Printer Model and Least-Squares Halftoning Using Genetic Algorithms

*Chih-Ching Lai and Din-Chang Tseng**
Institute of Computer Science and Information Engineering, National Central University, Chung-li, Taiwan 320

Abstract

In this paper, a least-square model-based halftoning technique using a genetic algorithm is proposed to produce a halftone image by minimizing the perceived reflectance difference between the halftone image and its original image. We use a least-square criterion incorporating with the property of the human visual system to measure the difference between the two images. The genetic algorithm is used for investigating the complicated search problem. The standard halftoning techniques, such as error diffusion and least-square halftoning, produce gray-level distortion because of dot-gain problem. In this study, we use a modified dot-overlap printer model to compensate the gray-level distortion. The printer model combines with a measurement-based algorithm to estimate the print-dot radius and makes the proposed halftoning approach adapted to a wide variety of printers and papers. Experiments show that the proposed approach produces more accurate gray levels than several common-used halftoning methods produce.

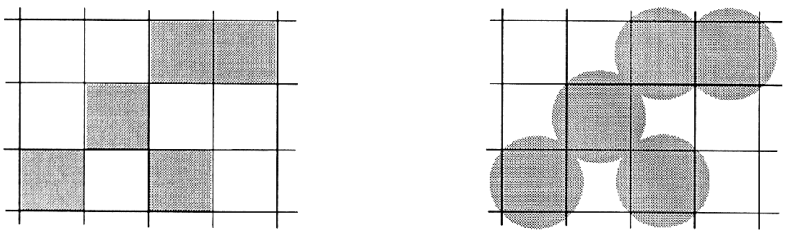
Introduction

Digital halftoning refers to any algorithmic process that creates the illusion of continuous-tone images from the judicious arrangement of binary picture elements.[1] Many halftoning techniques[1–4] have been proposed to improve printing quality such as, artifact, texture, or false-contour removing; deblurring; and contrast enhancing. However, only a few techniques have been proposed for accurate gray-level rendition.[5–8] Accurate gray-level rendition means to minimize the perceived gray-level difference between a printed halftone image and its original image. In halftone printing, we always assume that the printed dot is square; however, most printers produce circular dots and the dot size is always larger than the square size because of ink spread, wax compression, and ink absorption into papers. The increase in dot size was termed as the physical dot gain as shown in Fig. 1. This phenomenon causes the printed gray levels to be darker than the expectation. In this report, we compensate the physical dot-gain problem to produce halftone images with less gray-level distortion.

Three kinds of halftoning techniques have been proposed to compensate the dot-gain problems: (1) measurement-based calibration, (2) least-square halftoning, and (3) model-based halftoning.

The measurement-based calibration technique[9] was proposed to compensate for any gray-level distortion. In the method, an image consisting of 256 gray levels is halftoned and then printed. A tone–response curve[9] is generated for measuring the reflectance of the printed image. The curve is used to describe the relationship between the perceived gray level (input reflectance) and the output reflectance. The calibration technique corrects the output response of a particular halftoning algorithm on a specific device using an inverse mapping function. This process works well but has three drawbacks: (1) Each individual halftoning algorithm needs to be calibrated for each individual printer. (2) The calibration of dithered results may produce less than 256 gray levels. (3) Only specific pixel patterns are measured. If the microstructure of the halftone image greatly varies from the measured pattern, the gray levels predicted by the calibration curve may not be representative of what is printed on paper and the tones will not be reproduced accurately.[9]

On the basis of the least-square-error criterion, a halftoning technique can be regarded as a process of arranging black dots to simulate gray levels on a bilevel output device. Hence, a halftoning technique is taken as



(a) (b)

Figure 1. The physical dot gain: (a) ideal dots, (b) round dots cause dot gain.

an optimization algorithm that searches for a suitable placement for black dots to produce halftone images with less gray-level distortion. However, in our sense, a primary least-square halftoning is poor to compensate for the gray-level distortion; more treatments are needed to get better results.

Zakhor *et al.*[10] presented a class of dithering techniques for black and white images. They divided an image into small blocks and minimized the gray-level difference between every corresponding block pair in the original continuous-tone image and its low-pass filtered halftone image. They utilized quadratic programming with linear constraints to solve the standard optimization problem. However, they did not consider the printer characteristics needed to compensate for gray-level distortion; moreover, their method needs much computation.

Saito and Kobayashi[11] tried to produce a less-distortion halftone image by using evolutionary computation approaches, but they did not consider a printer model to compensate for the dot-gain problem; moreover, stochastic errors are associated with the simplest selection method in their evolutionary algorithm.

The model-based halftoning technique relies on an accurate printer model to predicate and compensate for gray-level distortion to produce less-distortion halftone images.[5] Several model-based halftoning approaches have been proposed. Anastassiou[12] proposed a "frequency weighted square error criterion" to minimize the square error between the eye-filtered binary image and the eye-filtered gray-level image. However, he assumed that printers generate non-overlapped perfect dots.

Pappas and Neuhoff[6] exploited both a printer model and a visual-perception model in a least-square model-based (LSMB) halftoning algorithm. The algorithm attempts to produce the "optimal" halftone reproduction by minimizing the square error between "the response of the cascade of the printer and a visual model to the binary image" and "the response of the visual model to the original gray-level image."[6] However, they used an exhaustive search to find the optimal binary image by updating the binary value of one pixel at a time and thus spend a huge processing time.

In this paper, we propose a genetic algorithm combined with a modified dot-overlap printer model for least-square halftoning. Genetic algorithms (GAs)[13–15] are probabilistic search methods guided by the principles of evolution and natural genetics. GAs are well known for their ability to explore large search spaces efficiently and adaptively.[15] Originally, GAs were modeled and developed by Holland,[16] and now have emerged as general purpose and robust optimization techniques. The process of arranging the placement of black dots in a halftone image to compensate for gray-level distortion is tedious. Genetic operators can dynamically arrange black dots on bilevel output devices; thus we use a GA as a search algorithm for halftoning problems. We consider the problem-specific knowledge in the GA and incorporate a printer model into the approach to improve the gray-level rendition.

Pappas *et al.*[5,7–8] have proposed two printer models: "circular dot-overlap" and "measurement of printer parameters" to compensate for gray-level distortion. We evaluated these two printer models and chose the circular dot-overlap printer model for further study because (1) the circular dot-overlap printer model is simpler and more flexible than the measurement model in use; (2) the measurement model needs to solve the constrained optimization problem, and an initial estimation of the solution that satisfies the constraints must be provided; (3) in the measurement model, several local minima are often presented in the solution space; it is not always possible to determine the global minimum; and (4) the parameters of the measurement model (e.g., 3×3 window or larger) are too many to solve the whole constrained optimization problem.

However, the circular dot-overlap printer model generates bias error in the printed images;[7] thus we propose a modified circular dot-overlap printer model to compensate for gray-level distortion and resolve the bias problem. Using the circular dot-overlap printer model, the dot radius must be known in advance. We here also use a preproposed measurement-based method to estimate the print-dot radius and make the modified dot-overlap printer model adaptable to a wide variety of printers and papers.

The remaining sections of this paper are organized as follows: First we present the proposed approach, then the experiments and discussion, and the conclusions are summarized in the last section.

Proposed Halftoning Approach

The most intuitive formulation of the halftoning problem can be stated as "to find a binary image such that the gray-level difference between an original continuous-tone image and its perceived bilevel image is minimized."[12] Because the human visual system acts as a low-pass filter, the least-square halftoning approach minimizes the mean square error between the low-pass version of a gray-level image and that of its halftone image.

The block diagram of the proposed least-square model-based GA halftoning technique is shown in Fig. 2. Initially, a least-square criterion is defined to measure the difference between the low-pass version of a gray-level image and that of its halftone image. A modified circular dot-overlap printer model is taken into the least-square criterion to compensate the gray-level distortion in the halftone image. Then a genetic algorithm is utilized to find the "optimal" halftone reproduction based on the least-square criterion. Here, the "optimal" halftone reproduction means that the output gray-level response is linear; that is, the tone-response curve is a straight line of slope one through the origin.[9] In processing, an image is divided into small blocks and the blocks are processed in the raster-scan order to propagate (block) gray-level errors to the right and the lower neighboring blocks.

Least-Squares Criterion. Assume that an $M \times N$ gray-level image is represented by $[g_{i,j}]$, $i = 1, ..., M$, and $j = 1, ..., N$. We use a 2-D Gaussian filter with impulse response $[h_{i,j}]$ to simulate the eye model to evaluate the gray-level difference between a gray-level image and its halftone image. The original image $[g_{i,j}]$ is partitioned into sev-

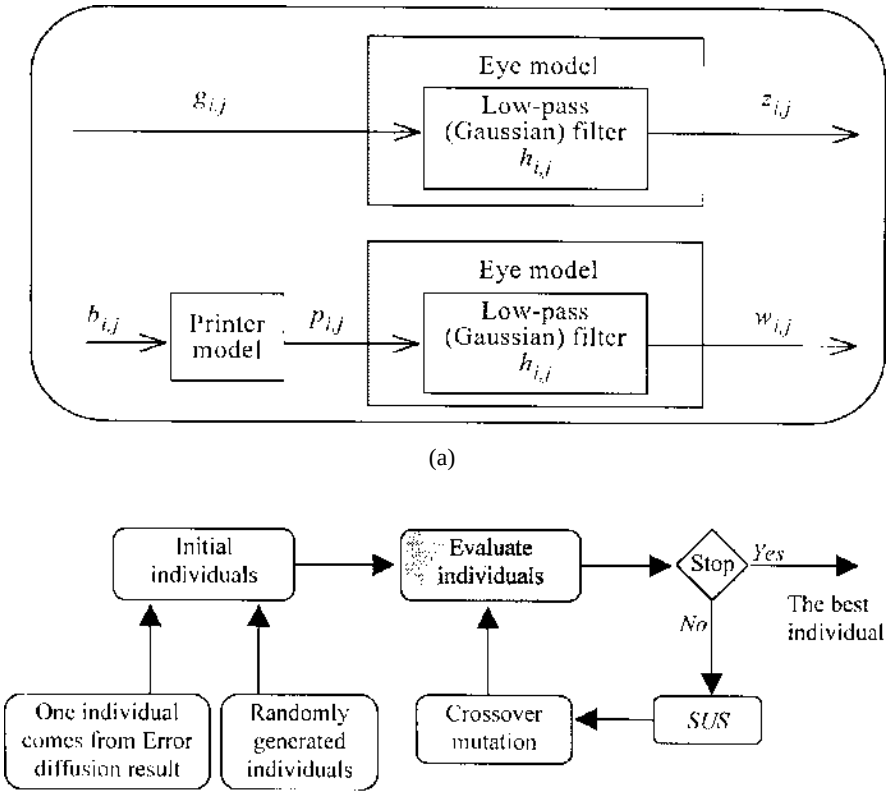


Figure 2. The least-square model-based GA halftoning: (a) the evaluation process; (b) the genetic algorithm.

eral $n \times n$ blocks, where both M/n and N/n are integers. Then the evaluation function for a block is defined as

$$E = \sum_{i=1}^{n} \sum_{j=1}^{n} (z_{i,j} - w_{i,j})^2, \qquad (1)$$

where

$$z_{i,j} = g_{i,j} * h_{i,j},$$

$$w_{i,j} = p_{i,j} * h_{i,j},$$

$$p_{i,j} = P(W_{i,j}),$$

and * indicates a convolution operator. The $p_{i,j}$ is a printer parameter of the modified dot-overlap printer model utilized instead of the output gray level $b_{i,j}$ for compensating the gray-level distortion and described in the Printer Model Section. Next, we use a genetic algorithm as an evolutionary computation to generate the halftone image $[b_{i,j}]$ by minimizing the square error E.

Genetic Algorithms. A genetic algorithm (GA) is a stochastic algorithm used to solve search and optimization problems. The algorithm is based on the mechanics of natural selection and genetics in biological systems.

In general, a GA contains a fixed-size population of potential solutions over the search space. These solutions are encoded as bit strings and called individuals or chromosomes. The initial population can be created randomly or based on problem-specific knowledge. At each iteration, called a *generation*, a new population is created. To generate a new population based on a preceding one, the algorithm performs the following three steps: (1) evaluation: each individual of the old population is evaluated by a fitness function and given a value to denote its merit, (2) selection: individuals with better fitness are selected to generate the next population, and (3) mating: genetic operators such as crossover and mutation are applied to the selected individuals to produce new individuals for the next generation. The above steps are iterated for many generations until a satisfactory solution is found or a stopping criterion is met. A standard *GA* is described as the following pseudocodes:

```
t ← 0
initialize P(t)
evaluate P(t)
while (the stop criterion is not met) do
    begin
            t ¨ t + 1
            select P(t) from P(t – 1)
            recombine P(t)
            evaluate P(t)
    end,
```

where $P(t)$ is the population at generation t.

Typically, using a GA to solve a problem, we must provide the following components: (1) a genetic representation of solutions to the problem, (2) one way to create an initial population of solutions, (3) an evaluation

function that rates each candidate solution according to its "fitness," (4) genetic operators that effect genetic information of children during reproduction, and (5) control parameters (e.g., population size, crossover, mutation rates, *etc*.)[15]

Solution Representation. In our utilization of a GA to find the "optimal" halftone image, a binary block is encoded as a bit string. We create a population of strings and evaluate each string, then select the best strings to construct the new population. Finally, the binary block $[b_{i,j}]$ with the highest fitness value is found.

Initial Population. In general, a *GA* creates its starting population by filling with randomly generated bit strings; however, we can heuristically rather than randomly generate the individuals in the initial population to improve the search performance. Heuristic initialization may be helpful but must be done carefully to avoid premature convergence. Here we use the result of standard error diffusion as an individual and the remaining individuals are randomly generated. Note that if the initial population contains a few individuals far superior to the rest of the population, the GA may quickly converge to a local optimum.

Fitness Function. Fitness function is the survival arbiter for individuals. In the halftoning problem, the objective is to find the binary halftone image block $[b_{i,j}]$ that minimizes the mean square error E. Because the fitness in GAs is to find the maximum profit, the fitness function in the halftoning problem is defined by

$$\begin{aligned} F &= C_{\max} - E \\ &= C_{\max} - \sum_{i=1}^{n} \sum_{j=1}^{n} (z_{i,j} - w_{i,j})^2, \end{aligned} \tag{2}$$

where $C_{\max}$ is a predefined value or the maximum of E.

Genetic Operators. Three primary genetic operators: selection, crossover, and mutation are generally involved in a GA.

Selection. The selection operator determines the surviving individuals. Each surviving individual is reproduced into several copies according to its relative responsibility. Two reproductive strategies were commonly used. Generational reproduction replaces the whole population in each generation, but *steady-state* reproduction[13] only replaces the least-fitted members in a generation.

Baker[17] compared various selection methods comprehensively and presented an improved version called the stochastic universal sampling (SUS) method. A SUS[17] procedure is described by the following *C* codes:

```
ptr = Rand( );
for(sum = i = 0; i < N; i++)
    for(sum += ExpVal[i]; sum > ptr; ptr++)
        Select_individual(i);
```

where Rand( ) returns a random real number between 0 and 1. We use steady-state reproduction and the SUS method in the proposed halftoning approach.

Crossover. The crossover operator randomly pairs individuals with a probability p_c and swaps parts of their genetic information to produce new individuals. Several types of crossover operators such as, one-point, two-point, and uniform-type splitting have been proposed. However, DeJong and Spears[18] concluded that the uniform crossover is more beneficial if the population size is small and, hence, gives a more robust performance. The uniform crossover is adopted in the proposed halftoning approach.

In the uniform crossover,[13] two parents are selected and two children are produced. One of the selected parents has the best fitness and the other is randomly selected. Each bit position in the children is created by copying the corresponding bit from either one of the parents. The uniform crossover randomly generates a crossover mask that is a bit string with the same length of an individual for each pair of parents. A "1" bit in the crossover mask means that the first child inherits the gene from the first parent and other genes from the second parent. The second child uses the opposite rule. One example of the uniform crossover is shown in Fig. 3.

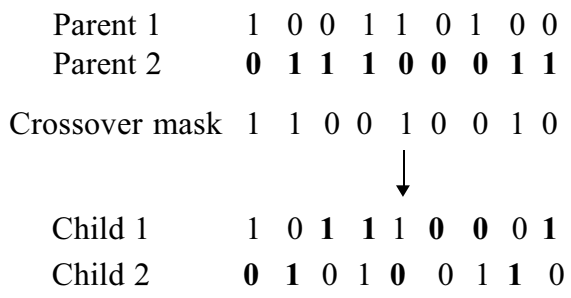


Figure 3. An example of the uniform crossover.

Mutation. The mutation operator creates a new individual by altering one or more genes of an individual with a probability p_m to increase the variability of the population. One or more bits in the crossover children are inverted; "1" is changed to "0" and "0" is changed to "1." In the proposed halftoning approach, the mutation operator works as the example shown in Fig. 4. The example shows three individuals of length 5 and a random number generated for each bit in each individual. The bit changes its value when the random number test is passed. The random number that causes a bit to change is printed in bold face.

Control Parameters. The population size influences the performance of GAs. A small-size population reduces the evaluation cost but results in premature convergence, because the population provides insufficient samples in the search space. For a large-size population, the GA can gain more information to search better solutions because the population contains more representative solutions over the search space.

Parent individual					Random numbers					New bit	Child individual				
1	0	0	1	0	.103	.305	.248	.851	.552	-	1	0	0	1	0
0	1	0	1	1	.203	.123	**.005**	.657	.341	1	0	1	**1**	1	1
0	0	1	1	0	.469	.778	.651	**.033**	.951	0	0	0	1	**0**	0

Figure 4. Examples of bit mutation used in the proposed approach.

Both crossover and mutation probabilities may influence the performance of GAs. A GA may stagnate in the search for new solutions if the crossover probability is low. However, if the crossover probability is too high, unstable solutions may be quickly substituted into the population for individuals with better fitness. But if the mutation probability is too high, the search of the GA becomes a random-like process. In the proposed halftoning approach, extra experiments are performed to find more appropriate values for the used parameters.

Printer Models. Most standard halftoning techniques produce darker halftone images than the expectation due to the dot-gain problem. To resolve this problem and achieve a less-distortion gray-level rendition, the printer characteristics should be considered. Model-based techniques[5–8] exploited the printer characteristics to compensate for gray-level distortion.

The Pappas–Neuhoff circular dot-overlap model[5,8] assumed the print dot to be circular with a uniform distribution of ink and the dot radius at least $T/\sqrt{2}$, where T is the spacing of the Cartesian grid, so that a black region can be blackened entirely. They used r to denote the ratio of the actual dot radius to the ideal dot radius $T/\sqrt{2}$. The amount of dot-gain area at each pixel is expressed in terms of parameters α, β, and γ as shown in Fig. 5(a). These parameters are the ratios of the areas of the shaded regions shown in Fig. 5(a) to T^2. The parameters a, b, and g are expressed in terms of ρ as follows:

$$\alpha = \frac{1}{4}\sqrt{2\rho^2 - 1} + \frac{\rho^2}{2}\sin^{-1}\left(\frac{1}{\sqrt{2\rho}}\right) - \frac{1}{2}, \tag{3}$$

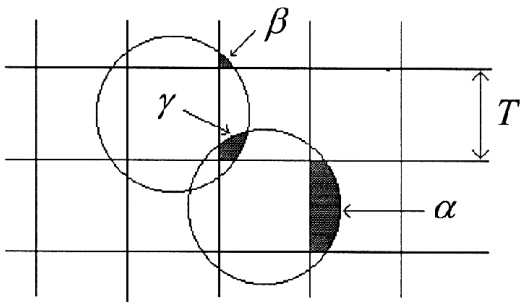
$$\beta = \frac{\pi\rho^2}{8} - \frac{\rho^2}{2}\sin^{-1}\left(\frac{1}{\sqrt{2\rho}}\right) - \frac{1}{4}\sqrt{2\rho^2 - 1} + \frac{1}{4}, \tag{4}$$

$$\gamma = \frac{\rho^2}{2}\sin^{-1}\left(\sqrt{\frac{\rho^2 - 1}{\rho^2}}\right) - \frac{1}{2}\sqrt{\rho^2 - 1} - \beta. \tag{5}$$

The constraint of r is $1 \le \rho \sqrt{2}$. In terms of these parameters, the circular dot-overlap model is defined as

$$p_{i,j} = P(W_{i,j}) = \begin{cases} 1, & \text{if} \quad b_{i,j} = 1 \\ f_1\alpha + f_2\beta - f_3\gamma, & \text{if} \quad b_{i,j} = 0 \end{cases}, \tag{6}$$

where $W_{i,j}$ denotes a window consisting of $b_{i,j}$ and its eight neighbors, f_1 is the number of horizontal and vertical neighboring black dots, f_2 is the number of diagonal neighboring black dots not adjacent to any horizontal or vertical neighboring black dot, and f_3 is the number of pairs of neighboring black dots in which one is a horizontal neighbor and the other is a vertical neighbor. One example for describing the circular dot-overlap model is given in Fig. 6(a).



(a) Definition of α, β, and γ.

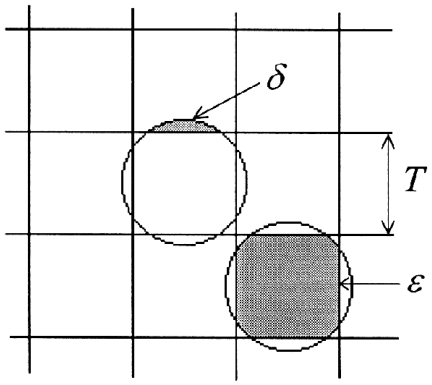
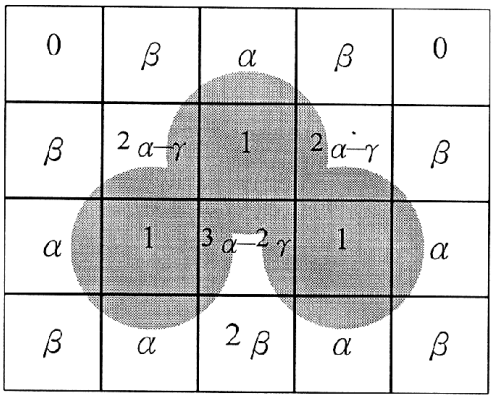


Figure 5. Definition of parameters used in the circular dot-overlap printer model; (a) definition of α, β, and γ; (b) definition of δ and ε.

Pappas and Neuhoff[8] indicated ρ = 1.25 in the circular dot-overlap printer model for *HP* laser printers. However, the estimated print-dot sizes are different for different printers or on different papers. Here we release the constraint of $1 \le \rho \le \sqrt{2}$. We use our measurement-based method to estimate statistically the dot radius for the printer model to adapt a wide variety of printers and papers and then we extend the circular dot-overlap print model to include all possible cases of print-dot radii.

Measurement of Print-Dot Radii. In general, the more print dots are considered, the more accurate dot area is estimated. In order to find a reasonable dot radius, we consider a larger area of dots. At first, we define a number of 3×3 print patterns with different "0 & 1" permutations to analyze the physical dot gain. The 3×3 print patterns can define 2^9 different "0 & 1" permutations. We reduce the number of patterns by taking the reflected or rotated patterns to be the same and then get

102 different "0 & 1" patterns including two special patterns: all "0" (white) and all "1" (black) patterns.



printer parameters $p_{i,j}$'s when $r_f \geq T$

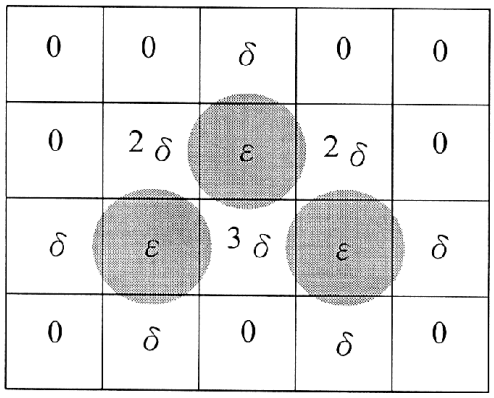


Figure 6. Two examples of the printer parameters $p_{i,j}$ for different dot radii; (a) the printer parameters $p_{i,j}$ when $r_f \geq T/\sqrt{2}$, (b) ther printer parameters $p_{i,j}$ when $T/2 \leq r_f < T/\sqrt{2}$.

Secondly, we print out all print patterns repeatedly in vertical and horizontal directions as one example shown in Fig. 7 and measure the densities of these patterns using a reflection densitometer. Then the Murray-Davies equation[19]

$$A = \frac{1-10^{-D_t}}{1-10^{-D_s}} \tag{7}$$

is utilized to describe the relationship among the effective dot area, A; the density of a solid dot, D_s; and the resultant density of the dot area pattern, D_t. Assume that the measured density of pattern i is D_i; the measured densities of the all-black-dot and all-white-dot patterns are D_b and D_w, respectively. Let the measured black area of pattern i be S_i, then S_i can be calculated by

$$S_i = \frac{1-10^{-(D_i-D_w)}}{1-10^{-(D_b-D_w)}} S_b, \quad i = 1,2,...,100, \tag{8}$$

where S_b is the 3×3 grid area.

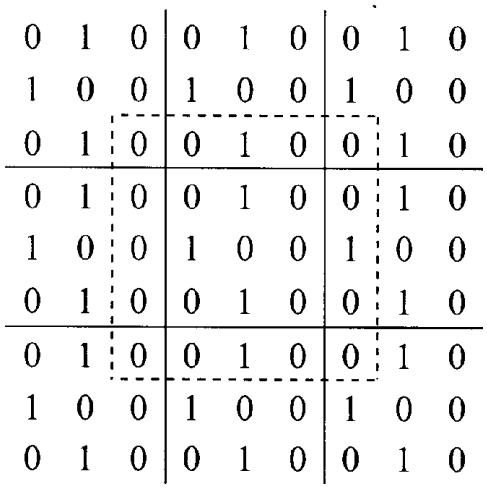


Figure 7. A sample of periodic 3×3 print patterns. The 5×5 dashed-line window is one example for defining the coefficients a_i in calculating the black area of the internal 3×3 print pattern.

On the basis of the dot-overlap printer model, the black area of pattern i can be calculated by

$$A_i = \begin{cases} (a_1 + a_2\alpha + a_3\beta - a_4\gamma)T^2, & \text{if} \quad T/\sqrt{2} \leq r \\ (a_1\varepsilon + a_2\delta)T^2, & \text{if} \quad T/2 \leq r < T\sqrt{2} \\ a_1\rho r^2, & \text{if} \quad r < T/2 \end{cases} \tag{9}$$

where r is the estimated radius of print dots; a, b, and g are the ratios of the areas of the shaded regions shown in Fig. 5(a) to T^2; and d and e are the ratios of the areas of the shaded regions shown in Fig. 5(b) to T^2. The parameters d and e are expressed in terms of r as follows:

$$\delta = \frac{\rho^2}{2}\cos^{-1}\left(\frac{1}{\sqrt{2}\rho}\right) - \frac{1}{4}\sqrt{2\rho^2 - 1}, \tag{10}$$

$$\varepsilon = \frac{\pi\rho^2}{2} - 4\delta. \tag{11}$$

The definition of coefficients a_2, a_3, and a_4 is similar to that of f_i in the circular dot-overlap printer model Eq. 6; however, the former is more complicated. The coefficients a_2, a_3, and a_4 are defined by considering a 5×5 window centered at a 3×3 print pattern in periodic print patterns as one example shown in Fig. 7 with the dashed-line block. For every white dot in the 3×3 pattern, we compute its dot-gain area in terms of a, b, g, d, and e and then accumulate the dot-gain area for all white dots in the 3×3 pattern. Coefficient a_1 is the number of black dots in the 3×3 print pattern. Coefficient a_2 is the accumulated number found by counting the horizontal and vertical neighboring black dots for every white dot in the 3×3 print pattern noting that all neighboring black dots in the 5×5 window are considered. Coefficient a_3 is the accumulated number found by counting the diagonal neighboring black dots for every white dot in the

3×3 print pattern while no black dot is simultaneously adjacent to both the white dot and the counting black dot. Coefficient a_4 is the accumulated number found by counting the pairs of adjacent neighboring black dots in which one is a horizontal neighbor and the other is a vertical neighbor for every white dot in the 3×3 print pattern.

We take the measured black area and the calculated black area in a print pattern to be equal; that is, using the equation

$$S_i = A_i, \; i = 1, 2, \ldots, 100, \tag{12}$$

to estimate the radii for all print patterns. Due to the measured error and non-uniform roughness of printed papers, the dot radii are different and distributed. We need to determine the best-fitted dot radius for the printer model. The best-fitted dot radius means that the dot radius r_f minimizes the sum of the square errors between the measured densities and the calculated densities of all print patterns. At first, we substitute each estimated radius r_j into every print-pattern equation (i.e., Eq. 9) to calculate the black area $A_i^{r_j}$ and then substitute the area and Eq. 8 into Eq. 12 to get the calculated density $D_i^{r_j}$,

$$D_i^{r_j} = D_w - \log\left[1 - \frac{A_i^{r_j}}{S_b}\left(1 - 10^{-(D_b - D_w)}\right)\right], \quad i = 1,2,\ldots,100. \tag{13}$$

The sum of square errors for the estimated radius r_j is given as

$$SE(r_j) = \sum_{i=1}^{100} \left(D_i - D_i^{r_j}\right)^2, \quad j = 1,2,\ldots,100, \tag{14}$$

where D_i is the measured density of print pattern i. The best-fitted radius r_f is the radius with the minimum SE error.

Modified Dot-Overlap Printer Model. To include all possible cases of print-dot radius, we extend the circular dot-overlap printer model[8] as follows:

$$p_{i,j} = \begin{cases} P_1(W_{i,j}), & \text{if} \quad b_{i,j} = 0 \\ P_2(W_{i,j}), & \text{if} \quad b_{i,j} = 1' \end{cases} \tag{15}$$

$$P_{i,j}(W_{i,j}) = \begin{cases} f_1\alpha + f_2\beta - f_3\gamma, & \text{if} \quad T/\sqrt{2} \le r_f \\ f_1\delta, & \text{if} \quad T/2 \le r_f < T/\sqrt{2}, \\ 0, & \text{if} \quad r_f < T/2 \end{cases} \tag{16}$$

$$P_2(W_{i,j}) = \begin{cases} 1, & \text{if} \quad T/\sqrt{2} \le r_f \\ \varepsilon, & \text{if} \quad T/2 \le r_f < T/\sqrt{2}, \\ \pi\rho^{2}\!/_{2}, & \text{if} \quad r_f < T/2 \end{cases} \tag{17}$$

where $W_{i,j}$ denotes a window consisting of $b_{i,j}$ and its eight neighbors; P_1 and P_2 are two functions for calculating the estimated gray level $p_{i,j}$ for dot $b_{i,j}$; the parameters a, b, g, d, and e are defined in Eqs. 3 through 5 and Eqs 10 and 11; and the definition of coefficients f_1, f_2, and f_3 is the same as the definition of the Pappas–Neuhoff printer model[8] given in Eq. 6. One example of $p_{i,j}$ for the case of $T/2 \le r_f < T/\sqrt{2}$ is shown in Fig. 6(b).

The differences between the Pappas–Neuhoff dot-overlap printer model and the modified dot-overlap printer model are (1) the dot radius in the Pappas–Neuhoff model is obtained from an experience or an assumption value, but the dot radius in the modified model is measured to adapt to a wide variety of printers and papers and (2) the Pappas–Neuhoff model assumes the dot radius is always larger than the cover square area, but the modified model considers all possible cases of dot radius.

During the halftoning process, all image blocks are processed in the raster-scan order. The mutation operator incorporated with the printer model embedded in the fitness function is used to test whether a binary block is good enough. We adaptively adjust the combination of 0's and 1's through both the mutation operator and the printer model to reduce gray-level distortion. After we find a near-optimal binary block, the right-most (binary) pixel values are recorded for the processing of the right block and the lower most pixel values are also recorded for the processing of the lower block to propagate (block) gray-level errors to obtain near-unbiased-error halftone images.

Experiments

We demonstrate here the experimental results of the proposed approach and compare the printed results and errors with those generated by other halftoning techniques. A HP LaserJet 5MP printer was used to print halftone images, and a Macbeth RD-1200 reflection densitometer was used to measure the densities of print-dot patterns. All halftone images were printed on the same uncoated papers commonly used for copy machines.

Note that the parameter values in GAs, such as maximum generation number and probability of genetic operators, always influence the performance of the algorithms.[20] However, GAs are always robust with respect to these parameters; thus only a few experiments are needed to specify the parameter values. Of course, reasonable parameters ensure good results and give rise to quick convergence. The parameters used in the experiments are (1) each individual represents a possible permutation of 0's and 1's in a 5×5 image block; (2) the generation number is 150; (3) the population size is 30; (4) the adopted selection and crossover operations are stochastic universal sampling and uniform crossover; and (5) the probabilities of crossover and mutation are 0.7 and 0.1, respectively.

Nine halftoning algorithms: Floyd–Steinberg error diffusion, Jarvis–Judice–Ninke error diffusion, Stucki error diffusion, 4×4 dither-matrix ordered dither, 8×8 dither-matrix ordered dither, dot diffusion, least-square *GA* halftoning without printer model, least-square GA halftoning with Pappas–Neuhoff circular dot-overlap printer model, and the proposed least-square GA halftoning with the modified circular dot-overlap printer

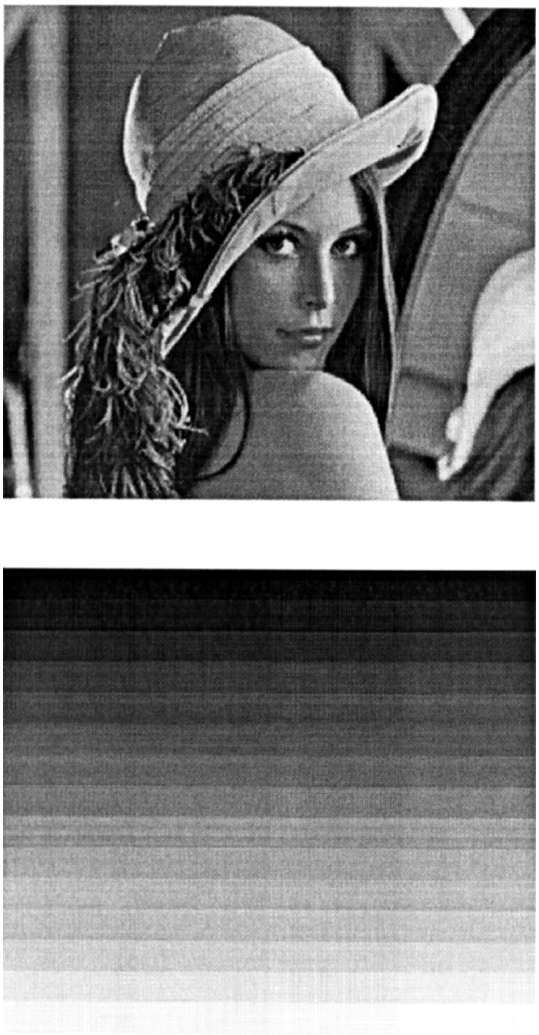


Figure 8. Two test images were printed using the HP LaserJet 5MP at 600 dot/inch (dpi) resolution

model, were compared. These halftoning algorithms were examined by using two images: "Lena" and the "gray-level chart" as shown in Fig. 8. Both images have 765 ¥ 720 pixel resolution with 256 gray levels. Eighteen halftone images generated by the nine halftoning techniques were printed at 300 dpi, but only four representative halftone images are shown in Fig. 9.

The image obtained via the proposed least-square GA halftoning with the modified circular dot-overlap printer model has fewer worm-like artifacts than the error diffusion methods and does not have the false contour as the ordered dither method produces. Comparing Figs. 9(a) and 9(d), we find the proposed approach produces blocking-effect halftone images. The phenomenon resulted from the fact that the proposed approach is a block-based halftoning method. The thresholded errors in error diffusion are "diffused" pixel by pixel, but the proposed approach is based on the idea of breaking up a gray-level image into small blocks and solving the optimum for each block with a genetic algorithm. Zakhor et al.[10] also produced blocking-effect halftone images. The blocking effect can be avoided when the image is not partitioned into blocks for processing; however, such processing expends a huge amount of time as spent by the Pappas and Neuhoff[6] process. The uniformity of a halftone image is influenced by the bandwidth of a 2-D Gaussian filter and the block frequency $1/n$, where block size is $n \times n$. When the bandwidth of a 2-D Gaussian filter is wider and the block frequency $1/n$ is smaller, the halftone image is very blurred.

The purpose of this study was to print halftone images with less gray-level distortion. We compared the nine halftoning algorithms not only browsing the printed images, but also inspecting the tone-response curves. We printed out the "gray-level chart" using the nine halftoning algorithms and used a Macbeth RD-1200 reflection densitometer to acquire the density and then transform to the (output) reflectance. Four representative tone-response curves are given in Fig. 10. In these curve charts, the abscissa denotes the perceived gray level from 0 (black) to 1 (white) and the ordinate denotes the measured reflectance of a printed halftone image. The perceived gray level is proportional to the amount of ink on the printed paper.[7] Thus, the tone-response curve of a tone-correct halftone image is a straight line of slope one through the origin. The curves of all standard halftoning algorithms are more concave than that of the proposed algorithm. This means that these standard halftoning techniques produce more gray-level distortion than the proposed technique. There is stepped output reflectance in the curves of the ordered dithers because they only produce 17 and 65 gray levels, respectively.

Based on the tone-response curves, two sum-square-error criteria were used to evaluate all halftoning techniques as shown in Table I. The absolute square error (ASE) measures the difference between the tone-response curve and the straight line of slope one through the origin of the chart. The value of ASE describes the deviation from the ideal response. The relative square error (RSE) measures the difference between the tone-response curve and its least-square fitting straight line. The value of RSE indicates the degree of the linearity of the output reflectance. The least-squares GA halftoning with the modified circular dot-overlap printer model always has the least error. We conclude that the proposed halftoning approach produces less distorted halftone images than other commonly-used halftoning techniques.

Conclusions

On the basis of a least-squares criterion, a genetic algorithm combined with the modified circular dot-overlap printer model was proposed to produce halftone images with less gray-level distortion. The proposed approach minimized the gray-level difference between the low-pass version of a continuous-tone image and that of its halftone image. We used genetic operators, crossover, and mutation in the *GA* to arrange the placement of black dots in a halftone image and to find the "optimal" halftone reproduction. We proposed the modified circular dot-overlap printer model to describe the printer characteristics and then compensate for gray-level distortion by means of spatial adjustment of black-dot locations. We also quoted our measurement-based method to estimate statistically the radii of print dots for the printer

(a) (b)

(c) (d)

Figure 9. The printed images using four representative halftoning techniques at 300-dpi resoution: (a) Jarvis–Judice–Ninke error diffusion; (b) least-square GA halftoning without printer model; (c) least-square GA halftoning with Pappas-Neuhoff circular dot-overlap printer model; (d) least-square GA halftoning with the modified circular overlap printer model.

model. The experimental results revealed the proposed approach reduces the gray-level distortion and produces more accurate gray levels than a number of halftoning techniques. From the experimental results, several aspects for future work may be enumerated:

1. Incorporating problem-specific knowledge into genetic algorithms has been acknowledged as an effective problem-solving tool in many research fields. Combining more halftoning characteristics with genetic algorithms to obtain better printed results is a potential research topic.
2. The human visual system plays an important role in digital halftoning. In the proposed approach, we use a simple 2-D Gaussian filter as the eye model. More complicated or more suitable eye models should be incorporated into the proposed approach to improve the printed results further.
3. Recently color printers have been broadly used in the office and home. Color printers also generate color distortion. To apply the proposed approach to color printing by considering the relationship among all color attributes is worth studying.

Acknowledgment

The authors would like to thank the anonymous reviewers for indicating the wrong citation of equations and providing suggestions to improve this article's quality and presentation.

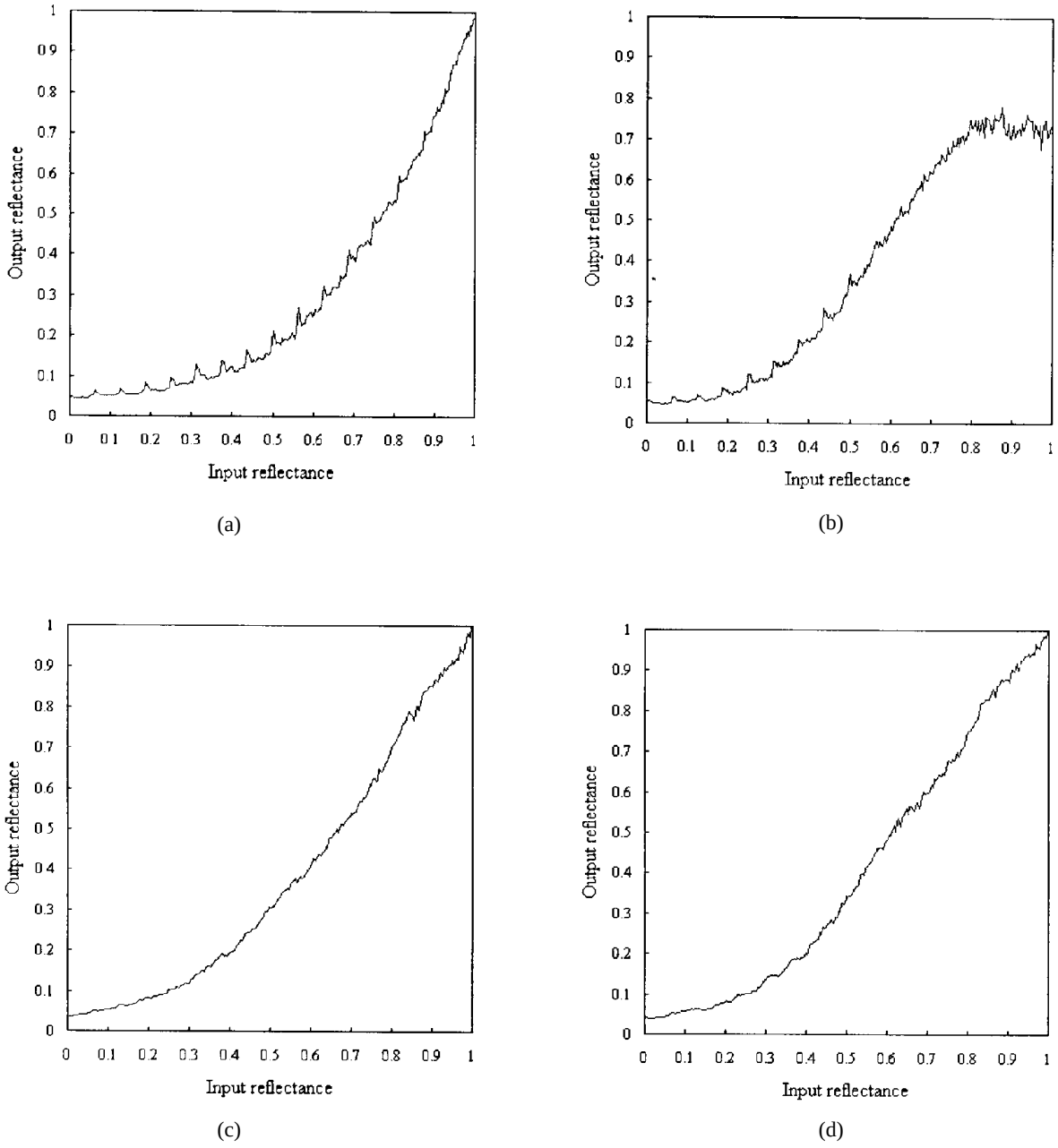


Figure 9. The tone-response curves of four representative halftoning techniques: (a) Jarvis–Judice–Ninke error diffusion; (b) least-squares GA halftoning without printer model; (c) least-squares GA halftoning with Pappas–Neufhoff circular dot-overlap printer model; (d) least-squares GA halftoning with the modified circular dot-overlap printer model.

References

1. R. Ulichney, *Digital Halftoning*., MIT Press, MA, 1987.
2. R. Eschbach, Reduction of artifacts in error diffusion by means of input-dependent weights, *J. Electron. Imaging* **2,** 352 (1993).
3. D. E. Knuth, Digital halftones by dot diffusion, *ACM Trans. Graphics* **6,** 245 (1987).
4. J. C. Stoffel and J. F. Moreland, A survey of electronic techniques for pictorial image reproduction, *IEEE Trans. Commun.* **29,** 1898 (1981).
5. T. N. Pappas and D. L. Neuhoff, Model-based halftoning, *Proc. SPIE* **1453,** 244 (1991).
6. T. N. Pappas and D. L. Neuhoff, Least-squares model-based halftoning, *Proc. SPIE* **1666,** 165 (1992).
7. T. N. Pappas, C.-K. Dong, and D. L. Neuhoff, Measurement of printer parameters for model-based halftoning, *J. Electron. Imaging* **2,** 193 (1993).
8. T. N. Pappas and D. L. Neuhoff, Printer models and error diffusion, *IEEE Trans. Image Proces.* **4,** 66 (1995).
9. C. J. Rosenberg, Measurement-based evaluation of a printer dot model for halftone algorithm tone correction, *J. Electron. Imaging* **2,** 205 (1993).
10. A. Zakhor, S. Lin, and F. Eskafi, A new class of b/w halftoning algorithms, *IEEE Trans. Image Process.* **2,** 499 (1993).
11. H. Saito and N. Kobayashi, Evolutionary computation approaches to halftoning algorithm, *Proc. First IEEE Conf.*

Evolutionary Computation, IEEE, NJ, 1994, p. 787.
12. D. Anastassiou, Error diffusion coding for A/D conversion, *IEEE Trans. on Circuit Sys.* **36,** 1175 (1989).
13. D. Beasley, D. R. Bull, and R. R. Martin, An overview of genetic algorithms: Part 2, research topics, *Univ. Comput.* **15,** 170 (1993).
14. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addision-Wesley, Reading, MA, 1989.
15. M. Michalewicz, *Genetic Algorithms + Data Structure - Evolution Program*, Springer-Verlag, Berlin, 1992.
16. J. H. Holland, *Adaptation in Natural and Artificial Systems*, The Univ. of Michigan Press, Ann Arbor, MI, 1975.
17. J. E. Baker, Reducing bias and inefficiency in the selection algorithm, *Proc. Second Int. Conf. Genetic Algorithm*, Hillsdale, NJ, 1987, p. 14.
18. K. A. DeJong and W. M. Spears, An analysis of the inter acting roles of population size and crossover in genetic algorithms, in *Parallel Problem Solving from Nature*, H.-P. Schwefel and R. Manner, Eds., Springer-Verlag, Berlin, 1990, p. 38.
19. J. R. Huntsman, A new model of dot gain and its application to a multi-layer color proof, *J. Imaging Technol.* **13,** 136 (1987).
20. J. J. Grefenstette, Optimization of control parameters for genetic algorithms, *IEEE Trans. Sys. Man. Cyber.* **16,** 122 (1986).

❋ Previously published in the *Journal of Imaging Science and Technology*, **42**(3), pp. 241–249, 1998.