Direct Binary Search (DBS) algorithm with constraints

Conference Paper *in* Proceedings of SPIE - The International Society for Optical Engineering · February 2013

DOI: 10.1117/12.2002554

Direct Binary Search (DBS) algorithm with constraints

Kartheek Chandu[†], Mikel Stanich[†], Chai Wah Wu[*] and Barry Trager[*]

[†] Ricoh Production Print Solutions, LLC, 6300 Diagonal Highway, Boulder, CO 80301, USA
[*] IBM T. J. Watson Research Center, P. O. Box 218, Yorktown Heights, NY 10598, USA

ABSTRACT

In this paper, we describe adding constraints to the Direct Binary Search (DBS) algorithm. An example of a useful constraint, illustrated in this paper, is having only one dot per column and row. DBS with such constraints requires greater than two toggles during each trial operation. Implementations of the DBS algorithm traditionally limit operations to either one toggle or swap during each trial. The example case in this paper produces a wrap-around pattern with uniformly distributed ON pixels which will have a pleasing appearance with precisely one ON pixel per each column and row. The algorithm starts with an initial continuous tone image and an initial pattern having only one ON pixel per column and row. The auto correlation function of Human Visual System (HVS) model is determined along with an initial perceived error. Multiple operation pixel error processing during each iteration is used to enforce the one ON pixel per column and row constraint. The constraint of a single ON pixel per column and row is used as an example in this paper. Further modification of the DBS algorithm for other constraints is possible, based on the details given in the paper. A mathematical framework to extend the algorithm to the more general case of Direct Multi-bit Search (DMS) is presented.

Keywords: Direct Binary Search, DBS, Halftone, Inkjet printing, flushing

1. INTRODUCTION

In recent years, inkjet printing technology has enabled use of printing as a manufacturing fabrication technique to create displays, solar panels, electronic boards, human organs and 3-D lithographic printing.[1] Along with this evolution, inkjet printing also has brought new trends in business and commercial printing. This include wide format printers, ultra-high speed production printers, digital textile printers, and packaging related applications.

Many present high speed production inkjet printers employ fixed-array printheads, capable of printing on paper webs greater than 20 inches wide. The fixed-array system is constructed from multiple printhead modules which are stitched together edge-to-edge. The array remains stationary while the media moves past the printheads, creating a printing process capable of operating at thousands of feet per minute. Previously, each printhead module in the fixesd-array contained hundreds of nozzles and could print less than one-inch width. Printheads wider than one-inch are now being used, having thousands of nozzles. A significant problem with inkjet printheads is clogging of nozzles and the greater the number of nozzles in a printhead the larger the problem becomes. Factors which influence clogging include drying of ink in the nozzle, accumulated paper dust at the nozzle plate, increase in ink viscosity, and intrusion of air bubbles. Any of these factors can stop a nozzle from functioning creating a “jet out”, which results in no data being printed by that nozzle. The artifact produced by a jet out is a white streak on the printed output. Clogged nozzles in some cases may be recovered by cleaning. If a clogged nozzle is left uncleaned for a prolonged period of time, there will be only a small probability of recovering that nozzle. In addition clogging may propagate, leading to clogging of its neighboring nozzles. A single clogged nozzle might require replacement of the entire fixed-array printhead, depending on configuration of the individual printhead modules.

Over a wide range of print jobs, sheet content alone is not adequate to keep the nozzles from clogging. While variable data printing will have different content on every sheet, nozzles at the edges or between pages may be used infrequently or not at all. Monochrome printing, employing on the black channel, may similarly cause

Further author information:
K.C.: E-mail: kartheek.chandu@ieee.org, Telephone: 1 720 663 3621
C.W.W.: E-mail: chaiwahwu@ieee.org, Telephone: 1 914 945 1567

Color Imaging XVIII: Displaying, Processing, Hardcopy, and Applications,
edited by Reiner Eschbach, Gabriel G. Marcu, Alessandro Rizzi, Proc. of SPIE Vol. 8652
86520K · © 2013 SPIE-IS&T · CCC code: 0277-786X/13/$18 · doi: 10.1117/12.2002554

complete set of other color printheads to not be used. Clogging can be significantly improved by forcefully ejecting ink drops from each nozzle at a given frequency on a continuous basis. This procedure is referred to as "flushing". A common method of forcefully ejecting ink drops from each nozzle prints a straight line across the entire web at the beginning of each page. This technique is known as "line flushing". Another flushing method, commonly called "random flushing", prints drops from each nozzle in a random fashion, producing a noisy background. Many different flushing methodologies exists in literature.[2–5] A method to reduce the visibility of the flushing pattern uses intelligent flushing to hide flushing under print data. None of these published descriptions have discussed a framework to create flushing patterns. Therefore an improved method is required to create a visually pleasing flushing pattern optimized to produce the required frequency.

In this paper, we discuss an algorithm to produce a visually pleasing flushing mask using the proposed Direct Binary Search (DBS)[6,7] algorithm with constraints. For example, assume that the specification for preventing the nozzles from drying is to eject one drop for every one inch from each nozzle. This requires a mask printing one PEL over a one inch distance. A $one-inch \times one-inch$ mask with a one dot per column constraint satisfies this requirement. Logically, the single dot per column constraint also results in a single dot per row. The resulting mask has the same constraint in both directions, permitting it to be used in any rotational orientation or flipped. A single mask design is replicated across the entire print width (scan direction) to create a mask for a fixed-array system of any size. A single mask with a given orientation is employed to create the web width mask. This is because the design approach applies circular convolution to implement wraparound so that a mask seamlessly matches the adjacent mask, avoiding tiling artifacts. Since the mask size is one inch, the pattern will be printed repeatedly every inch in the process direction. This produces a mask with a uniform distribution of PELS having a "random flushing" pattern. It has a pleasing appearance and the constraint guarantees that every nozzle ejects exactly one drop over every inch in both directions. The proposed algorithm can also be used for other applications, such as generating a halftone seed pattern.

In the current section we introduce the content and motivations of the paper. In Section 2 we present the background of DBS and in Section 3 we describe the proposed algorithm. Finally the results are covered in Section 4.

2. DIRECT BINARY SEARCH (DBS) BACKGROUND

DBS produces binary Frequency Modulation (FM) halftone designs. Binary FM halftoning[8] fixes the dot size, rendering different tones by varying the frequency of the dots. Commonly used FM halftoning algorithms include dispersed dot screening,[9] error diffusion,[10–12] and search-based halftone methods such as DBS[7] and DMS.[13] The DBS algorithm is an iterative/recursive search heuristic that uses a perceptual filter, such as a Human Visual System (HVS) model, to minimize a perceived difference between a continuous tone image and its corresponding halftone image by searching for the best possible configuration of available drop sizes to create the halftoned image. Iterative methods are the most computationally intensive of all digital halftoning methods. They yield visually pleasing output quality which is significantly better than ordered dithering and error diffusion. In this paper, we will use notations similar to Allebach et. al paper.[7]

DBS can be thought of as an iterative/recursive optimization, which is used to minimize a cost function $\boldsymbol{\varepsilon}$, defined as the error between the perceived halftone image and the perceived continuous tone image:

$$\boldsymbol{\varepsilon} = |h(x,y) * *g(x,y) - h(x,y) * *f(x,y)|^2 dxdy, \tag{1}$$

where $**$ denotes 2-dimensional convolution, $h(x,y)$ represents the Point Spread Function (PSF) of the human visual system, $f(x,y)$ is the continuous tone original image and $g(x,y)$ is the corresponding rendered halftone image, having levels which are either 0 (white) or 1 (black). The halftone image itself incorporates a printer model

$$g(x,y) = \sum_m \sum_n g[m,n] p(x - mX, y - nX), \tag{2}$$

which represent the combination of $g[m,n]$ with a spot profile $p(x,y)$ having a device PEL spacing X, where X is the inverse of the printer addressability. Printer addressability is expressed as Dots Per Inch (DPI). Superposition is assumed in this model for the interaction between overlapping spots. The digital halftone image $g[m,n]$ can have absorptance value either 0 (white) or 1 (black).

DBS is a computationally expensive algorithm that requires several iterations through an image before converging to the final halftone. DBS starts with an initial halftone image and a local improvement in the halftone is produced by swapping and toggling PELs in the initial halftone image. "Swapping" is defined as the operation of switching the absorptance values of nearby pixels and "toggling" is defined as the operation of changing the absorptance value of individual pixels. The cost function $\boldsymbol{\varepsilon}$ can be represented as

$$\boldsymbol{\varepsilon} =< \tilde{e}, \tilde{e} >, \tag{3}$$

where $< \cdot, \cdot >$ denotes the inner product and

$$\tilde{e}(x,y) = h(x,y) ** (g(x,y) - f(x,y)), \tag{4}$$

represents the perceptually filtered error. We will assume that the continuous-tone image $f(x,y)$ can also be expressed in terms of its samples $f[m,n]$ where (m,n) are coordinates for the PELs in the halftone array, using the same structure as given above for the halftone image in terms of the printer spot function. Then the perceived error is given by

$$\tilde{e}(x,y) = \sum_{m,n} e[m,n]\tilde{p}(x - mX, y - nX), \tag{5}$$

where,

$$e[m,n] = g[m,n] - f[m,n], \tag{6}$$

and $\tilde{p}(x,y) = h(x,y) ** p(x,y)$ is the perceived printer spot profile. Considering the effect of a trial change, the new perceived error will be

$$\tilde{e}' = \tilde{e} + \Delta\tilde{e}. \tag{7}$$

Therefore, the new cost function $\boldsymbol{\varepsilon'}$ is represented as

$$\boldsymbol{\varepsilon'} =< \tilde{e}', \tilde{e}' > . \tag{8}$$

Substituting Eq.(7) in Eq.(8) and expanding the inner product, we obtain

$$\boldsymbol{\varepsilon}' = \boldsymbol{\varepsilon} + 2 < \Delta\tilde{e}, \tilde{e} > + < \Delta\tilde{e}, \Delta\tilde{e} >, \tag{9}$$

assuming all signals are real-values. Either a toggle at pixel (m_0, n_0) or a swap between pixels (m_0, n_0) and (m_1, n_1) can be represented as

$$g'[m,n] = g[m,n] + a_0\delta[m - m_0, n - n_0] + a_1\delta[m - m_1, n - n_1]. \tag{10}$$

The above equation can be generalized to

$$g'[m,n] = g[m,n] + \sum_i a_i\delta[m - m_i, n - n_i]. \tag{11}$$

Substituting this into Eq.(6) and using the result in Eq.(5), we obtain

$$\Delta\tilde{e}(x,y) = \sum_i a_i\tilde{p}(x - m_iX, y - n_iX). \tag{12}$$

Finally, substituting Eq.(12) into Eq.(9), we obtain

$$\Delta\boldsymbol{\varepsilon} = 2\sum_i a_i c_{\tilde{p}\tilde{e}}[m_i, n_i] + \sum_{i,j} a_i a_j c_{\tilde{p}\tilde{p}}[m_i - m_j, n_i - n_j], \tag{13}$$

where,

$$c_{\tilde{p}\tilde{e}}[m,n] =< \tilde{p}(x,y), \tilde{e}(x + mX, y + nX) >, \tag{14}$$

and

$$c_{\tilde{p}\tilde{p}}[m,n] =< \tilde{p}(x,y), \tilde{p}(x+mX, y+nX) > . \tag{15}$$

The value of a_i represents the amount of change in the gray level for a toggle as

$$a_i = g_{new}[m_i, n_i] - g_{old}[m_i, n_i]. \tag{16}$$

A swap between pixels i and j is equivalent to two toggles with $g_{new}[m_i, n_i] = g_{old}[m_j, n_j]$ and $g_{new}[m_j, n_j] = g_{old}[m_i, n_i]$ which can be written as

$$a_i = g_{old}[m_j, n_j] - g_{old}[m_i, n_i], \tag{17}$$

and

$$a_j = g_{old}[m_i, n_i] - g_{old}[m_j, n_j]. \tag{18}$$

Thus, $a_j = -a_i$ except for $j = 0$ (e.g., toggle, $a_0 = 0$).

DBS is heavily dependent on the HVS model. DBS yields different halftone patterns depending on the HVS model employed. In our work, a richer class of HVS model is implemented that yields enhanced halftoning results. The HVS model used is a mixed Gaussian model proposed by Kim and Allebach[14] whose functional form is

$$c_{\tilde{p}\tilde{p}}[u,v] = k_1 exp(-\gamma/2\sigma_1^2) + k_2 exp(-\gamma/2\sigma_2^2), \tag{19}$$

where $\gamma = u^2 + v^2$ and $k_1, k_2, \sigma_1, \sigma_2$ are constants. Assuming that $c_{\tilde{p}\tilde{p}}$ is symmetric, then Eq.(13) can be written as

$$\Delta\boldsymbol{\varepsilon} = 2(\sum_i a_i c_{\tilde{p}\tilde{e}}[m_i, n_i] + \sum_{i<j} a_i a_j c_{\tilde{p}\tilde{p}}[m_i - m_j, n_i - n_j]) + \sum_i a_i^2 c_{\tilde{p}\tilde{p}}[0,0]. \tag{20}$$

In each iteration, the operation that yields the maximum error decrease in $\Delta\boldsymbol{\varepsilon} < 0$ among all toggles or swaps is selected. The final "accepted operation", which produces the maximum error decrease, requires updating $g(m,n)$ and $c_{\tilde{p}\tilde{e}}$ using

$$c_{\tilde{p}\tilde{e}}[m,n]' = c_{\tilde{p}\tilde{e}}[m,n] + \sum_i a_i c_{\tilde{p}\tilde{p}}[m - m_i, n - n_i]. \tag{21}$$

The end criteria is met when no accepted operations are performed during the last iteration.

3. PROPOSED ALGORITHM

Published literature has limited trial changes to only one or two toggle operations during each iteration. Note that a swap can be considered equivalent to two toggle operations. There are instances where more than two toggles are necessary. For instance, assume that the desired pattern contains only one dot per column as explained in Section 1. The proposed algorithm starts with an initial pattern containing one dot per column. The algorithm performs multiple swap operations, using DBS at every iteration, preserving the one dot per column constraint.

Columns

Rows	1	2	3	4	5	6	7	8
1	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0
4	0	0	0	1	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	1

Figure 1. Initial seed pattern $g(m,n)$ for a 8×8 size mask

In the DBS algorithm, the change in error $\Delta\boldsymbol{\varepsilon}$ resulting from a single swap between pixels (m_0, n_0) and (m_1, n_1) simplifies Eq.(20) to

$$\Delta\boldsymbol{\varepsilon} = 2(a_0 c_{\tilde{p}\tilde{e}}[m_0, n_0] + a_1 c_{\tilde{p}\tilde{e}}[m_1, n_1] + a_0 a_1 c_{\tilde{p}\tilde{p}}[m_0 - m_1, n_0 - n_1]) + (a_0^2 + a_1^2) c_{\tilde{p}\tilde{p}}[0, 0]. \tag{22}$$

We have expanded this equation to perform two swap operations i.e. four toggles during each iteration. The change in error $\Delta\boldsymbol{\varepsilon}$ from swapping between (m_0, n_0) and (m_1, n_1), and also between (m_2, n_2) with (m_3, n_3) during the same iteration expands Eq.(20) to

$$\begin{aligned}\Delta\boldsymbol{\varepsilon} = 2 * (a_0 c_{\tilde{p}\tilde{e}}(m_0, n_0) + a_1 c_{\tilde{p}\tilde{e}}(m_1, n_1) + a_2 c_{\tilde{p}\tilde{e}}(m_2, n_2) + a_3 c_{\tilde{p}\tilde{e}}(m_3, n_3)\\ + a_0 a_1 c_{\tilde{p}\tilde{p}}(m_0 - m_1, n_0 - n_1) + a_0 a_2 c_{\tilde{p}\tilde{p}}(m_0 - m_2, n_0 - n_2)\\ + a_0 a_3 c_{\tilde{p}\tilde{p}}(m_0 - m_3, n_0 - n_3) + a_1 a_2 c_{\tilde{p}\tilde{p}}(m_1 - m_2, n_1 - n_2)\\ + a_1 a_3 c_{\tilde{p}\tilde{p}}(m_1 - m_3, n_1 - n_3) + a_2 a_3 c_{\tilde{p}\tilde{p}}(m_2 - m_3, n_2 - n_3))\\ + (a_0^2 + a_1^2 + a_2^2 + a_3^2) c_{\tilde{p}\tilde{p}}(0, 0),\end{aligned} \tag{23}$$

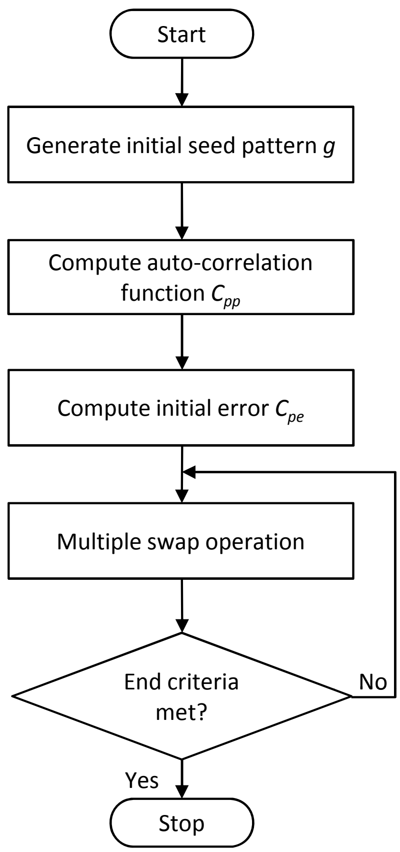


Figure 2. DBS Flowchart

The algorithm starts with an initial seed pattern g of size $N \times N$ with $g(m, n) = 1$, with $m = n$ and $m = 1, 2, 3, ...N$. This seed pattern is shown in Fig.1 where $N = 8$. In this paper, we describe the algorithm using binary values of either 0 (OFF) or 1 (ON). While this is not described in this paper, the concepts can be extended to the more general "multi-bit" case using the Direct Multi-bit Search (DMS)[13] algorithm. This is applicable to printers capable of producing S possible different output drop sizes with absorptance levels $\phi_1, \phi_2, ..., \phi_S$.

A continuous tone image f of size $N \times N$, with all pixel values at a constant gray level, is used for the objective function. Then, the auto-correlation function $c_{\tilde{p}\tilde{p}}[m,n]$ is computed using Eq.(19) and the initial error $c_{\tilde{p}\tilde{e}}[m,n]$ is computed using Eq.(14). The multiple swap operation is performed on all ON pixels in $g(m,n)$, until the end criteria is met. A flowchart describing this process is shown in Fig.2.

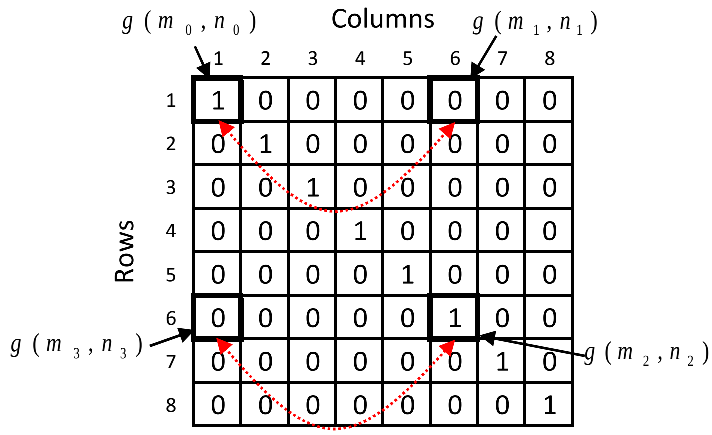


Figure 3. Initial seed pattern with dual swap operation

An ON pixel is identified in the first column col_α, in the multiple swap operation, where $col_\alpha = 1, 2, 3, ...N$. Let the identified ON pixel location be (m_0, n_0) in row row_α. An ON pixel is then identified in another column col_β, where $col_\beta = 1, 2, 3, ...N$. The identified ON pixel location will be (m_2, n_2) in row row_β. Note that the one dot per column constraint requires the ON pixel at (m_0, n_0) can only be swapped with a OFF pixel location at (m_1, n_1) in the same row row_α i.e. $m_0 = m_1 = row_\alpha$, similarly, $m_2 = m_3 = row_\beta$. The OFF pixel location at (m_1, n_1) should be in the same column as (m_2, n_2) i.e. $n_1 = n_2 = col_\beta$, similarly, $n_0 = n_3 = col_\alpha$. For example in Fig.3, the $col_\alpha = 1$ and (m_0, n_0) is at $(1, 1)$, and $col_\beta = 6$ and (m_2, n_2) is at $(6, 6)$. The ON pixel at $(1, 1)$ can only be swapped with OFF pixel at $(1, 6)$ and the ON pixel at $(6, 6)$ can only be swapped with $(6, 1)$ to maintain the one dot per column constraint. Using the DBS algorithm with this assumption, this can be treated as column swapping. Therefore, using Eqs.(17) and (18), the values of a_0 and a_1 are -1 and 1 respectively, and the values of a_2 and a_3 are -1 and 1 respectively. Substituting a_0, a_1, a_2 and a_3 values in Eq.(23), the $\Delta\varepsilon$ is simplified to

$$\begin{aligned}\Delta\varepsilon = 2 * (&-c_{\tilde{p}\tilde{e}}(row_\alpha, col_\alpha) + c_{\tilde{p}\tilde{e}}(row_\alpha, col_\beta) - c_{\tilde{p}\tilde{e}}(row_\beta, col_\beta) + c_{\tilde{p}\tilde{e}}(row_\beta, col_\alpha)\\ &-c_{\tilde{p}\tilde{p}}(0, col_\alpha - col_\beta) + c_{\tilde{p}\tilde{p}}(row_\alpha - row_\beta, col_\alpha - col_\beta)\\ &-c_{\tilde{p}\tilde{p}}(row_\alpha - row_\beta, 0) - c_{\tilde{p}\tilde{p}}(row_\alpha - row_\beta, 0)\\ &+c_{\tilde{p}\tilde{p}}(row_\alpha - row_\beta, col_\beta - col_\alpha) - c_{\tilde{p}\tilde{p}}(0, col_\beta - col_\alpha))\\ &+4 * c_{\tilde{p}\tilde{p}}(0,0),\end{aligned} \tag{24}$$

which further simplifies to

$$\begin{aligned}\Delta\varepsilon = 2 * (-c_{\tilde{p}\tilde{e}}(row_\alpha, col_\alpha) + c_{\tilde{p}\tilde{e}}(row_\alpha, col_\beta) - c_{\tilde{p}\tilde{e}}(row_\beta, col_\beta) + c_{\tilde{p}\tilde{e}}(row_\beta, col_\alpha))\\ +4 * (c_{\tilde{p}\tilde{p}}(row_\alpha - row_\beta, col_\alpha - col_\beta) - c_{\tilde{p}\tilde{p}}(0, col_\alpha - col_\beta) - c_{\tilde{p}\tilde{p}}(row_\alpha - row_\beta, 0) + c_{\tilde{p}\tilde{p}}(0,0)).\end{aligned} \tag{25}$$

This process is repeated for all col_β columns. The change in error $\Delta\varepsilon$ is calculated for the multiple swap operation performed between ON pixel in each column col_β with the ON pixel in column col_α. The operation that yields the maximum decrease in error, minimum in $\Delta\varepsilon < 0$, is selected. Every accepted operation requires updating $c_{\tilde{p}\tilde{e}}$ using Eq.(21) and $g(m,n)$ for (m_0, n_0), (m_1, n_1), (m_2, n_2) and (m_3, n_3). The next step advances col_α to the next column. The entire multiple swap process is repeated with the ON pixel in each column col_β. This process is illustrated using a flowchart in Fig.4. When all col_α columns are processed, the algorithm tests for the end criteria of having no operations performed in the last iteration. If the end criterion has not been met, the algorithm starts again from the beginning and repeats until the end criteria is met.

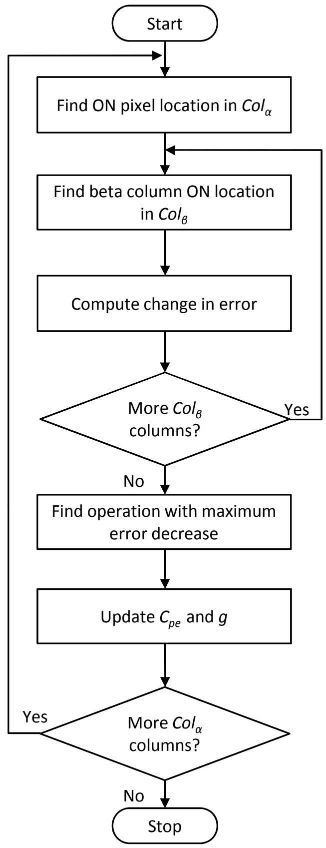


Figure 4. Dual operation DBS flowchart

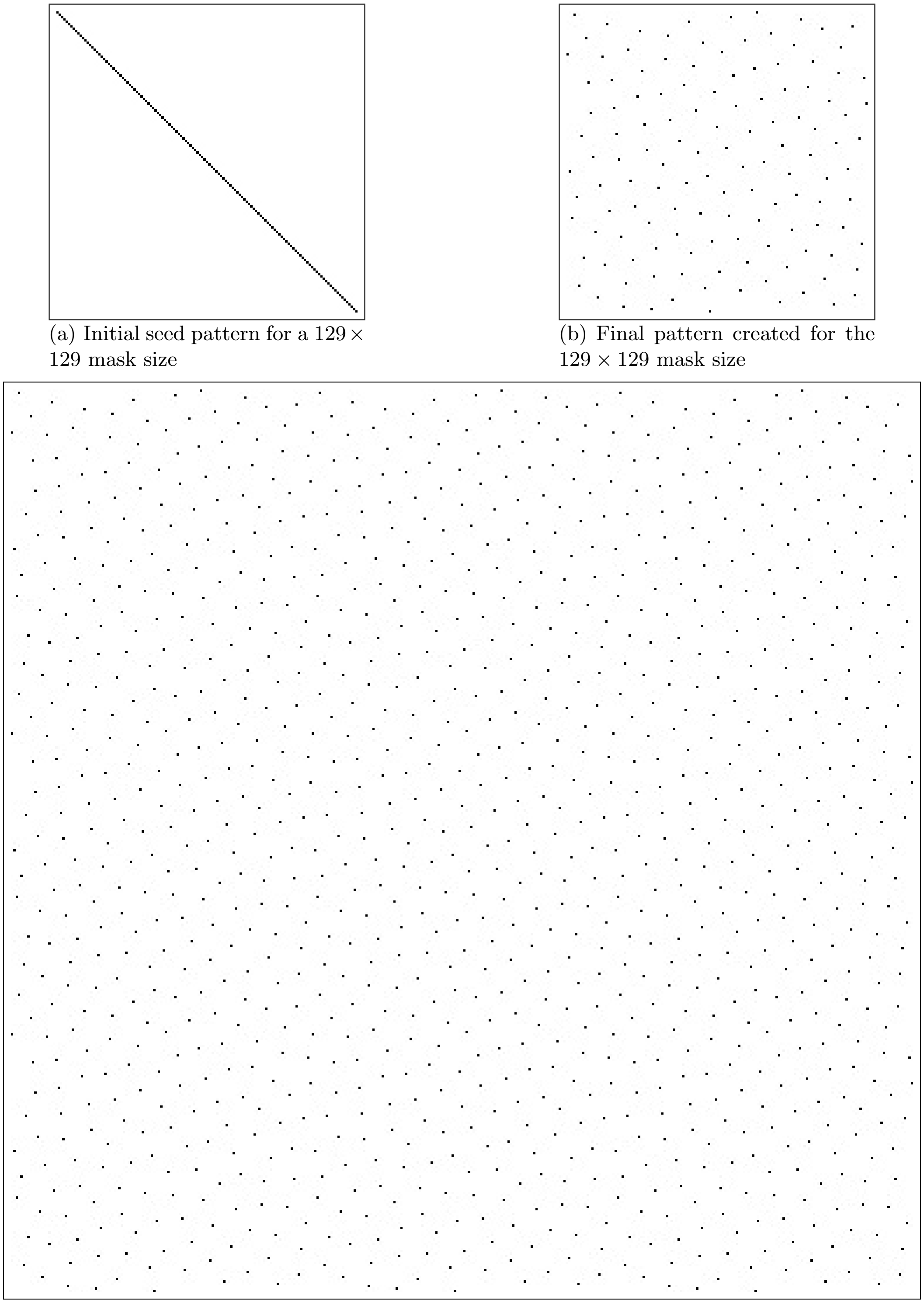
4. RESULTS AND DISCUSSION

Figure 5(a) shows the initial seed pattern g of size 129×129 with $g(m,n) = 1$, where $m = n$ and $m = 1, 2, 3, ...129$. The final output of the proposed algorithm is shown in Fig. 5(b). This visually pleasing pattern has precisely only one dot per row and column. The algorithm also maintains visually pleasing patterns when replicated in either directions due to its symmetric properties. Figure 5(c) is the final output pattern Fig.5(b) replicated 3×3 in both directions.

The algorithm iterated 12 times in this example before it produced the final pattern in Fig.5(b). The constants $k_1, k_2, \sigma_1, \sigma_2$ used to calculate $c_{\tilde{p}\tilde{p}}$ in Eq.(19) are 43.2, 38.7, 0.02, 0.06 respectively. The total pattern design takes only 5 seconds in Matlab.

5. CONCLUSION

We propose, a framework to create constrained visually pleasing masks using a multiple operation Direct Binary Search algorithm. An example mask design, which is appropriate for use as a flushing pattern, is used to demonstrate the algorithm. Alternate design constraints, such as two PELs per column, are possible by modifying the mathematics which extend the concept to create other useful designs. The basic mathematical framework presented can be used to extend the algorithm to the more general case of Direct Multi-bit Search (DMS). Iterative search based halftoning algorithms may be computationally expensive and time consuming, especially for creating a very large pattern. A parallel implementation of the general DBS algorithm was implemented using Graphics Processing Unit(GPU)[15] technology to increase the performance and speed.



(a) Initial seed pattern for a 129 × 129 mask size

(b) Final pattern created for the 129 × 129 mask size

(c) 129 × 129 mask replicated 3 × 3 to produce a 387 × 387 mask size

Figure 5. Results for a mask $g(m, n)$ of 129 × 129 size mask

REFERENCES

[1] Le, H., "Progress and trends in ink-jet printing technology," *The Journal of imaging science and technology* **42**(1), 49–62 (1998).

[2] Chandu, K., Johnson, S., Buckingham, A., and Bildstein, C., "Non-interfering flushing method for inkjet printers," (aug 2008). US Patent App. 12/231,086.

[3] Yamaguchi, S., "Apparatus and method for flushing ink-jet recording heads without suspension of printing," (May 11 1999). US Patent 5,903,288.

[4] Hatayama, F., "Printing system, controller for printing apparatus, method of executing printing process, and program for flushing ejection," (Nov. 10 2009). US Patent 7,614,723.

[5] Inoue, K., "Printing system, controller, print job creation apparatus, method of executing printing process, and program," (oct 2008). US Patent 7,431,420.

[6] Analoui, M. and Allebach, J., "New results on reconstruction of continuous-tone from halftone," *Acoustics, Speech, and Signal Processing, 1992., 1992 IEEE International Conference on* **3**, 313–316 (mar 1992).

[7] Allebach, J. P., "DBS: retrospective and future directions," *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts VI* **4300**(1), 358–376, SPIE (2000).

[8] Ulichney, R., "A review of halftoning techniques," *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts V* **3963**, 378–391 (2000).

[9] Sullivan, J., Ray, L., and Miller, R., "Design of minimum visual modulation halftone patterns," *Systems, Man and Cybernetics, IEEE Transactions on* **21**, 33–38 (jan/feb 1991).

[10] Floyd, R., "An adaptive algorithm for spatial gray-scale," in [*Proc. Soc. Inf. Disp.*], **17**, 75–77 (1976).

[11] Ulichney, R., "Dithering with blue noise," *Proceedings of the IEEE* **76**, 56–79 (jan 1988).

[12] Kolpatzik, B. W. and Bouman, C. A., "Optimized error diffusion for image display," *Journal of Electronic Imaging* **1**(3), 277–292 (1992).

[13] Chandu, K., Stanich, M., Trager, B., and Wu, C. W., "Direct multi-bit search (DMS) screen algorithm," *Image Processing (ICIP), 2012 IEEE International Conference on* , 817–820, IEEE (oct. 2012).

[14] Kim, S. H. and Allebach, J., "Impact of HVS models on model-based halftoning," *Image Processing, IEEE Transactions on* **11**, 258–269 (mar 2002).

[15] Trager, B., Wu, C. W., Stanich, M., and Chandu, K., "GPU-enabled parallel processing for image halftoning applications," *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on* , 1528–1531, IEEE (may 2011).