A Genetic Algorithm Search for Improved Halftone Masks

Peter G. Anderson*[1], *Jonathan S. Arney, Samuel A. Inverso, Daniel R. Kunkle, Timothy M. Lebo, and Chadd Merrigan
Rochester Institute of Technology
Rochester, New York, USA

Abstract

We present a genetic algorithm that automatically generates halftone masks optimized for use in specific printing systems. The search is guided by a single figure of merit based on a detailed model of the printing process and the human visual system. Our experiments show that genetic algorithms are effective in finding improved halftone masks and that two methods of reducing the search space to particular subsets of possible halftone masks greatly enhance the performance of the GA.

1. Introduction

Techniques for rendering excellent black and white approximations of grayscale images, called digital halftoning, are already abundant [7]. However, the reproduction of these halftone images by electrophotographic printers is not absolutely faithful, and the quality of each halftoned image is affected nonlinearly and in a manner specific to the particular printer used. The development of halftoning algorithms and masks has traditionally been a manual process, however, and thus it is too expensive to afford the creation of a specific mask for every printing system.

A number of efforts have already made use of genetic algorithms in designing halftone masks, including [1, 2, 6]. These evolve better masks by the reproduction, recombination, and modification over time of the best masks, as determined by a measure of their fitnesses. We use a fitness measure that can determine the quality of a given halftone mask relative to a specific printing system, given a few specific parameters obtained through a print sample analysis. Our present work focuses on the behavior of electrophotographic (laser) printers. The other significant improvements presented here are two methods of reducing the search space to a smaller subset of masks, all with above average fitness.

Digital halftone masks are represented as $H \times W$ grids containing each of the numbers $1 \ldots H \times W$ exactly once, though in any order. For example, the following spiral-dot mask:

$$\begin{bmatrix} 13 & 12 & 11 & 10 \\ 14 & 3 & 2 & 9 \\ 15 & 4 & 1 & 8 \\ 16 & 5 & 6 & 7 \end{bmatrix}$$

To generate a halftone image using a mask, the following process is executed, at least conceptually.

1. The mask grid is tiled to completely cover the image.
2. Pixels in the continuous image with gray levels less than or equal to the corresponding number in the mask which covers it become 0 in the halftone image; those greater become 1.

In practice, the mask or the image must first be scaled to the range of the other (usually $[0, 1]$ or $[0, 255]$), and step one can be simulated by a modulo operation in step two.

The best halftone images present the least noticeable disturbance in the continuity of a flat gray image to the eye, visible as such artifacts as worms, tears, and checkerboards. Due to the combined effects of neighboring pixels, random toner scattering, transmittance of light through ink, and other nonlinear effects, many halftone masks which appear to deliver very smooth images in their ideal form deliver very poor output in reality. Our fitness model as described in section 3 accounts for these factors to ensure that the masks selected for reproduction will provide the highest quality end result.

Section 4 presents two methods of creating halftone masks from a genotype, the *gear wheel* method and the *hybrid* method. Using either of these methods limits the search space to a subset containing only individuals with relatively high fitnesses and significantly improves the performance of the GA, though the hybrid method clearly provides the greatest enhancement.

2. The Genetic Algorithm

The GA we use is generational, based in part on Goldberg [4].

[1]Send correspondence to `pga@cs.rit.edu`. Internet: `www.cs.rit.edu/~pga`

1. A population of n random possible solutions is created.
2. The fitness of each individual (mask) is evaluated.
3. Repeat the following steps $n/2$ times to create a new generation.
   (a) Choose two parents using tournament selection.
   (b) With probability p_c, crossover the parents to create two children; otherwise clone parents to create children.
   (c) With probability p_m for each child, mutate that child.
   (d) Place the two new children into the next generation.
4. Repeat new generation creation until a solution is found or the process times out.

Some typical parameters for this genetic algorithm are $n \in \{20, 30, ..., 160\}$, $p_c = 0.8$, and $p_m = 0.3$. In this case, a mutation refers to the randomization of one location of the individual's genome.

The individuals in this case are permutations which map to halftone masks by any of our three methods detailed in section 4. A permutation must contain each member of a set in all cases, however, and a simple mutation by randomizing one element in a permutation is guaranteed to affect this complete inclusion. Thus they are represented in the genome by a signature which describes a sequence of reorderings of a simpler sequence of ordered integers. The signature can thus be mutated without affecting the inclusion of each member of the set in the final permutation.

These masks are evaluated according to the fitness function described in section 3. By continually selecting masks with higher fitness and creating new masks through the genetic operators of crossover and mutation the population of masks increases in quality. This procedure is usually given a specific amount of time and is then halted, at which time the best mask found is reported.

3. A Unified Figure of Merit

An earlier iteration of this project employed a multi-objective fitness function to optimize the printer noise level and the visual response separately. As with most multi-objective problems, it is difficult to find an ideal balance between the importance of each objective. Furthermore, variations which will increase the fitness on one objective will often decrease that on the other, as was the case with our previous work.

The former two-part function has been replaced by a single figure of merit which accounts for the significant effects of the printing process on the image. As before, the fitness is based on the noise in a uniform gray image after the appropriate transformations of the printing system and the human visual system are performed.

To calculate the noise level perceived by the eye after printing using an electrophotographic printer, we begin with a continuous grayscale image of some value in the range $[0, 1]$. This image is then halftoned using the mask to be evaluated to produce our basic image M.

The image M is super-sampled by a factor of 5 to produce a new image C with pixels small enough to represent the effects within one addressable printer pixel:

$$C_{i,j} = M_{\lfloor i/5 \rfloor, \lfloor j/5 \rfloor} \tag{1}$$

for $0 \leq i < 5H$ and $0 \leq j < 5W$, where H and W are the height and width of the original image M.

3.1. Dot Gain

Dot gain is a phenomenon by which an idealized image composed of black and white squares is altered by the facts that (1) printers produce round dots, not square, which must be larger in area than the ideal square in order to ensure full coverage of printer toner when the image is completely black, and (2) electric field interactions modify the charge delivered to the printer drum, especially toward the edges of each dot. Our method models this phenomenon by convolving the super-sampled image C with an experimentally derived point spread function PSF_m

$$C = M * PSF_m \tag{2}$$

where

$$\sigma = .034 \times (printer\ dots\ per\ millimeter) \times 5$$

and

$$PSF_m \propto \frac{1}{1 + (\frac{r}{\sigma})^4}$$

where r represents the distance from the center point of the PSF. The actual PSF_m is normalized to sum to 1, as are all of the point spread functions described here.

3.2. Toner Transfer

The transfer of toner to the paper is also a nonlinear phenomenon. Generally, there is a minimum charge on the photoconductive drum which is required to pick up and deposit any toner at all. Our model approximates this relationship by the equations

$$C_f = C^2 \tag{3}$$

and

$$C_n = C^2 + 2Ca \tag{4}$$

where a is a Gaussian random number near 0 with standard deviation 1.

Equation 3 calculates a simple relationship between charge level and actual toner transferred at any one point (i, j), recalling that the gray levels may range from 0 to 1. Equation 4, for C_n, represents the same relationship but adding the noise due to the intrinsic random toner particle scattering that is ubiquitous in the printing process. The two equations are necessary to be able to cache the random noise which results from this printer phenomenon and to add it to our final noise figure.

3.3. Paper and Ink Effects

With two separate images, C_n and C_f, we then calculate the actual reflectance from the printed image. Both calculations are performed as follows, here represented by only the noisy image C_n. First the transmittance of light through the ink is determined:

$$T_k = e^{-\epsilon \cdot n} \tag{5}$$

where $R_k = 0.05$ (reflectance of the ink), $R_g = 0.8$ (reflectance of the paper), and $\epsilon = -log(\sqrt{\frac{R_k}{R_g}})$. The result is convolved with the paper point spread function, PSF_p, and the overall reflectance R is calculated by the standard equation.

$$T_p = T_k * PSF_p \tag{6}$$

$$R = R_g \cdot T \cdot T_p \tag{7}$$

Two standard deviations are taken as a measure of the noise of each image, σ_n and σ_f. The difference of these two represents the noise attributable to the printer's random scattering effects.

$$\sigma_{printer} = max(\sigma_n - \sigma_f, 0) \tag{8}$$

3.4. Visual Effects

The human visual system can only perceive frequencies between alternations of light and dark with a minimum wavelength of $\frac{1}{150}$ of an inch [3]. Therefore, wavelengths of lesser lengths must be blurred out of the image if a true prediction of the eye's response is to be made. In our model, we use a simple Gaussian point spread function PSF_v (VTF in its familiar frequency-space form) to perform this blurring on the image without printer noise.

$$R_v = R_0 * PSF_v \tag{9}$$

Another standard deviation σ_v is extracted from this final image R_v.

3.5. Merit

The printer noise $\sigma_{printer}$ which was calculated in equation 8 is added to σ_v to provide a final noise estimation. Recalling that σ is a square root of the variance, the noises are combined:

$$\sigma = \sqrt{\sigma_v^2 + \sigma_{printer}^2} \tag{10}$$

The final figure of merit produces a number conveniently between 0 and 1, with 1 representing the highest fitness.

$$fig = \frac{\sigma_{max} - \sigma}{\sigma_{max}} \tag{11}$$

where σ_{max} represents the noise level resulting from an unblurred image, $(R_g - R_k) \cdot \sqrt{g(1-g)}$ where g is the testing gray level.

Because this maximum value varies with the gray level of the testing image, the fitness value is partially dependent upon the testing gray level. Early experiments using a different gray level for each generation used the iterative value $g_{next} = (\Phi g) \ mod \ 1$, where Φ is the golden ratio $\frac{\sqrt{5}-1}{2}$. Though this guaranteed an even coverage of all gray values over time, the fitness values varied wildly between generations.

Instead, the algorithm averages the fitnesses calculated from five test images of representative gray values chosen from the harmonic series $\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, ...$ but alternating between those fractions and one minus those fractions, that is $\{.5, .33, .75, .2, .833\}$.

4. Methods of Reducing the Search Space

A number of methods for representing and creating masks were developed. Three methods, full permutation, gear wheel, and hybrid, are presented here. The gear wheel and hybrid methods reduce the search space by limiting the genetic algorithm to masks that are of better than average fitness; the full permutation method is provided as an unoptimized baseline.

4.1. Full Permutation

The full permutation method simply evolves masks over the full search space of permutations. For masks of size $H \times W$ the search space contains $(HW)!$ possible halftone masks. Normal mask sizes for the presented results are 64×64, resulting in an enormous search space of size $4096! \approx 1.3 \times 10^{13019}$.

This naive method is not efficient; it is merely described here as a contrast to the more intelligent methods described below. Figure 1 shows the evolution of a GA population over 12 hours of run time on a 440MHz Sun Ultra-10 workstation, or about 13,000 fitness evaluations.

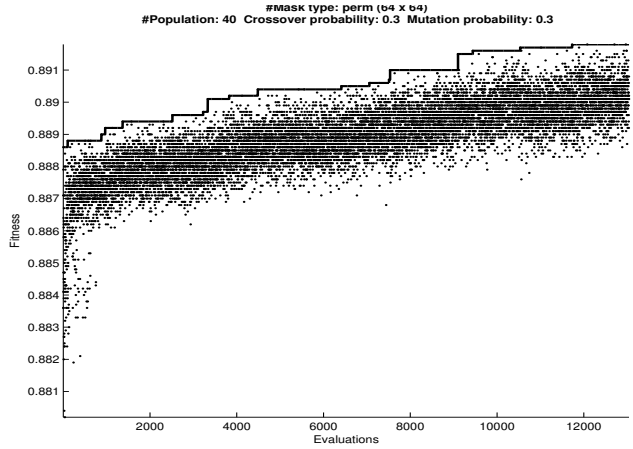


Figure 1: *Full Permutation Method results - Every fitness value plotted against the point at which it occurred during GA evolution. The solid line represents best fitness found as of a given evaluation.*

4.2. Gear Wheel

The gear wheel method evolves orderings of the two sets $\{1, 2, 3, ..., H\}$ and $\{1, 2, 3, ..., W\}$. This reduces the search space to only $H!W!$ possibilities, approximately 2×10^{12857} times smaller than the space searched by the full permutation method for a similarly sized mask.

The two permutations are used to create one permutation of size $H \times W$ that represents a halftone mask. Conceptually, the two permutations are placed on separate gear wheels with tooth-counts H and W, and the wheels are turned one tooth at a time. The two numbers that meet where the two wheels touch describe the row and column in the mask where the next number in the permutation is placed. For this method to cover every coordinate pair of rows and columns, the height and width of the mask must be relatively prime (i.e., there is no number, other than 1, that is a factor of both H and W). By turning the H gear wheel W full rotations, the whole mask is filled. The table below shows the process of creating a 3×4 mask. The teeth on the hypothetical H and W gears are numbered in this instance by the trivial permutations $\langle 1, 2, 3 \rangle$ and $\langle 1, 2, 3, 4 \rangle$. Each numbered tooth in the H gear represents a row in the mask, and each in the W gear represents a column. Note that by the end of this table, each possible combination of H and W has appeared exactly once.

H Gear (row)	W Gear (column)	Mask value
1	1	1
2	2	2
3	3	3
1	4	4
2	1	5
3	2	6
1	3	7
2	4	8
3	1	9
1	2	10
2	3	11
3	4	12

The resulting mask has the following structure:

$$\begin{bmatrix} 1 & 10 & 7 & 4 \\ 5 & 2 & 11 & 8 \\ 9 & 6 & 3 & 12 \end{bmatrix}$$

Although gear wheels give a good conceptual visualization of this method, it is computationally impractical in our implementation language, MATLAB. An equivalent but more efficient algorithm is to initialize an $H \times W$ matrix using the two trivial permutations, as shown above, then permute the rows and columns of this matrix using the two chosen permutations. This allows us to create a trivial permutation mask like the one in the table above at the beginning of a GA run and use the row and column permutations when calculating the fitness of each possible mask.

Figure 2 shows the evolution of the GA population over 12 hours and approximately 8000 fitness evaluations. The best fitness values found using the gear wheel permutation method were 0.9428.

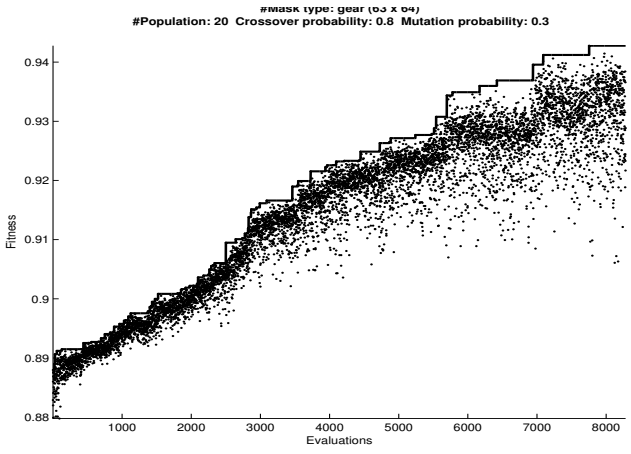


Figure 2: *Gear Wheel Method results - Every fitness value plotted against the point at which it occurred during GA evolution. The solid line represents best fitness found as of a given evaluation.*

4.3. Hybrid

The hybrid method simulates the construction of hybrid masks [5], which tend to be successful, using recursive tesellations. In this construction, two or more masks are combined recursively to produce the final mask which has dimensions equal to the product of the component masks' dimensions.

The combination of component masks M and N of size $h_1 \times w_1$ and $h_2 \times w_2$ respectively will result in a mask of size $h_1 h_2 \times w_1 w_2$ as shown below.

$$R = M \star N \tag{12}$$

$$R = \begin{bmatrix} (hw)M + N_{11} & \cdots & (hw)M + N_{1w} \\ \vdots & \ddots & \vdots \\ (hw)M + N_{h1} & \cdots & (hw)M + N_{hw} \end{bmatrix} \tag{13}$$

where h and w are the height and width of N, and N_{pq} is the value of N at row p and column q.

Visually, the matrix combination can be seen as tiling $h_2 \times w_2$ copies of the multiple $h_2 w_2 M$, where the (p, q) copy is offset by N_{pq}.

This process can be repeated with any number of matrices to produce the final mask whose fitness is evaluated as described earlier, using the visual and printer fitness measures.

For the results listed here, two component matrices, both of size 8×8 were used, producing a mask of size 64×64. The method ran for 12 hours computing approximately 12,000 fitness evaluations. Figure 4 shows the results of a GA using this method. GAs using this method consistently provide better results than either the full permutation or the gear wheel methods.

Using the recursive tiling method described above, any number of component matrices of any size can be used to create a mask. Tests were conducted with all possible combinations of powers of two whose product is 64. There are 31 such combinations, arising from the 10 sets of dimensions and the permutations of those sets. The best performing combination has been demonstrated to be $(8 \times 8) \star (8 \times 8)$.

Figure 5 compares the results of the full permutation, gear wheel, and hybrid methods. The hybrid method produces masks with significantly higher fitness than the other methods. Specifically, recursive methods whose first component matrix is 8×8 perform the best.

5. Results

The images below show selected halftone ramp images for some of the highest fitness masks created by the three methods presented.

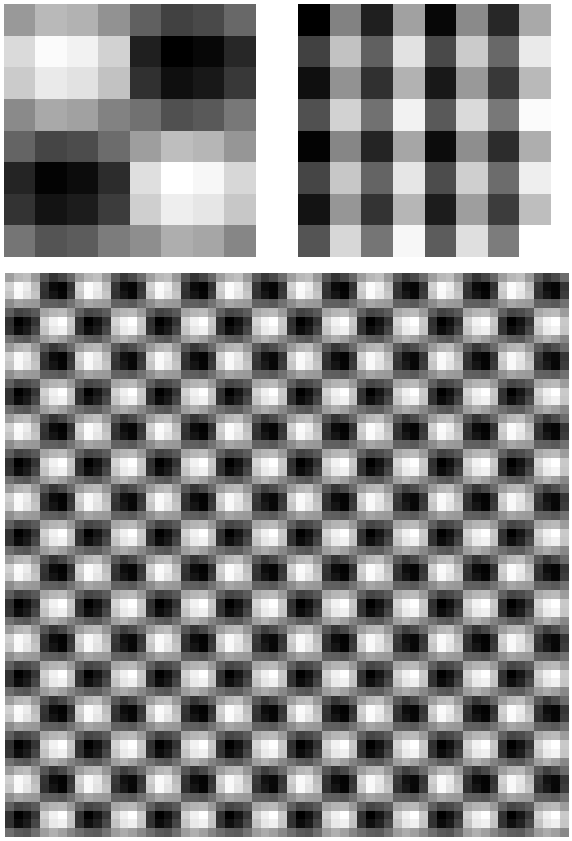


Figure 3: *A hand designed micro-mask (top-left), an evolved macro-mask (top-right), and the resulting hybrid mask (bottom).*

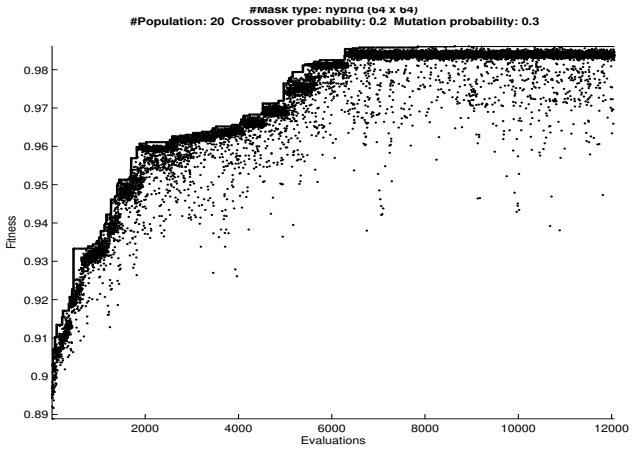


Figure 4: *Hybrid Method results - Every fitness value plotted against the point at which it occurred during GA evolution. The solid line represents best fitness found as of a given evaluation.*

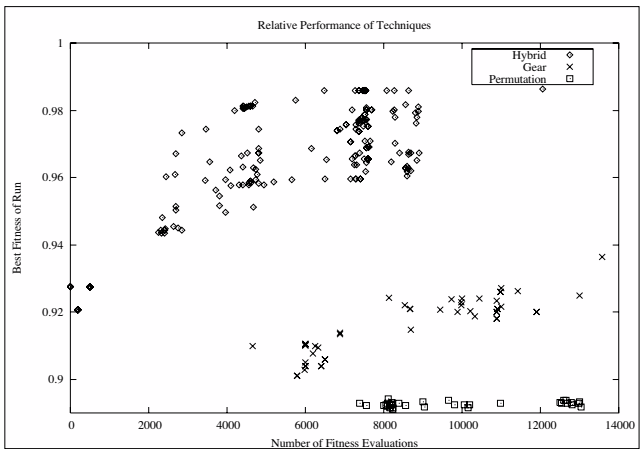
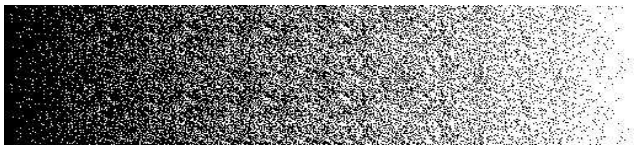
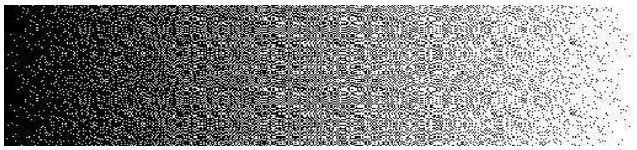


Figure 5: *Comparison of best fitnesses using full permutation, gear wheel, and hybrid methods including power of 2 products. Masks with greater fitness appear in upper right of graph. The recursive method consistently outperforms the other methods.*

Full Permutation



Gear Wheel



Hybrid

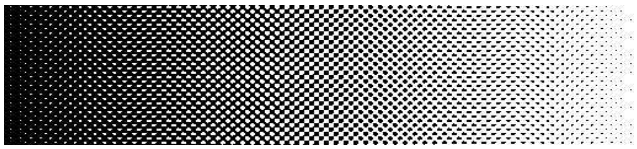


Figure 6: *Horizontal gray ramps halftoned with masks found using the genetic algorithm.*

It should be remembered that at the printed level of magnification, and without the effects of the printing process shown, it is difficult to distinguish a successful halftone mask from a casual glance at the images in figure 6.

6. Conclusion

Genetic algorithms are a valuable tool in designing halftone masks for specific printing systems. The effectiveness of the genetic algorithm is significantly increased by reducing the search space of halftone masks to a subset with above average fitness. This was accomplished using the gear wheel and hybrid methods, of which the hybrid method was the most effective. Ongoing work focuses on the development of even more accurate printer models.

7. Acknowledgments

We gratefully acknowledge the financial support of Hewlett-Packard for this research.

References

[1] Hernán E. Aguirre, Kiyoshi Tanaka, Tatsuo Sugimura, and Shinjiro Oshita. Halftone image generation with improved multiobjective genetic algorithm. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 501–515. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

[2] Jarmo T. Alander, Timo Mantere, and Tero Pyylampi. Threshold matrix generation for digital halftoning by genetic algorithm optimization. In David P. Casasent, editor, *Intelligent Systems and Advanced Manufacturing: Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision*, pages 204–212, 1998.

[3] Peter Engeldrum. *Introduction to Image Microstructure*. Imcotek, Inc., Walworth, NY, 1986.

[4] D. E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[5] H. R. Kang. Frequency analysis of microcluster halftoning. In R. Eschbach, editor, *Recent progress in digital halftoning, II*. IS&T, 1999.

[6] Chih-Ching Lai and Din-Chang Tseng. Printer model and least-square halftoning using genetic algorithms. *Journal of Imaging Science and Technology*, 42(3):241–249, 1998.

[7] Robert Ulichney. *Digital Halftoning*. MIT Press, Cambridge, MA, USA, 1987.