Good Halftone Masks via Genetic Algorithms

Peter G. Anderson, Jonathan S. Arney, Samuel A. Inverso, Daniel R. Kunkle, Timothy M. Lebo, and Chadd Merrigan
Laboratory for Applied Computing, RIT, Rochester, NY

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 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 search performance.[1][2]

Genetic Algorithms

A genetic algorithm (GA) mimics evolutionary processes–i.e., selective breeding–to find good solutions to hard problems indirectly. A GA works with a *population* of proposed solutions to a problem and iteratively removes poor ones by *selection*, combines the pieces of superior ones by *selection and breeding*, and occasionally slightly modifying some by *mutation*.

A GA requires a *fitness* criterion for the solutions. In the present case, the fitness is the figure of merit for a permutation array when used as a halftoning mask. Our figure of merit expresses how well a mask renders a constant gray image—from the point of view of the human visual system and the ability of a given printer to render small dots.

[1]Presented at the 2003 Western New York Image Processing Workshop, Rochester, NY, October 17, 2003.

[2]We gratefully acknowledge Hewlett Packard's financial support for this research.

Our GA begins by randomly initializing a population of individual solutions and determining their fitnesses. Then it iteratively selects pairs of relatively superior individuals to crossover to produce pairs of new individuals. It removes pairs of relatively poor individuals to keep the population size constant. It mutates then evaluates the new individuals.

This process terminates after a predetermined time or when a satisfactory individual appears.

GAs combine *exploration* of the search space, using a large population and mutation, with *exploitation* of a promising region of the search space, using selection and crossover. (Extreme examples of exploration and exploitation are random search and hill-climbing, respectively. Neither of these techniques is well suited to hard problems with enormous search spaces.)

Halftoning Masks

A halftoning mask, M, is an array of threshold values used to convert a continuous tone image, I, to a bi-level image B.

The values of I and M are numbers in the same range, typically 0–255. The values in B are 0 or 1.

A mask halftoning process converts I to B using M according to the following rule:

$$I_{pq} < M_{pq} \Rightarrow B_{pq} = 0 \qquad I_{pq} \geq M_{pq} \Rightarrow B_{pq} = 1$$

When the image B is printed on a high-resolution laser or ink jet printer, it should appear "the same as" the continuous tone image I. A good halftoning mask will not introduce any visible artifacts (e.g., unwanted textures) and it will use dot patterns that the printer can reproduce faithfully.

GA Fitness for a Mask

Our implementation of a mask's fitness is how smoothly the human visual system perceives constant gray image after it has been thresholded by the mask, and rendered on paper by a particular printer. Thus, we take into account concepts such as dot gain and printer instability in evaluating masks.

The mask translates a constant gray image, with gray level $g \in [0, 1]$, to a binary (black and white) image with fraction g of black pixels. This image convolves with a linear spread function representing the laser transfer to the photoconducting drum. Toner attaches to the drum and is then transfered to paper in a noisy and nonlinear manner.

The parameters for the above transformations are experimentally derived for individual printers and individual toners in color printers.

Permutation Masks

A genetic algorithm will necessarily perform better if some knowledge of the problem is built into the search procedure—other than just the black box of the fitness evaluation.

For this application, we limit our search to $H \times W$ halftone masks containing all the values 0–$HW - 1$, i.e., permutation matrices. For most of the work $H = W = 64$. We modify the comparison details of the halftoning threshold by scaling—i.e., comparing $256 \cdot M_{pq}$ and $H \cdot W \cdot I_{pq}$.

We know that certain families of permutations perform better than arbitrary ones, so we further restrict our searches, as discussed below.

Unrestricted, Full Permutations

This is the largest ($(H \cdot W)! = 64^2! \approx 1.3 \cdot 10^{13019}$ possible elements) and probably the worst choice for any type of search.

Randomly chosen masks in this space behave like Roberts's Method, producing mottled images.

However, the GA searching this space does make significant—but slow—progress. See the top chart of fitness improvement (Figure 1) and the top of the bi-level ramp images in Figure 3.

Gear Wheels

For this type of mask we require that H and W have no common factors (H and W are "relatively prime"), so we choose $H = 63, W = 64$.

Imagine two meshing gear wheels of H and W teeth. Label the teeth on each with permutations of $\{0, 1, 2, \cdots, H-1\}$ and $\{0, 1, 2, \cdots, W-1\}$. The relatively-prime criterion guarantees that every pair of teeth from the two wheels will meet exactly once in W revolution of the H wheel (and H revolutions of the W wheel). Starting at time $t = 0$ and rotating the gears one tooth at every moment, we have the tooth pair (p_t, q_t) meeting at time t. The permutation matrix is defined by $M_{p_t,q_t} = t$. We see there are $H! \cdot W!$ possible permutation matrices achieved this way. For our chosen parameters, $H! \cdot W! \approx 2.5 \cdot 10^{176}$.

The above description of how to create a gear-wheel permutation matrix is not particularly suitable to our chosen programming language, MatLab. An equivalent algorithm works along the following lines. Start with a matrix M constructed from two trivial permutations, $(0, 1, 2, \cdots H-1)$ and $(0, 1, 2, \cdots W - 1)$. Then, rearrange the rows of M according to an H-permutation and the columns of M using an W-permutation.

The gear-wheel masks are generally superior to arbitrary permutation masks, and the populations evolve faster—see Figures 1–3.

Hybrid Masks

This type of mask formulation appears to give the best performing masks. It combines two 8×8 permutation masks, K and L, to form a 64×64 mask—thus the size of the search space is $64!^2 \approx 1.64 \cdot 10^{178}$. The combining process works like this. Lay out 8×8 copies of $64 \cdot K$ and add L_{pq} to the (p, q)-copy of $64 \cdot K$.

GA Performance

Figure 1 shows the relative growth of fitnesses towards 1.00 for the three search spaces. Figure 2 shows the combined results of a variety of experiments. Finally, Figure 3 shows three ramps rendered using the best halftone masks for our three methods.

References

More extensive explanations of this material are in these conference proceedings which are available from the authors:

- Jonathan S. Arney and Peter G. Anderson, "Optimizing Halftone Masks with Genetic Algorithms and a Printer Model," *Proceedings of the International Conference on Digital Printing Technologies (NIP 19)*, New Orleans, Fall, 2003, The Society for Imaging and Technology

- Peter G. Anderson, Jonathan S. Arney, Samuel A. Inverso, Daniel R. Kunkle, Timothy M. Lebo, and Chadd Merrigan, "A Genetic Algorithm Search for Improved Halftone Masks," *Proceeding of ANNIE 2003: Artificial Neural Networks in Engineering*, St. Louis, MO, November, 2003.

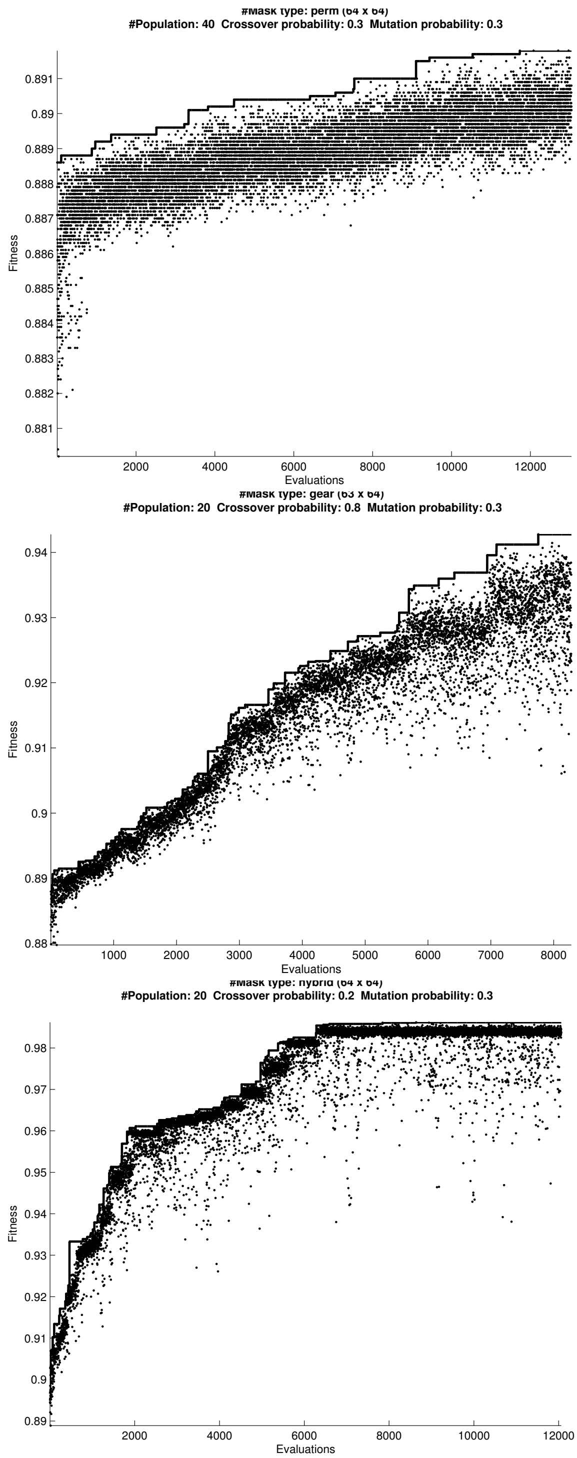

Figure 1: The improvements over time of the GA searching the three spaces of permutation masks. Think of the x-axis as time. *Top: full permutation, middle: gear wheel, bottom: hybrid. Every mask's fitness value is indicated. The solid lines enveloping the top of the scatter patterns represent the best fitness values found as of that time.*

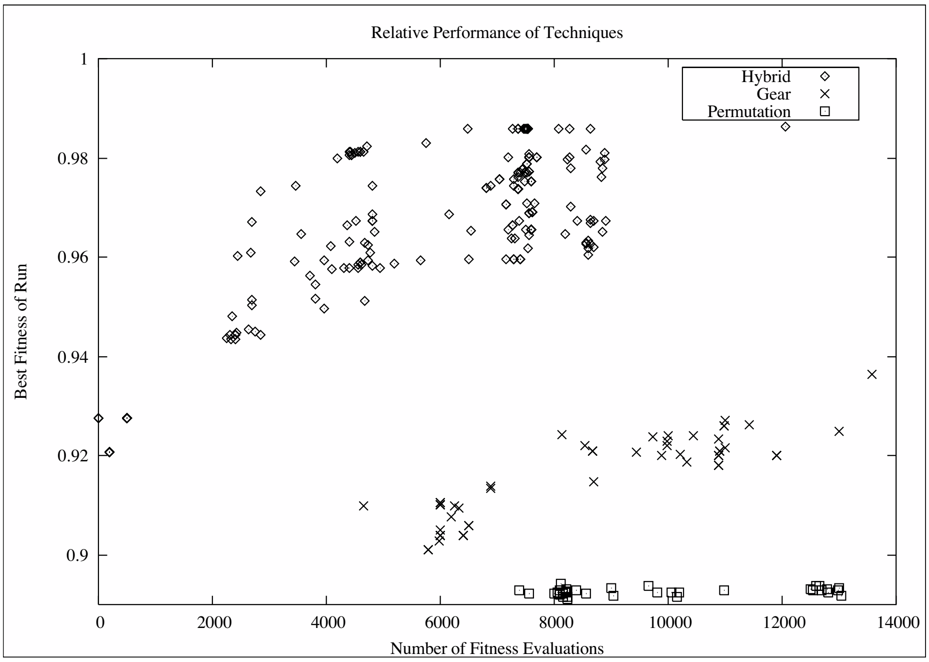

Figure 2: *Comparison of best fitnesses using full permutation, gear wheel, and hybrid methods including power of 2 products. The hybrid method consistently outperforms the other methods.*

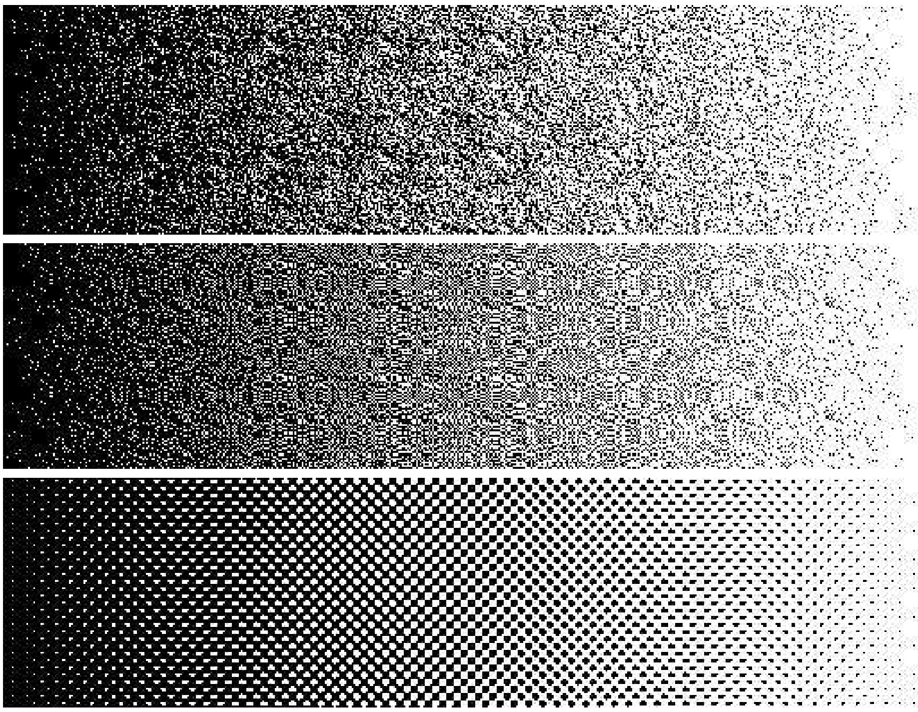

Figure 3: *Horizontal gray ramps halftoned with masks found using the genetic algorithm. Top: full permutation, middle: gear wheel, bottom: hybrid. Note: the masks that produced these images were evolved for a printer and a resolution other than that of the present document. This illustration only shows mask structure, not the actual images they produce.*