Hybrid Algorithms for Digital Halftoning and Their Application to Medical Imaging[*]

Eugene A. Sandler[†], Dmitri A. Gusev[‡], and Gregory Y. Milman[♠]

Abstract. Error diffusion, ordered dither, and patterning are well-known digital halftoning techniques, each with its advantages and disadvantages. Error diffusion is known for correlated artifacts looking like zebra stripes. Images produced by ordered dither and patterning suffer from artificial contours in slowly varying regions of pictures. Use of ordered matrices by both ordered dither and patterning results in poor rendition of small details, so the images appear blurred. We discuss how the modified Floyd–Steinberg error diffusion algorithm reduces correlated artifacts compared to the classical Floyd–Steinberg algorithm without increase in the amount of work. It also simplifies combining of error diffusion with other methods. We introduce two families of hybrid algorithms. One of them combines the modified Floyd–Steinberg algorithm with ordered dither and the other one combines it with patterning. We fight the correlated artifacts with ordered matrices while reducing the accompanying artificial contours by error diffusion. The latter achievement allows us to use smaller matrices and thus affect small detail representation at low resolutions. We show that hybrid algorithms are useful for digital simulation of classical screens. Combining error diffusion and patterning results in speedups essential for printing medical images at high resolutions. The article considers application of the hybrid algorithms to printing of medical images on laser printers.

Keywords: digital halftoning, error diffusion, serpentine raster, modified Floyd–Steinberg algorithm, blue noise, ordered dither, patterning, hybrid, classical screen, image printing, medical imaging, laser printer, resolution.

[*] This work was partially supported by NSF Grant CCR 94-02780.

[†] (1944–1995) Associate Professor, Department of Cybernetics, Moscow State Institute of Radioengineering, Electronics, and Automation (University of Technology).

[‡] Graduate student, Indiana University, Computer Science Department, Lindley Hall, Bloomington, IN 47405-4101 (**dmiguse@cs.indiana.edu**).

[♠] Chief Engineer-Researcher, VIDAR Ltd., 11 Yauzskaya st., bldg 12, rm 14, Moscow, Russia (**milm@vidar.msk.ru**).

1. Introduction

Human vision possesses ability spatially to average small luminance deviations [16 (p. 621)]. Error diffusion [4, 5], ordered dither [1, 12, 13, 14], patterning [7, 10, 15, 17 (Section 2.28)], and other algorithms for digital halftoning (see, for example, [6, 11, 22]) take advantage of this averaging process, which can be loosely described as two-dimensional "demodulation" of graphic information. So does the analog method known as the "classical" screen, which has been used to do halftoning since the nineteenth century [24 (Section 5.1)]. Readers interested in a comprehensive comparison of halftoning techniques are referred to [6, 11, 20, 24]. Many related neurobiological aspects of vision are discussed in [8]. Our opinions on benefits and deficiencies of known algorithms are based primarily on the material found in these sources. This introduction will provide descriptions and definitions forming the basis for presentation of the hybrid algorithms for digital halftoning.

Since Floyd and Steinberg presented the first *error diffusion* algorithms [4, 5], many modifications were proposed to reduce characteristic correlated artifacts (zebra patterns) [3, 9, 21, 23]. Most of such modifications involve significantly more processing than the early versions of error diffusion. The original Floyd–Steinberg error diffusion algorithm [4] for processing of a halftone image represented by an $m \times n$ array (matrix) A of real values between 0 and 1 can be written, after a slight modification involving use of a serpentine raster [24 (pp. 266–267)], as

```
for $i := 1$ to $m$ do
  begin
    if $i$ is odd then
      for $j := 1$ to $n$ do
        begin
          if $A_{i,j} < 1/2$ then $B_{i,j} := 0$ else $B_{i,j} := 1$;
          $\varepsilon := A_{i,j} - B_{i,j}$; {$\varepsilon$ is the current value of error.}
          $A_{i,j+1} := A_{i,j+1} + \varepsilon * \alpha$;
          $A_{i+1,j} := A_{i+1,j} + \varepsilon * \gamma$;
          $A_{i+1,j+1} := A_{i+1,j+1} + \varepsilon * \delta$;
        end;
    else
      for $j := n$ downto 1 do
        begin
          if $A_{i,j} < 1/2$ then $B_{i,j} := 0$ else $B_{i,j} := 1$;
          $\varepsilon := A_{i,j} - B_{i,j}$;
          $A_{i,j-1} := A_{i,j-1} + \varepsilon * \alpha$;
          $A_{i+1,j} := A_{i+1,j} + \varepsilon * \gamma$;
          $A_{i+1,j-1} := A_{i+1,j-1} + \varepsilon * \delta$;
        end;
  end.
```

Our implementation of the *modified Floyd–Steinberg algorithm* uses $\alpha = \gamma = 14/38 \approx 0.368$, $\delta = 10/38 \approx 0.263$. These values of *error diffusion coefficients (weights)* are close to the ones recommended by Floyd and Steinberg ($\alpha = \gamma = 3/8 = 0.375$, $\delta = 1/4 = 0.25$) [4]. The $m \times n$ array (matrix) B of zeros (black dots) and ones (white dots) serves as output of the algorithm. B represents a binary approximation to the input image.

The *classical Floyd–Steinberg algorithm* [5] has one more weight. It can be written as

```
for $i := 1$ to $m$ do
  for $j := 1$ to $n$ do
    begin
      if $A_{i,j} < 1/2$ then $B_{i,j} := 0$ else $B_{i,j} := 1$;
      $\varepsilon := A_{i,j} - B_{i,j}$; {$\varepsilon$ is the current value of error.}
      $A_{i,j+1} := A_{i,j+1} + \varepsilon * \alpha$;
      $A_{i+1,j-1} := A_{i+1,j-1} + \varepsilon * \beta$;
      $A_{i+1,j} := A_{i+1,j} + \varepsilon * \gamma$;
      $A_{i+1,j+1} := A_{i+1,j+1} + \varepsilon * \delta$;
    end.
```

Our implementation uses the error diffusion coefficients recommended by the authors: $\alpha = 7/16$, $\beta = 3/16$, $\gamma = 5/16$, and $\delta = 1/16$. It is straightforward to see that the amounts of processing required by the classical Floyd–Steinberg algorithm and the modified Floyd–Steinberg algorithm are roughly the same.

Most illustrations in this paper are printed at the resolution of 300 dpi on an HP LaserJet IVsi laser printer. The other illustrations are printed on the same printer at 600 dpi: discrepancies between the pictures produced by different digital halftoning algorithms are generally harder to notice at this and higher resolutions (cause and implications of this phenomenon will be discussed later).

The same digitized photograph (portrait of Anya Pogosyants) was used as source for all pictures shown in Figure 1. Figure 1(a) features results of digital halftoning by the classical Floyd–Steinberg algorithm at 300 dpi. Figures 1(b) and 1(c) illustrate the results of halftoning by the modified Floyd–Steinberg algorithm at 300 and 600 dpi, respectively. As Figures 1(a) and 1(b) demonstrate, the modified Floyd–Steinberg algorithm reduces correlated artifacts compared to the classical one. We will discuss the reasons of it in the next section, where our choice of the modified Floyd–Steiberg algorithm for the purpose of combining error diffusion with other techniques is justified.

Halftoning by *ordered dither* can be described as follows [24 (Chapter 5)]. An ordered dither algorithm generates a binary halftone image by comparing pixels from an original continuous-tone image to a threshold value from a deterministic, periodic array. The thresholds are "ordered" rather than "random". Ordered dither is a point operation, that is, the output depends only on the state of the current pixel.

Patterning maps an input pixel into a *cell* (tile, pattern, character of a gray shade font) with the average intensity value close to the input value. This technique is also known as PSAM (Pulse-Surface-Area Modulation), but we prefer the simpler name [17 (Section 2.28)]. One way to do patterning would be pick output pixels from imaginary planes tiled with the patterns. This approach could give us a one to one mapping. However, if we want to avoid unpleasant effects along the borders of areas of different intensity, we will arrive at equivalents of ordered dither. It is more common to replace each input pixel by a cell: an implied scale change occurs.

There are two main problems with ordered dither and patterning. One of them is poor rendition of small details: as a result, images appear blurred. The other one is presence of artificial contours in the areas with slowly varying intensity. This effect, often called *contouring*, interferes with the use of matrices smaller than 8×8: the number of levels of gray that can be represented by distinct matrices is not large enough to make artificial contours reasonably inconspicuous. On the other hand, use of larger matrices hurts representation of small details too much. Interestingly enough, the very presence of the matrix patterns is liked by some people and disliked by others. Generally, the readers should keep in mind that the evaluation of how relatively pleasant different halftoned images are remains a somewhat subjective process.

There were several attempts to combine use of matrices with error diffusion. Knuth [11] introduced *dot diffusion* and *smooth dot diffusion*, where 8×8 matrices were used to control the flow of error diffusion. The dot diffusion technique fails to get rid of artificial contours, and the image blurring effect is significant. Smooth dot diffusion is more involved computationally, and the small image details are still blurred. Stephen [20] presented a fairly complex algorithm using blue noise (randomized error diffusion) to dynamically generate sets of 8×8 cells (gray shade fonts). The halftoning time for a 256×256 image turned out to be more than 20 times larger than that of conventional 8×8 clustered-dot font patterning. Earlier, Billotet-Hoffman and Bryngdahl [2] proposed using conventional 8×8 ordered dither arrays in place of the fixed threshold used in the classical Floyd–Steinberg error diffusion algorithm. However, the resulting methods failed to break its strong correlated artifacts and significantly worsened small detail representation.

What went wrong? At the distances from which people usually look at halftoned pictures, the resolution capacity of their vision system is between 100 and 130 dpi. As a result, for combining error diffusion with matrix-using techniques and subsequent presentation of the results at low resolutions, 8×8 matrices are simply too large. Also, as we shall see in the next section, the classical Floyd–Steinberg algorithm is not the best choice for basing hybrid algorithms upon. Our hybrid algorithms use other varieties of error diffusion to reduce contouring, while fighting the zebra stripes with small ordered matrices.

2. Advantages of the Modified Floyd–Steinberg Algorithm

On the intuitive level, the reasons why the modified Floyd–Steinberg algorithm is better than the classical one can be explained as follows.

Following the approach presented in [18, 19], we can interpret outputs $B_{i,j}$ of a digital halftoning algorithm as values of random variables $e_{i,j}$. Let S be an area of the image, consisting of pixels that are close together, and let $T(S)$ be the set of all possible two-element subsets $\{(i_1, j_1), (i_2, j_2)\}$ of S. Let the correlation coefficient of e_{i_1,j_1} and e_{i_2,j_2} be denoted as $\rho(e_{i_1,j_1}, e_{i_2,j_2})$. Because of the "demodulation" property of human vision, it is desirable to construct $e_{i,j}$ so that the variance

$$V(\sum_S e_{i,j}) = \sum_S V(e_{i,j}) + 2 \sum_{T(S)} \rho(e_{i_1,j_1}, e_{i_2,j_2}) \tag{1}$$

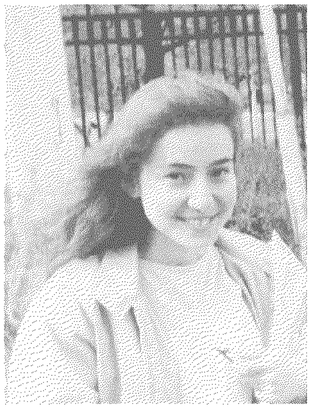
is minimum on the condition that the expected values

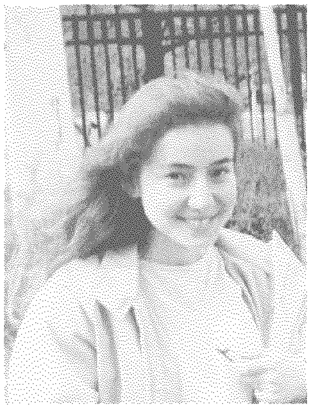
$$E(e_{i,j}) = A_{i,j} \tag{2}$$

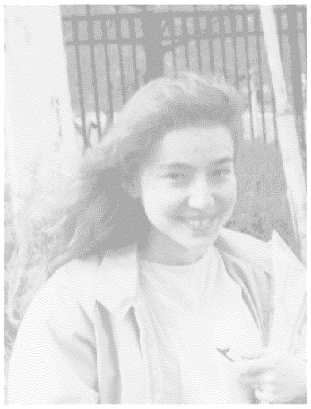
for all (i, j) in S.

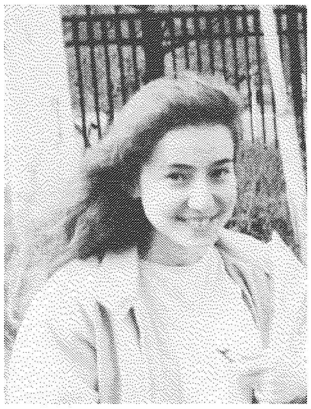
Notice that the underlying assumption that the vision system averages intensity levels of pixels in S with equal weights is just an approximation. Moreover, direct transition from the local quasi-optimality criterion to a practical halftoning algorithm is nontrivial and is related to an open problem posed in [19]. A good solution to the problem may result in improvement of quality of digital halftoning. This may also be a way to find a better generator of *blue noise* [24 (Chapter 8)] than the known digital halftoning algorithms. Anyhow, the closer together any two pixels are, the less correlated the corresponding random variables should be (on the condition that their expected values coincide with the inputs). In our opinion, error diffusion algorithms are better whenever they come closer to meeting this intuitive requirement. Many other approaches have also been used to evaluate quality of digital halftoning [1, 6, 16, 24]. A more extensive discussion of error metrics would go beyond the scope of the paper.

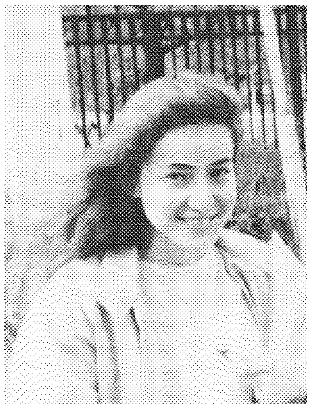
As Figure 2 shows, the serpentine raster of the modified Floyd–Steinberg algorithm allows it to take more neighbors into account.

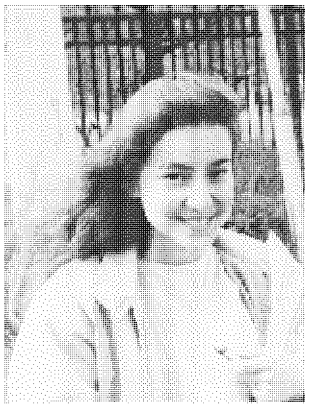

a)

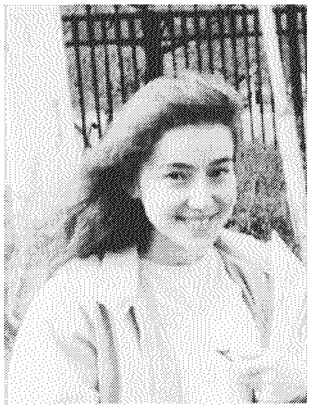

b)

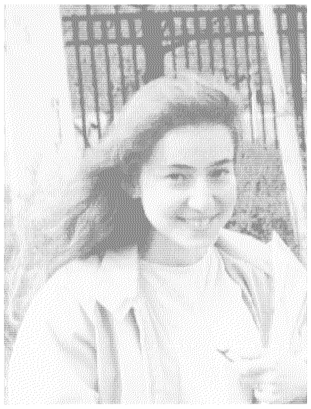

c)


d)


e)


f)


g)


h)

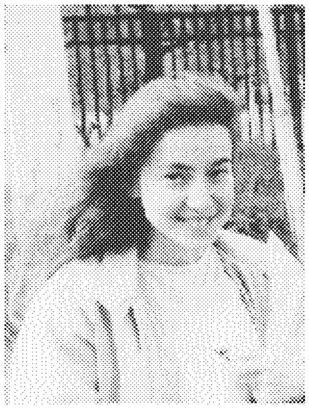

i)

Fig. 1. Portrait of Anya Pogosyants

a) Classical Floyd–Steinberg Algorithm, 300 dpi
b) Modified Floyd–Steinberg Algorithm, 300 dpi
c) Modified Floyd–Steinberg Algorithm, 600 dpi
d) Dithered Serpentine Diffusion, 4×4 matrix, 300 dpi
e) Dithered Serpentine Diffusion, 6×6 matrix, 300 dpi
f) Patterned Serpentine Diffusion, 3×3 tiles, 300 dpi
g) Patterned Double-Cross Diffusion, 2×2 tiles, 300 dpi
h) Patterned Double-Cross Diffusion, 2×2 tiles, 600 dpi
i) Patterned Double-Cross Diffusion, 3×3 tiles, 300 dpi

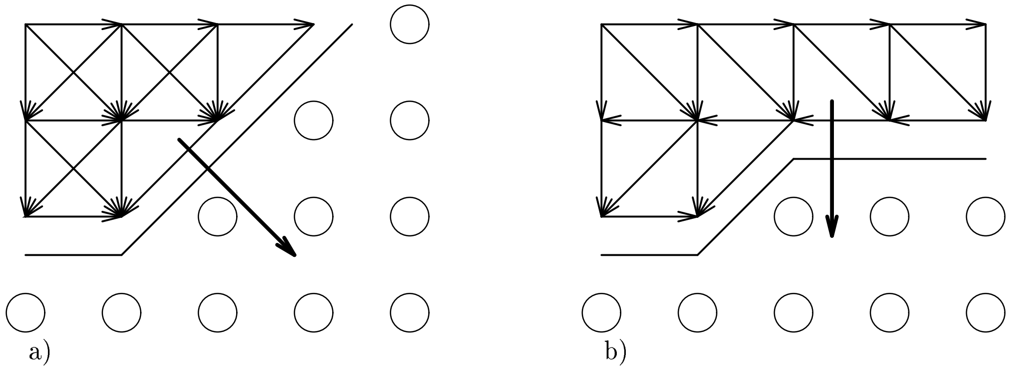


Fig. 2. Flow of Error Diffusion
a) For the Classical Floyd–Steinberg Algorithm
b) For the Modified Floyd–Steinberg Algorithm

The correlated artifacts (zebra stripes) tend to be roughly perpendicular to the general direction of error diffusion. The theory behind the fact is presented in [19]. Figure 2(b) shows that the flow of error diffusion for the modified Floyd–Steinberg algorithm is nicely balanced in the horizontal direction. This simplifies choice of weights and makes the orientation of remaining correlated artifacts approximately horizontal. The latter feature, along with the overall reduction of the zebra stripes, makes the modified Floyd–Steinberg error diffusion algorithm a good candidate for combining with ordered dither and patterning.

3. Combining Error Diffusion on a Serpentine Raster and Ordered Dither

Let D be an $M \times N$ matrix of real values between 0 and 1. This section introduces *dithered serpentine diffusion*, the technique obtained by substituting $A_{i,j} < D_{(i \bmod M)+1,(j \bmod N)+1}$ for $A_{i,j} < 1/2$ in the modified Floyd–Steinberg error diffusion algorithm.

Our implementations of dithered serpentine diffusion employ the weights chosen earlier for the modified Floyd–Steinberg algorithm. Figure 1(d) corresponds to the 4×4 matrix

$$D = \begin{pmatrix} 1/9 & 2/9 & 5/9 & 6/9 \\ 4/9 & 3/9 & 8/9 & 7/9 \\ 5/9 & 6/9 & 1/9 & 2/9 \\ 8/9 & 7/9 & 4/9 & 3/9 \end{pmatrix}. \tag{3}$$

Figure 1(e) demonstrates the effect of the 6×6 matrix

$$D = \begin{pmatrix} 13/19 & 15/19 & 10/19 & 9/19 & 3/19 & 6/19 \\ 16/19 & 18/19 & 14/19 & 5/19 & 1/19 & 2/19 \\ 11/19 & 17/19 & 12/19 & 7/19 & 4/19 & 8/19 \\ 9/19 & 3/19 & 6/19 & 13/19 & 15/19 & 10/19 \\ 5/19 & 1/19 & 2/19 & 16/19 & 18/19 & 14/19 \\ 7/19 & 4/19 & 8/19 & 11/19 & 17/19 & 12/19 \end{pmatrix}. \tag{4}$$

Comparison with Figures 1(a,b) shows that dithered serpentine diffusion achieves further reduction of the correlated artifacts and breaks their directionality at the cost of image blurring. Artificial contours and extra granularity noticeable in the case of the 4×4 matrix subside when the matrix size is increased. Very light tones retain some correlated artifacts, and appearance of some of the corresponding areas even gets worse. In the meanwhile, the gray areas acquire structure similar to that of the 45° classical screen (this is due to our choices of D). The quality of halftoning by dithered serpentine diffusion may be improved through adjustment of the error diffusion coefficients and/or D.

Notice that the speed of the dithered serpentine diffusion algorithms is slightly worse than that of the modified Floyd–Steinberg algorithm. Faster hybrid algorithms are given in the next two sections.

4. Combining Error Diffusion on a Serpentine Raster and Patterning

Let $C^{(0)}, C^{(1)}, \ldots, C^{(MN)}$ be $M \times N$ matrices of zeros and ones such that, for any k, $0 \le k \le MN$, $C^{(k)}$ has exactly k zero elements. Let $\{l_0 = 0, l_1 = 1/(MN), \ldots, l_{M \cdot N} = 1\}$ be a set of $MN+1$ equidistant quantization levels.

Patterned serpentine diffusion is a combination of the modified Floyd–Steinberg algorithm and patterning. Its input A is the same as before, but the elements of the output matrix B take values from $\{C^{(k)}\}$. The technique can be described algorithmically as follows.

```
for $i := 1$ to $m$ do
  begin
    if $i$ is odd then
      for $j := 1$ to $n$ do
        begin
          $q :=$ the index of the nearest of $l_k$ $(k = 0, \ldots, MN)$ to $A_{i,j}$;
          $B_{i,j} := C^{(q)}$;
          $\varepsilon := A_{i,j} - l_q$; {$\varepsilon$ is the current value of error.}
          $A_{i,j+1} := A_{i,j+1} + \varepsilon * \alpha$;
          $A_{i+1,j} := A_{i+1,j} + \varepsilon * \gamma$;
          $A_{i+1,j+1} := A_{i+1,j+1} + \varepsilon * \delta$;
        end;
    else
      for $j := n$ downto $1$ do
        begin
          $q :=$ the index of the nearest of $l_k$ $(k = 0, \ldots, MN)$ to $A_{i,j}$;
          $B_{i,j} := C^{(q)}$;
          $\varepsilon := A_{i,j} - l_q$;
          $A_{i,j-1} := A_{i,j-1} + \varepsilon * \alpha$;
          $A_{i+1,j} := A_{i+1,j} + \varepsilon * \gamma$;
          $A_{i+1,j-1} := A_{i+1,j-1} + \varepsilon * \delta$;
        end;
  end.
```

Our implementations of patterned serpentine diffusion use the weights chosen for the modified Floyd–Steinberg algorithm. One of the algorithms uses 3×3 matrices

$$C^{(0)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad C^{(1)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad C^{(2)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix},$$

$$C^{(3)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \quad C^{(4)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad C^{(5)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad C^{(6)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \qquad (5)$$

$$C^{(7)} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad C^{(8)} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad C^{(9)} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix},$$

to simulate a 0° classical screen. Its effect is shown in Figure 1(f). Again, the correlated artifacts are curbed at the cost of increased granularity, image blurring, and some contouring. All these negative effects except the image blurring can be reduced by using larger tiles. The structure of the tiles appears to be significant, too.

The other implementation of patterned serpentine diffusion involves 4×4 matrices

$$\begin{gathered}
C^{(0)} = \begin{pmatrix} 0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0 \end{pmatrix}, \quad
C^{(1)} = \begin{pmatrix} 0&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&0 \end{pmatrix}, \quad
C^{(2)} = \begin{pmatrix} 0&0&0&0\\0&0&0&1\\0&1&0&0\\0&0&0&0 \end{pmatrix}, \\
C^{(3)} = \begin{pmatrix} 0&0&1&0\\0&0&0&0\\1&0&0&0\\0&0&0&1 \end{pmatrix}, \quad
C^{(4)} = \begin{pmatrix} 0&0&1&0\\1&0&0&0\\0&0&0&1\\0&1&0&0 \end{pmatrix}, \quad
C^{(5)} = \begin{pmatrix} 0&1&0&0\\0&0&0&1\\1&0&1&0\\0&0&1&0 \end{pmatrix}, \\
C^{(6)} = \begin{pmatrix} 1&0&0&1\\0&1&0&0\\0&0&0&1\\1&0&0&1 \end{pmatrix}, \quad
C^{(7)} = \begin{pmatrix} 1&0&1&0\\0&1&0&1\\0&0&1&0\\0&1&0&1 \end{pmatrix}, \quad
C^{(8)} = \begin{pmatrix} 0&1&0&1\\1&0&1&0\\0&1&0&1\\1&0&1&0 \end{pmatrix}, \\
C^{(9)} = \begin{pmatrix} 1&0&1&0\\0&1&0&1\\1&0&1&1\\1&0&1&0 \end{pmatrix}, \quad
C^{(10)} = \begin{pmatrix} 0&1&0&1\\1&0&1&1\\1&1&0&1\\1&0&1&0 \end{pmatrix}, \quad
C^{(11)} = \begin{pmatrix} 1&1&0&1\\0&1&1&0\\1&1&1&0\\1&0&1&1 \end{pmatrix}, \\
C^{(12)} = \begin{pmatrix} 1&0&1&1\\1&1&1&0\\0&1&1&1\\1&1&0&1 \end{pmatrix}, \quad
C^{(13)} = \begin{pmatrix} 1&1&1&1\\1&0&0&1\\1&0&1&1\\1&1&1&1 \end{pmatrix}, \quad
C^{(14)} = \begin{pmatrix} 1&1&1&1\\0&1&1&1\\1&1&0&1\\1&1&1&1 \end{pmatrix}, \\
C^{(15)} = \begin{pmatrix} 1&1&1&1\\1&1&1&1\\1&0&1&1\\1&1&1&1 \end{pmatrix}, \quad
C^{(16)} = \begin{pmatrix} 1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1 \end{pmatrix}.
\end{gathered} \tag{6}$$

The apparent asymmetry is related to the issue of image printing, so the discussion of this algorithm will take place in Section 6.

The implied scale change can be implemented very efficiently. Moreover, this kind of scaling is often needed even for the algorithms allowing a one to one mapping. In the meanwhile, the number of operations required by a patterned serpentine diffusion algorithm is roughly MN times smaller than the number of operations needed by each of the other error-diffusion-based algorithms considered so far to produce an output image of the same size. Difference in the nature of the operations, along with necessary overheads, accounts for the fact that this dependency is just an approximation.

5. Fast Hybrid Algorithms for Emulation of 45° Classical Screens

In the course of our digital halftoning research, we met a few people who liked classical screens, and a few people who didn't. Did these meetings influence the direction of our work? Somewhat. In Section 3, we showed that the dithered serpentine diffusion algorithms can emulate 45° screens. In this section, we present faster hybrid algorithms capable of doing the same thing (one of them happened to be good for medical image printing).

Let $R = \{R^{(0)}, R^{(1)}, \ldots, R^{(2N^2)}\}$ and $W = \{W^{(0)}, W^{(1)}, \ldots, W^{(2N^2)}\}$ be sets of $N \times N$ matrices of zeros and ones. Let $\{l_0 = 0, l_1 = 1/(2N^2), \ldots, l_{2N^2} = 1\}$ be a set of $2N^2 + 1$ equidistant quantization levels.

Patterned double-cross diffusion is a hybrid of error diffusion and patterning. Values assigned to the entries of B are elements of $R \cup W$. The algorithmic description of the process is as follows.

```
$c := (n \bmod 2)$;
**for** $i := 1$ **to** $m$ **do**
  **begin**
    $d := (i \bmod 2) + 1$;
    **for** $j := d$ **by** 2 **to** $n - c - (d \bmod 2)(1 - 2c)$ **do**
      **begin**
        $q :=$ the index of the nearest of $l_k$ $(k = 0, \ldots, 2N^2)$ to $A_{i,j}$;
        $B_{i,j} := R^{(q)}$;
        $\varepsilon := A_{i,j} - l_q$; {$\varepsilon$ *is the current value of error.*}
        $A_{i,j+2} := A_{i,j+2} + \varepsilon * \alpha$;
        $A_{i+1,j-1} := A_{i+1,j-1} + \varepsilon * \beta$;
        $A_{i+1,j+1} := A_{i+1,j+1} + \varepsilon * \delta$;
      **end**;
    **for** $j := n + c + (d \bmod 2)(1 - 2c) - 1$ **by** 2 **downto** $(d \bmod 2) + 1$ **do**
      **begin**
        $q :=$ the index of the nearest of $l_k$ $(k = 0, \ldots, 2N^2)$ to $A_{i,j}$;
        $B_{i,j} := W^{(q)}$;
        $\varepsilon := A_{i,j} - l_q$;
        $A_{i,j-2} := A_{i,j-2} + \varepsilon * \alpha$;
        $A_{i+1,j+1} := A_{i+1,j+1} + \varepsilon * \beta$;
        $A_{i+1,j-1} := A_{i+1,j-1} + \varepsilon * \delta$;
      **end**;
  **end**.
```

Figure 3 shows the flow of double-cross error diffusion.

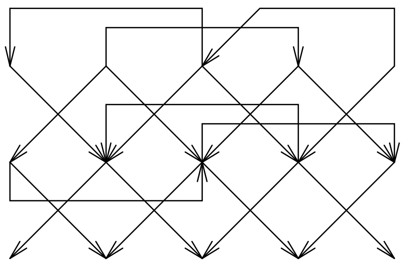


Fig. 3. Flow of Double-Cross Error Diffusion

We implemented two patterned double-cross diffusion algorithms, using $\alpha = 0.32$, $\beta = \delta = 0.29$. The sets of cells R and W for one of them were obtained by using thresholds $17/18, 15/18, \ldots, 1/18$ to perform conventional two-level quantization of matrices

$$\tilde{R} = \begin{pmatrix} 4/9 & 3/9 \\ 2/9 & 1/9 \end{pmatrix}, \quad \tilde{W} = \begin{pmatrix} 5/9 & 7/9 \\ 6/9 & 8/9 \end{pmatrix}. \tag{7}$$

Figures 1(g) and 1(h) demonstrate the effect of this algorithm applied to the digitized portrait of Anya Pogosyants at 300 dpi and 600 dpi. The algorithm was applied to medical image printing as well. This application will be discussed in the next section.

The other patterned double-cross diffusion algorithm we implemented uses sets of cells produced by conventional two-level quantization of

$$\tilde{R} = \begin{pmatrix} 6/19 & 4/19 & 9/19 \\ 2/19 & 1/19 & 5/19 \\ 8/19 & 3/19 & 7/19 \end{pmatrix}, \quad \tilde{W} = \begin{pmatrix} 10/19 & 16/19 & 13/19 \\ 14/19 & 18/19 & 17/19 \\ 12/19 & 15/19 & 11/19 \end{pmatrix} \tag{8}$$

with thresholds $37/38, 35/38, \ldots, 1/38$. Its effect is shown in Figure 1(i).

Look at similarities and differences between Figures 1(d) and 1(g), then compare Figures 1(e) and 1(i). The patterned double-cross diffusion algorithms appear to curb correlated artifacts stronger than the corresponding dithered serpentine diffusion algorithms. This is achieved at the expense of small details of the image. Occurrence of characteristic vertical artifacts may be further fought by means of weight adjustment. Differences become smaller when the matrix sizes increase.

Patterned double-cross algorithms outperform error diffusion and dithered serpentine diffusion: approximately N^2 times fewer operations are required. Actual savings aren't as high due to difference in the nature of operations and presence of overheads.

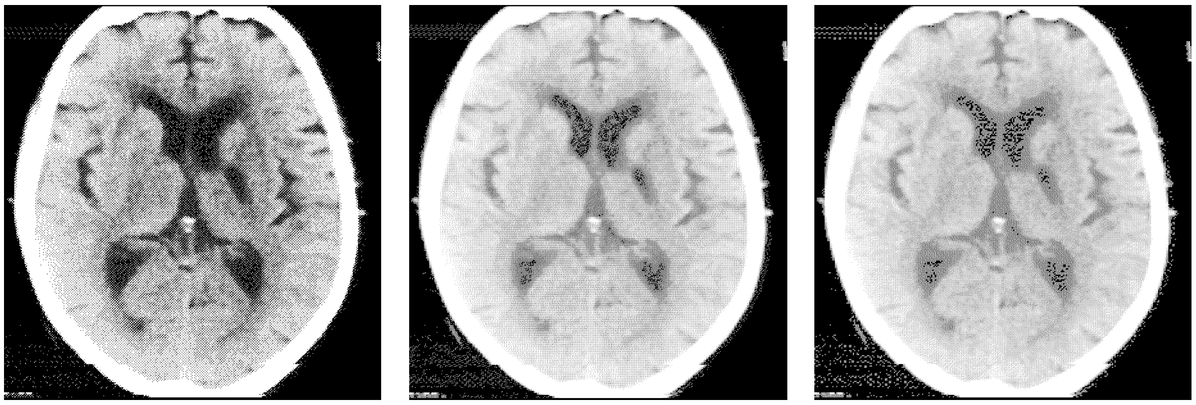
6. Application of Hybrid Algorithms to Medical Image Printing

Three of our hybrid algorithms found application in the medical imaging software by VIDAR Ltd. Traditionally, the printing of computer tomography images and ultrasound diagnostics images had been done on thermo printers. The image printing on laser printers is less expensive, and our hybrid algorithms ensured good quality at high speed.

Before we cover the details, a preliminary remark about image printing in general. *Tone scale adjustment* [24 (Section 1.3)] is an essential part of image printing on laser printers. In particular, it is usually necessary to compensate for broadening of black pixels which darkens images. Despite our having performed this procedure in the course of illustration preparation for this manuscript, slight differences of brightness of some areas of similar images may be observed in Figures 1, 4, and 5. These differences are small compared to what happens when the same image is printed at different toner levels. Figure 6 shows gray scale ramps produced **without** tone scale adjustment. The appearance of a ramp does not depend solely on the digital halftoning algorithm applied. It is influenced by the choice of a printer, the toner level, and the resolution. In particular, algorithms equally good at representing intensity levels by appropriate proportions of black and white pixels may produce very different ramps, because the dots produced by the printer are not ideal pixels.

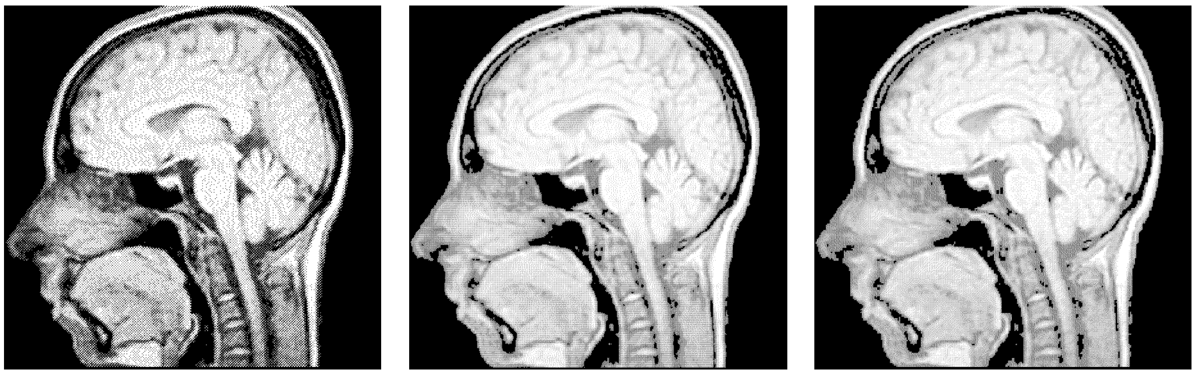
It is not a good idea to print medical images at 300 dpi, unless one wants a hard copy of an unimportant image fast. For this purpose, patterned double-cross diffusion with sets of patterns generated from the matrices in Eq. (7) was used. This algorithm is also good for printing at 600 dpi, but the patterned serpentine diffusion algorithm with tiles from Eq. (6) was selected instead, because it is faster. Limits on resolution capacity of the vision system cause the differences between digital halftoning algorithms to become less conspicuous at higher resolutions of printing and allow the use of more primitive, faster algorithms.

Pictures shown in Figures 4 and 5 feature two CAT scan images: the cyst (Figure 4) and the head of a healthy man (Figure 5). The cyst (a closed sac having a distinct membrane and developing abnormally in someone's brain) is the obnoxious dark area in the right-hand side of the first image. Figures 4(a) and 5(a) are printed at 300 dpi, using the appropriate patterned double-cross diffusion algorithm. Figures 4(b) and 5(b) show outputs of the same algorithm at 600 dpi. Figures 4(c) and 5(c) represent results of the faster patterned serpentine diffusion algorithm. In reality, hard copies of medical images are approximately four times larger (each size being doubled) than our illustrations. Figures 1(b,c,g,h) help to illustrate the effect of improved resolution. Quality difference between Figures 1(c) and 1(h) is smaller than between Figures 1(b) and 1(g).



a) b) c)

Fig. 4. The Cyst
a) Patterned Double-Cross Diffusion, 2×2 tiles, 300 dpi
b) Patterned Double-Cross Diffusion, 2×2 tiles, 600 dpi
c) Patterned Serpentine Diffusion, 4×4 tiles, 600 dpi



a) b) c)

Fig. 5. The Head of a Healthy Man
a) Patterned Double-Cross Diffusion, 2×2 tiles, 300 dpi
b) Patterned Double-Cross Diffusion, 2×2 tiles, 600 dpi
c) Patterned Serpentine Diffusion, 4×4 tiles, 600 dpi

For medical image printing, VIDAR Ltd. recommends LEXMARK printers, which seem to need less tone scale adjustment than, say, HP LaserJet IVsi used to prepare illustrations in this paper. The asymmetry in Eq. (6) allows to further reduce the necessary adjustment.

Finally, a patterned serpentine diffusion algorithm using 8×8 matrices was developed for medical image printing at 1200 dpi on LEXMARK laser printers. It is analogous to the algorithm for printing at 600 dpi.

Acknowledgments. The authors thank Paul W. Purdom, Jr. and Peter Shirley for helpful suggestions on presentation of the paper. We are deeply grateful to Gregory Pogosyants for his permission to use the digitized portrait of his daughter, Anya Pogosyants (1969–1995). A grad student at the Massachusetts Institute of Technology, Ms. Pogosyants was involved in research of properties of randomness-containing programs and other topics in computer-aided verification and distributed computing. The medical images in this paper are published with permission of VIDAR Ltd.

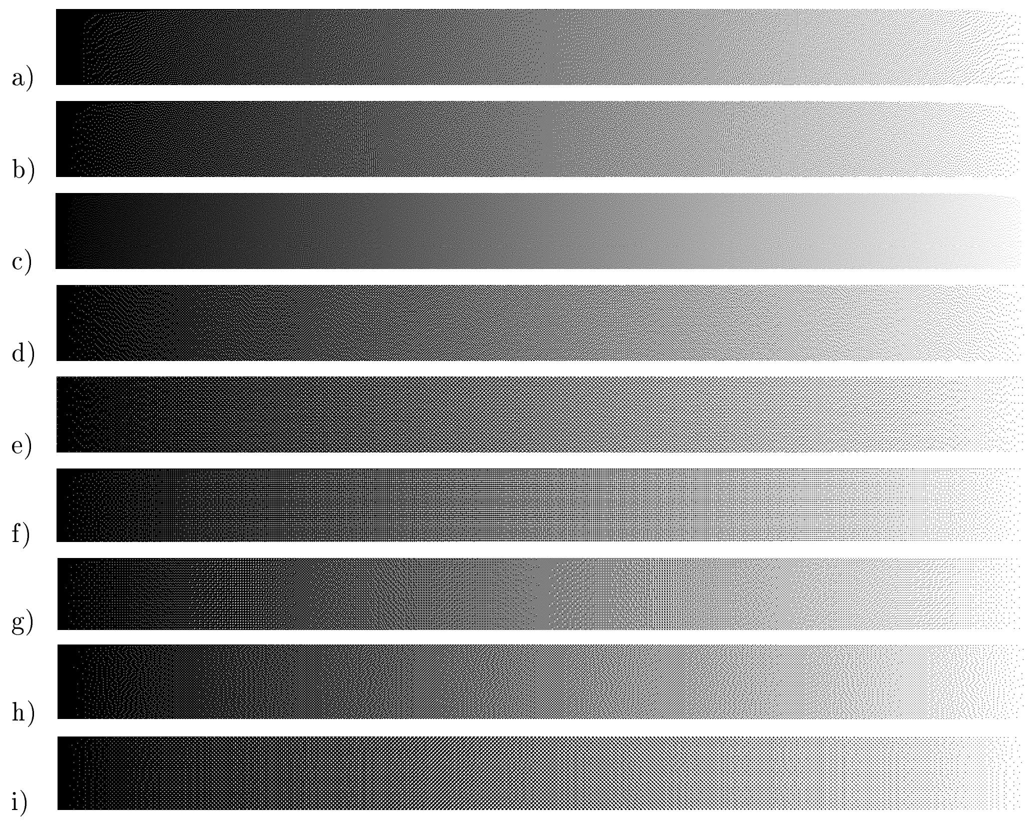


Fig. 6. Gray Scale Ramp
a) Classical Floyd–Steinberg Algorithm, 300 dpi
b) Modified Floyd–Steinberg Algorithm, 300 dpi
c) Modified Floyd–Steinberg Algorithm, 600 dpi
d) Dithered Serpentine Diffusion, 4×4 matrix, 300 dpi
e) Dithered Serpentine Diffusion, 6×6 matrix, 300 dpi
f) Patterned Serpentine Diffusion, 3×3 tiles, 300 dpi
g) Patterned Serpentine Diffusion, 4×4 tiles, 300 dpi
h) Patterned Double-Cross Diffusion, 2×2 tiles, 300 dpi
i) Patterned Double-Cross Diffusion, 3×3 tiles, 300 dpi

References

1. B. E. Bayer, *An Optimum Method for Two-Level Rendition of Continuous-Tone Pictures*, Conference Record, IEEE International Conference on Communications **1**, IEEE, New York (1973) pp. (26-11)–(26-15).
2. C. Billotet-Hoffman and O. Bryngdahl, *On the Error Diffusion Technique for Electronic Halftoning*, Proc. of Society for Information Display **24** (1983) pp. 253–258.
3. R. Eschabach, *Error Diffusion Algorithm with Reduced Artifacts*, IS&T's 45th Annu. Conf. "Imaging '92", New World Explore, Emerg. Technol. for the Year 2000, Springfield (1992) pp. 133–136.
4. R. W. Floyd and L. Steinberg, *An Adaptive Algorithm for Spatial Grey Scale*, SID 75 Digest, Society for Information Display (1975) pp. 36–37.
5. R. W. Floyd and L. Steinberg, *An Adaptive Algorithm for Spatial Greyscale*, Proc. of Society for Information Display **17/2** (1976) pp. 75–77.
6. Robert Geist, Robert Reynolds, and Darrell Suggs, *A Markovian Framework for Digital Halftoning*, ACM Transactions on Graphics **12** (1993) pp. 136–159.

7. P. Hamill, *Line Printer Modification for Better Grey Level Pictures*, Computer Graphics and Image Processing **6** (1977) pp. 485–491.
8. David H. Hubel, *Eye, Brain, and Vision*, Scientific American Library, New York (1988).
9. J. F. Jarvis, C. N. Judice, and W. H. Ninke, *A Survey of Techniques for the Display of Continuous-Tone Pictures on Bilevel Displays*, Computer Graphics and Image Processing **5** (1976) pp. 13–40.
10. K. Knowlton and L. Harmon, *Computer-Produced Greyscales*, Computer Graphics and Image Processing **1** (1972) pp. 1–20.
11. Donald E. Knuth, *Digital Halftones by Dot Diffusion*, ACM Transactions on Graphics **6** (1987) pp. 245–273.
12. J. O. Limb, *Design of Dither Waveforms for Quantized Visual Signals*, Bell Syst. Tech. J. **48** (1969) pp. 2555–2582.
13. B. Lippel and M. Kurland, *The Effect of Dither on Luminance Quantization of Pictures*, IEEE Trans. Commun. Tech. **COM-19** (1971) pp. 879–888.
14. Alan W. Paeth, *Ideal Tiles for Shading and Halftoning*, in Paul S. Heckbert (ed.), *Graphic Gems IV*, Academic Press Professional, Boston (1994), pp. 486–492.
15. B. Perry and M. L. Mendelsohn, *Picture Generation with a Standard Line Printer*, Comm. of the ACM **7** (1964) pp. 311–313.
16. William K. Pratt, *Digital Image Processing*, John Wiley & Sons, New York (1978).
17. David F. Rogers, *Procedural Elements for Computer Graphics*, McGraw-Hill, New York (1985).
18. Eugene A. Sandler, Gregory Y. Milman, Dmitri A. Gusev, *New Methods for Computer-Aided High-Quality Printing of Halftone Images*, Proceedings of the International Exhibition-Seminar COGRAPH-93 (Computational Geometry and Computer Graphics in Education), Nizhni Novgorod State University of Technology, Nizhni Novgorod (1993) p. 48 [in Russian].
19. Eugene A. Sandler, Dmitri A. Gusev, Gregory Y. Milman, and Mikhail L. Podolsky, *On Statistical Properties of Delta-Sigma-Modulated Sequences*, Technical Report No. 447, Computer Science Department, Indiana University (1996).
20. Graham A. Stephen, *A Comparison of Selected Digital Halftoning Techniques*, Microproc. and Microsys. **15** (1991) pp. 249–255.
21. R. L. Stevenson and G. R. Arce, *Binary Display of Hexagonally Sampled Continuous-Tone Images*, J. Opt. Soc. Am. A, **2** (1985) pp. 1009–1013.
22. P. Stucki, *Image Processing for Document Reproduction*, in P. Stucki (ed.), *Advances in Digital Image Processing*, Plenum Press, New York (1979).
23. P. Stucki, *MECCA — a Multiple-Error Correcting Computation Algorithm for Bilevel Image Hardcopy Reproduction*, Research Report RZ1060, IBM Research Laboratory, Zurich (1981).
24. Robert Ulichney, *Digital Halftoning*, The MIT Press, Cambridge, Mass. (1987).