

PRINTER MODELS AND THE DIRECT BINARY SEARCH ALGORITHM

Farhan A. Baqai and Jan P. Allebach

Electronic Imaging Systems Laboratory
School of Electrical and Computer Engineering
Purdue University, W. Lafayette, IN 47907-1285
{farhan+allebach}@ecn.purdue.edu

ABSTRACT

We incorporate a higher order measurement-based model for printer dot interactions within the iterative direct binary search (DBS) halftoning algorithm. We also present an efficient strategy for evaluating the change in computational cost as the search progresses. Experimental results are shown which demonstrate the efficacy of the approach.

1. INTRODUCTION

With the advent of low-cost desktop printers, there is an increasing demand for high quality printing. Real life images are continuous tone. Most printers are binary or multilevel with a few levels. Digital halftoning provides a mechanism for rendering continuous tone images on such devices.

In halftoning, it is assumed that the perceived gray level is proportional to the fraction of dots in the pattern. Since printer dots overlap, the perceived gray level is darker than expected. This effect is commonly known as “dot overlap”. To deal with the problem, Stevenson and Arce proposed a hard circular dot (HCD) model [1]. It assumes that the printer produces circular spots with absorptance 1 within the dot boundary and 0 elsewhere. Also, the effect of dot overlap is modeled as a logical OR. Pappas and Neuhoff parameterized this model, characterizing dot overlap by three variables [2]. They applied the HCD model to several halftoning techniques including the iterative least-squares algorithm [2, 3]. The least-squares approach is of particular interest because it yields images of very high quality. We have developed an iterative least-squares halftoning algorithm which we call DBS [4]. In our work, we have emphasized efficient computation of the halftone image [4, 5]. Preliminary results of DBS with the HCD model are given in [4], while a detailed study is contained in [5].

The printing mechanism determines the physical properties of the printed dot. Due to the statistical nature of the processes involved, the printer dot is neither round nor square. For example, consider the output from an electrophotographic printer when the input is four replications of the bit map shown in Fig. 1(a). A comparison of high resolution scans of the printed output for printer resolutions of 300 dpi and 600 dpi is given in Figs. 1(b) and 1(c). Note that the halftone in Fig. 1(c) has been scaled up by a factor of two. It can be seen from Fig. 1(b) that the printed dots at 300 dpi can be modeled as hard circular spots. However, it is difficult to fit the same model at printer resolutions of 600

dpi, as is clear from Fig. 1(c). Also, the interaction between overlapping regions of adjacent dots is not a logical OR. Therefore, at higher resolutions, the HCD model does not accurately predict the microstructure of printed dots. Motivated by the above observations, we focus on a higher order printer model which has also been used by Pappas et al [3]. To avoid alignment of the measuring device, they employed a densitometer to measure the average reflectance and solved an optimization problem to get the printer parameters. We use a Howtek 4000 dpi drum scanner to directly measure the absorptance for various dot combinations.

In this paper, we incorporate the measurement based printer model into the DBS algorithm. DBS coupled with the measurement model exploits the intermediate gray levels caused by dot overlap and properties of both output device and viewer to generate images of excellent quality. To make the algorithm computationally tractable, an efficient strategy for evaluating the change in computational cost as the search progresses is also presented.

The remainder of the paper is organized as follows. In Sec. 2, we describe the measurement-based empirical printer model. Efficient implementation of the DBS algorithm is treated in Sec. 3. The experimental results are presented in Sec. 4. Finally, our conclusions are given in Sec. 5.

2. MEASUREMENT-BASED PRINTER MODEL

The function of a printer model is to accurately predict the gray level of the printed pixel. Since, printer dots overlap, the gray level at any pixel is dependent on itself and its neighbors. We assume that it can be determined by pixels in a 3×3 window. We account for dot overlap by replacing the digital halftone $g[m, n]$ by its equivalent grayscale image $\hat{g}[m, n]$, which is defined to have the same average absorptance on a pixel-by-pixel basis as the actual printed halftone image. For instance, Fig. 2(b) contains the measured equivalent grayscale image of the halftone in Fig. 2(a). So $\hat{g}[m, n]$ is given by

$$\hat{g}[m, n] = \mathcal{F}(W[m, n]) \quad (1)$$

where $W[m, n]$ consists of $g[m, n]$ and its eight neighbors, and \mathcal{F} is some function of W . Once all the measurements are made, we have a model describing the dependence of \mathcal{F} on the eight neighbors of $[m, n]$. These values are stored in a look-up-table (LUT), which is used by the halftoning algorithm.

To reduce the number of dot configurations from $2^9 = 512$, we assume that configurations obtained by shifting or flipping other configurations are redundant. Using these assumptions, the

number of dot configurations required to complete the printer model comes down to 158. Note that unlike [3] we have not assumed rotational invariance. To get a good estimate, we average over 100 different replications of the dot configuration. Another important issue is the precise alignment of the pattern. Due to printer position errors, the location of the dot configuration can vary significantly. To get the precise position, we print four fiducial marks around each configuration. The centroids of the fiducial marks are obtained by using least-squares. Based on the centroid locations, precise alignment is achieved.

3. EFFICIENT IMPLEMENTATION OF THE DBS ALGORITHM

The point spread function of human visual system (HVS) $\tilde{p}[m, n]$ is derived from a linear model based on the spatial frequency dependence of the contrast sensitivity function of the human viewer. Perceived versions of the halftone and the continuous-tone image are obtained by convolving them with $\tilde{p}[m, n]$. DBS is a recursive search heuristic which uses a HVS model to minimize the difference $\tilde{e}[m, n]$ between the perceived halftone $\tilde{g}[m, n]$ and the perceived continuous-tone image $\tilde{f}[m, n]$. We use the total squared error as the measure of the difference

$$\varepsilon = \sum_m \sum_n |\tilde{e}[m, n]|^2 \quad (2)$$

The initial estimate of the halftone is obtained by randomly thresholding the continuous-tone image. The halftone image is iteratively scanned, and at each pixel location, we evaluate the effect of toggling its state or swapping its value with any of its 8 nearest neighbors. The toggle or swap that results in the greatest decrease in the squared error is accepted. An iteration is complete when every pixel in the image has been visited. When no changes are accepted during an iteration, the algorithm has converged to a local minimum of the error metric.

Consider evaluating the effect on error for a trial toggle at any location (m_0, n_0) in the image. Any change in (m_0, n_0) would generally also alter its 8 neighbors. Therefore, the new perceived halftone can be written as

$$\tilde{g}'[m, n] = \tilde{g}[m, n] + \sum_{i=1}^9 a_i \tilde{p}[m - m_0 + \Delta m_i, n - n_0 + \Delta n_i] \quad (3)$$

where a_1, a_2, \dots, a_9 represent the change in grayscale at the corresponding pixels in the 3×3 region when we toggle (m_0, n_0) . The new perceived error is $\tilde{e}'[m, n] = \tilde{f}[m, n] - \tilde{g}'[m, n]$.

Let $P \times P$ be the region of support for $\tilde{p}[m, n]$. Direct computation of a trial change in error due to a toggle using (2) requires $9(P+1)^2$ multiplications and $9(P+1)^2$ additions to get $\tilde{g}'[m, n]$, P^2 additions to compute the new $\tilde{e}'[m, n]$ within the support of $\tilde{p}[m, n]$, followed by $2P^2$ multiplications and $(2P^2 - 1)$ additions to get the change in ε . If the trial change is kept, we update the stored array $\tilde{e}[m, n]$ with the new block of values which were just computed. This requires P^2 memory transfers. The number of trial evaluations are expected to be much greater than the number of accepted changes. So it is desirable to minimize the computational cost of evaluating trial changes. Let $c_{\tilde{p}\tilde{p}}[m, n]$ be the autocorrelation function for $\tilde{p}[m, n]$, and $c_{\tilde{p}\tilde{e}}[m, n]$ be the

cross correlation between $\tilde{p}[m, n]$ and $\tilde{e}[m, n]$. Using (3) in (2), the change in the error metric can be expressed as

$$\begin{aligned} \Delta\varepsilon = & c_{\tilde{p}\tilde{p}}[0, 0] \sum_{i=1}^9 a_i^2 - \\ & 2 \sum_{i=-1}^1 \sum_{j=-1}^1 a_{3(i+1)+j+2} c_{\tilde{p}\tilde{e}}[m_0 + i, n_0 + j] + \\ & 2(a_1 a_9 c_{\tilde{p}\tilde{p}}[-2, -2] + (a_1 a_8 + a_2 a_9) c_{\tilde{p}\tilde{p}}[-2, -1] + \\ & (a_1 a_7 + a_2 a_8 + a_3 a_9) c_{\tilde{p}\tilde{p}}[-2, 0] + (a_2 a_7 + \\ & a_3 a_8) c_{\tilde{p}\tilde{p}}[-2, 1] + a_3 a_7 c_{\tilde{p}\tilde{p}}[-2, 2] + (a_1 a_6 + \\ & a_4 a_9) c_{\tilde{p}\tilde{p}}[-1, -2] + (a_1 a_5 + a_2 a_6 + a_4 a_8 + \\ & a_5 a_9) c_{\tilde{p}\tilde{p}}[-1, -1] + (a_1 a_4 + a_2 a_5 + a_3 a_6 + a_4 a_7 + \\ & a_5 a_8 + a_6 a_9) c_{\tilde{p}\tilde{p}}[-1, 0] + (a_2 a_4 + a_3 a_5 + a_5 a_7 + \\ & a_6 a_8) c_{\tilde{p}\tilde{p}}[-1, 1] + (a_3 a_4 + a_6 a_7) c_{\tilde{p}\tilde{p}}[-1, 2] + (a_1 a_3 \\ & + a_4 a_6 + a_7 a_9) c_{\tilde{p}\tilde{p}}[0, -2] + (a_1 a_2 + a_2 a_3 + \\ & a_4 a_5 + a_5 a_6 + a_7 a_8 + a_8 a_9) c_{\tilde{p}\tilde{p}}[0, -1]) \end{aligned} \quad (4)$$

Since $c_{\tilde{p}\tilde{p}}[m, n]$ is independent of the original image and the halftone, it can be precomputed and stored in a LUT. When a trial change is accepted, the LUT containing $c_{\tilde{p}\tilde{e}}[m, n]$ is updated to $c'_{\tilde{p}\tilde{e}}[m, n]$ as

$$c'_{\tilde{p}\tilde{e}}[m, n] = c_{\tilde{p}\tilde{e}}[m, n] - \sum_{i=1}^9 a_i c_{\tilde{p}\tilde{p}}[m - m_0 + \Delta m_i, n - n_0 + \Delta n_i] \quad (5)$$

It can be seen from (4) and (5) that updating the error only involves a few table lookups and some scalar multiplications and additions. For instance, if $P = 17$, direct evaluation of change in error requires 7276 operations, while with efficient implementation, it can be done in only 122 operations, where an operation implies one real addition or one real multiplication. Similarly, the error evaluation for swapping pixels can be efficiently implemented with LUTs.

4. EXPERIMENTAL RESULTS

In this section we evaluate the efficacy of the measurement approach. The measurement based printer model was developed at a printer resolution of 600 dpi on an HP LaserJet 4M printer. In our analysis, we use three images: Bullseye, Ramp and Bigtree. All images have been tone corrected prior to halftoning. In the following, we compare halftones generated by DBS with the measurement model with halftones generated by DBS with the HCD model and images printed using a clustered dot screen. Following the procedure suggested by Pappas and Neuhoff [2], we estimated the ratio of the ideal dot radius to the actual dot radius for the HCD model to be 1.40.

The halftones of the Bullseye pattern are presented in Figs. 3, 4 and 5. It can be observed that the halftone generated by the clustered dot screen, shown in Fig. 3, has objectionable moire patterns. No such artifacts exist in halftones generated by DBS with the measurement or HCD model. However, the halftone generated by DBS with the measurement model is significantly cleaner in the high frequency regions.

Figure 6 contains the halftones of the Ramp image. It can be seen from Fig. 6(a) that the clustered dot screen gives rise to visible steps in the gray scale ramp. The halftone generated by DBS with the HCD model, shown in Fig. 6(b), has in general a

noisy appearance, while the halftone generated by DBS with the measurement model, given in Fig. 6(c), is consistently smoother. The change in gray scale is more uniform throughout.

The halftones of the Big-tree image are given in Figs. 7, 8, and 9. The halftone produced by DBS with the HCD model has quantization artifacts, especially on the tree-trunk. It has a noisy appearance in the mid-tone regions, as is clear from Fig. 8. Since the HCD model does not accurately predict the dot interactions at 600 dpi as observed in Fig. 1(c), it is not surprising that DBS with the HCD model produces a poor halftone at 600dpi. The halftone produced by the clustered dot screen has more shadow detail than the one produced by DBS with the HCD model; but it is degraded by the easily visible halftone dots. In comparison, the halftone generated by DBS with the measurement model is sharper and has excellent detail rendition. Due to compression in the PDF format, the halftones may not appear as desired. The postscript version of the paper is available at <http://www.ecn.purdue.edu/EISL/pubs/ICASSP/baqai98.ps>

5. CONCLUSIONS

We have shown that a measurement-based printer model based on data collected from a high resolution scanner coupled with DBS yields images that have richer gray regions, very high spatial resolution and visually pleasing textures. Since the measurement model reflects the actual dot-interactions, it has the ability to better exploit the intermediate gray levels caused by dot overlap than other models. We have also shown that the algorithm can be efficiently implemented by recursively evaluating the effect of trial changes.

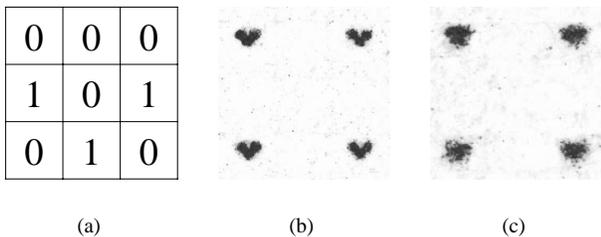


Figure 1: (a) Digital halftone and high resolution scan of printed output at: (b) 300 dpi (c) 600 dpi.

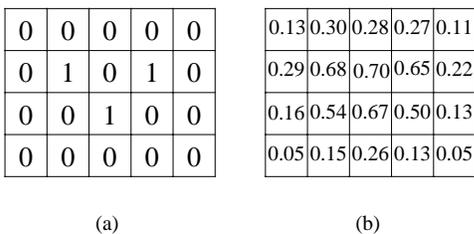


Figure 2: (a) Halftone (b) equivalent grayscale image.

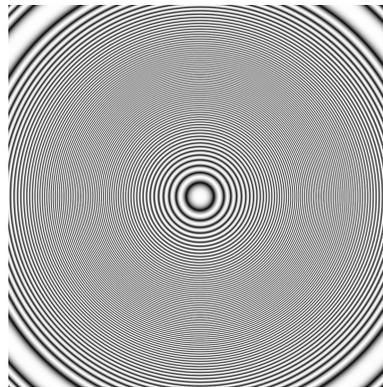


Figure 3: Bullseye halftoned by clustered dot screen.

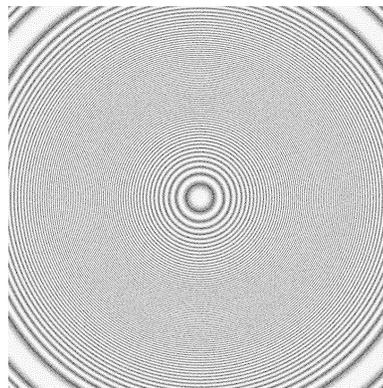


Figure 4: Bullseye halftoned by DBS with HCD model.

6. REFERENCES

- [1] R.L. Stevenson and G.R. Arce, "Binary display of hexagonally sampled continuous-tone images," *Journal of the Optical Society of America A*, vol. 2, no. 7, pp. 1009 – 1013, Jul. 1985.
- [2] T.N. Pappas and D.L. Neuhoff, "Printer models and error diffusion," *IEEE Trans. on Image Processing*, vol. 4, no. 1, pp. 66 – 80, Jan. 1995.
- [3] T.N. Pappas, C. Dong, and D.L. Neuhoff, "Measurement of printer parameters for model-based halftoning," *J. Electron. Imaging*, vol. 2, no. 3, pp. 193 – 204, Jul. 1993.
- [4] D.A. Carrara, M. Analoui, and J.P. Allebach, "Recent progress in digital halftoning," in *Proc. IS&T 8th Int. Cong. Adv. Non-Impact Print. Tech.*, Williamsburg, Oct. 1992, pp. 265 – 270.
- [5] F.A. Baqai, C.C. Taylor, and J.P. Allebach, "Halftoning via direct binary search using a hard circular dot overlap model," in *Proc. IS&T/OA Optics & Imaging in the Information Age*, Rochester, Oct. 20-24 1996, pp. 383 – 391.

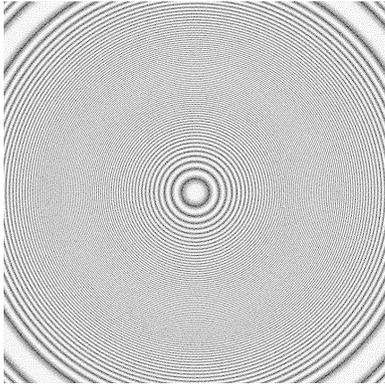
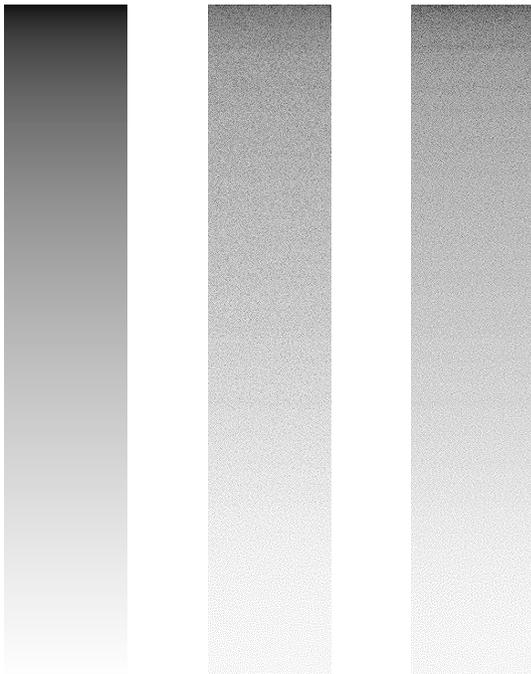


Figure 5: Bullseye halftoned by DBS with measurement model.



Figure 7: Big-tree halftoned by clustered dot screen.



(a)

(b)

(c)

Figure 6: Ramp halftoned by (a) clustered dot screen, (b) DBS with HCD model and (c) DBS with Measurement model.

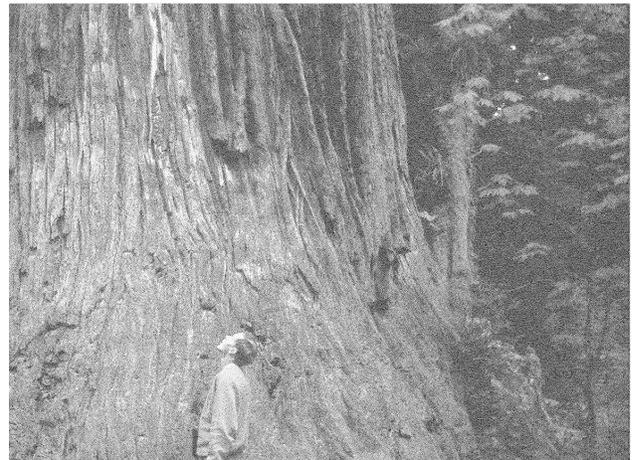


Figure 8: Big-tree halftoned by DBS with HCD model.

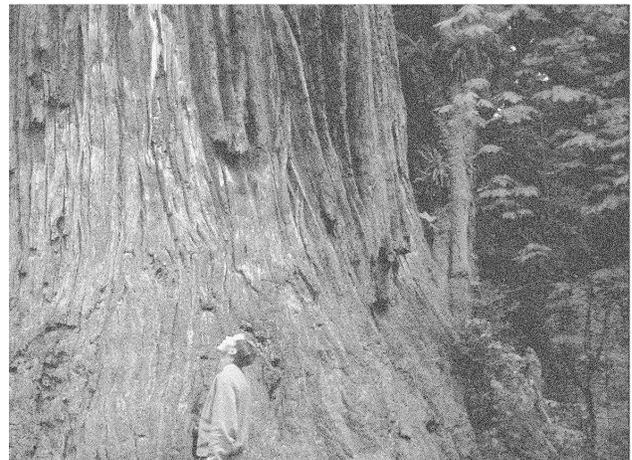


Figure 9: Big-tree halftoned by DBS with measurement model.