. This choice of S_1 can lead to a suboptimal local minimum of the cost function that can be improved upon by choosing S_1 to be a single pixel and S_2 a local neighborhood as is done in DBS [7].

3.3. Direct Binary Search (DBS)

In DBS, the cost E to be minimized is $\|L_{dbs}(P - I)\|_2$. The pixels are traversed in some order and for each pixel, a neighborhood of pixels is examined to determine which pixel can be swapped with the current pixel which reduces E the most[3]. This corresponds to setting S_1 equal to a single pixel and S_2 to be a neighborhood of S_1. The change in E due to swapping a pixel p_0 with a pixel p_1 is [8]:

$$\Delta E = 2(c_{pp}(0) - ac_{p\tilde{e}}(p_0) + ac_{p\tilde{e}}(p_1) - c_{pp}(p_0 - p_1))$$

where $a = 1$ if pixel p_0 is black and pixel p_1 is white and $a = -1$ if pixel p_0 is white and pixel p_1 is black. c_{pp} is the autocorrelation function of L_{dbs} and $c_{p\tilde{e}}$ is the correlation between L_{dbs} and $L_{dbs}(P - I)$. If a swap is accepted[4], $c_{p\tilde{e}}$ is updated by:

$$c_{p\tilde{e}}(p) \leftarrow c_{p\tilde{e}}(p) - ac_{pp}(p - p_0) + ac_{pp}(p - p_1)$$

Given a fixed pixel location p_0, a neighborhood $\mathcal{N}$ of p_0 is searched for a suitable pixel p_1 to swap with. Since the first two terms in ΔE are independent of p_1, this search can be accomplished by minimizing $w(p) = ac_{p\tilde{e}}(p) - c_{pp}(p_0 - p)$, i.e. $p_1 = \arg\min_{p \in \mathcal{N}} w(p)$. A swap is accepted if $\Delta E < 0$, or equivalently if $w(p_1) < w(p_0)$. Since c_{pp} is symmetric, the update of $c_{p\tilde{e}}$ can be written as $c_{p\tilde{e}}(p) \leftarrow a(w(p) + c_{pp}(p - p_1))$. It's easy to see that by considering $c_{p\tilde{e}}$ as the first filter error, $aw(\cdot)$ as the second filter error and c_{pp} as the impulse response of $L_1 = L_2$, this can be cast in the framework of Fig. 1. Similar to Void and Cluster, the pixel with the highest first filter error is swapped with the pixel with the lowest second filter error. In this case $L_1 = L_2$ in Fig. 1 is the autocorrelation function of the filter L_{dbs} used in defining E.

This approach has several speed and performance advantages over DBS as described in [8]. First, to evaluate

[3] DBS allows toggling of pixels, but it is not preferred in creating dither masks as the number of black pixels is known in advance. We concentrate mainly on pixel swapping due to limited space, but pixel toggling can be easily included. For instance, it is easy to show that toggling pixel p_0 reduces E more than swapping p_0 with p_1 if $w(p_1) > -\frac{1}{2}c_{pp}(0)$.

[4] In [8] heuristics were given to decide whether to swap or not, e.g. only one swap is allowed per block of pixels considered.

the effect of a swap takes less table lookups and additions than in DBS. Second, the computation of $w(p)$ used in finding the pixel p_1 to swap can be reused in the update of $c_{p\tilde{e}}$. Third, in [8], the search neighborhood $\mathcal{N}$ is fairly small (5 pixels) to speed up computation. The approach presented here allows this neighborhood to be enlarged to be the support of the filter anytime a swap is accepted without much increase in computation. In other words, once a swap between p_0 and p_1 is accepted, the choice of p_1 can be refined which can result in a larger decrease of E.

3.4. BIPPSMA

In this algorithm [6], the FFT (and pointwise multiplication) is used to perform the circular convolution. The main difference between BIPPSMA and other algorithms is that the second filter error is the same as the first filter error $L_1(P - I)$ as opposed to $L_2(P' - I)$ in the other algorithms, i.e. the calculation of P' (Step 4 in Fig. 1) is not needed. At each inner iteration the number of pixels swapped n is reduced by half until $n = 1$.

3.5. Supercell clustered dot dither masks with blue noise interpolation

In [2], the visually pleasant patterns in a supercell clustered dot dither mask (or a mask created by replicating single clustered dot cells) are chosen. These patterns are fixed and the remaining patterns are created as in Sect. 2.1.2. This results in a dither mask that preserves the look of a clustered dot dither while reducing the periodic artifacts in common supercell techniques. Furthermore, dot gain characteristics can be incorporated into the dither mask obviating the need for a separate tone reproduction curve.

3.6. Particle repulsion

A characteristic of stochastic dither patterns is that the black pixels are as far away from each other as possible. This suggests a physical model of particles with repulsive forces between them. The particles settle to an equilibrium which is the desired pattern. This can be cast into our framework if we consider $L(P)$ as the *potential* of the particles [9]. The filter impulse response is then the potential due to a single particle[5]. $L(I)$ corresponds to another potential which is subtracted from the particles potential $L(P)$ to form the *total potential* of the system $V_t = L(P - I)$. The equilibrium occurs when no forces act on the particles, i.e. the force field $\vec{\mathbf{E}} = -\nabla V_t = 0$ at the particles. A sufficient condition for this is when V_t is constant, which supports Void and Cluster's aim of minimizing the maximum variation of V_t. By adding a suitable constant, this equilibrium can be assumed to be $V_t = 0$, which can be achieved by minimizing the cost $E = \|V_t\| = \|L(P-I)\|$. The particles move towards this equilibrium and for small timesteps the particles move a small distance, corresponding to swapping a black pixel with a white pixel in a small neighborhood. Thus the motion of the particles can be mimicked in Fig. 1 by making S_1 to be all the black pixels and S_2 to be the local neighborhoods of the black pixels with the additional constraint that each black pixel can only swap with a white pixel in its neighborhood.

[5] For example, in the case of electrostatic force, $V(x, y) = \frac{q/4\pi\epsilon_0}{\sqrt{x^2+y^2}}$.

4. Conclusions

We have presented a framework for digital halftoning under which several classes of blue noise related halftoning algorithms fall. One benefit of this framework is that implementation details such as filter choice, filter errors update method, search heuristic, pattern initialization, and stopping criteria can be more easily shared among these algorithms. In particular, by using this framework, we presented an implementation of DBS which has some performance benefits over previous implementations.

References

[1] K. E. Spaulding, R. L. Miller, and J. Schildkraut, "Methods for generating blue-noise dither matrices for digital halftoning," *J. of Electronic Imaging*, vol. 6, no. 2, pp. 208–230, 1997.

[2] C. W. Wu, C. P. Tresser, G. R. Thompson, and M. J. Stanich, "Supercell dither masks with constrained blue noise interpolation," *Proc. IS&T's NIP17: Int. Conf. on Digital Printing Tech.*, pp. 487–490, 2001.

[3] R. Näsänen, "Visibility of halftone dot textures," *IEEE Tr. Syst. Man, Cyb.*, vol. 14, no. 6, pp. 920–924, 1984.

[4] J. Sullivan, L. Ray, and R. Miller, "Design of minimum visual modulation halftone patterns," *IEEE Tr. Syst. Man, Cyb.*, vol. 21, no. 1, pp. 33–38, 1991.

[5] R. Ulichney, "The void-and-cluster method for dither array generation," *Proc. of SPIE*, vol. 1913, pp. 332–343, 1993.

[6] M. Yao and K. J. Parker, "Modified approach to the construction of the blue noise mask," *J. of Electronic Imaging*, vol. 3, no. 1, pp. 92–97, 1994.

[7] D. J. Lieberman and J. P. Allebach, "On the relation between DBS and Void and Cluster," *IS&T's NIP 14: Int. Conf. on Dig. Print. Tech.*, pp. 290–293, 1998.

[8] J. P. Allebach, "DBS: retrospective and future directions," *Proc. of SPIE*, vol. 4300, pp. 358–376, 2001.

[9] G. R. Thompson, C. P. Tresser and C. W. Wu, US Patents 5,917,951 and 6,025,930.