Non-independent Randomized Rounding —
A Competitive Approach to the Digital
Halftoning Problem

Benjamin Doerr and Henning Schnieder

Mathematisches Seminar II, Christian-Albrechts-Universität zu Kiel, Ludewig-Meyn-Str. 4, D-24098 Kiel, Germany,
\{bed,hes\}@numerik.uni-kiel.de,
WWW home page: http://www.numerik.uni-kiel.de/~\{bed,hes\}/

Abstract. This paper analyzes a new approach to the digital halftoning problem, which is to round a given continuous tone intensity image (represented by a \([0,1]\) valued matrix) into a binary one (represented by a \([0,1]\) matrix). Though being a rounding problem par excellence, the generic approach of randomized rounding is known to behave very badly for this problem. It produces extremely grainy outputs and has no chance to compete with algorithms like error diffusion or dithering.

In this paper we experiment with non-independent randomized rounding. That is, the brightness levels of the pixels are not rounded independently, but we enrich the random experiment with suitable dependencies that are designed to reduce the chance that grains are generated. Experimental results show that our roundings look much less grainy than those of independent randomized rounding, and only slightly more grainy than error diffusion. On the other hand, the latter algorithm (like all known deterministic algorithms) tends to produce unwanted structures, a problem that randomized algorithms like ours are unlikely to encounter. In a number of further ways our algorithm performs better than some and worse than other existing algorithms. In summary, it seems that our approach is a reasonable approach from the practical point of view.

1 Introduction

This paper is concerned with the digital halftoning problem. This is to convert a continuous tone image (every pixel can have an arbitrary color on the white-to-black scale) into a binary one (pixels are either black or white), which can be printed on a laser printer for example. If we represent the input image by a \([0,1]\)-valued \(m \times n\) matrix and the desired output by a \(0,1\) (binary) matrix of same dimension, we note that the digital halftoning problem is in fact a rounding problem. Naturally, our objective is to find a rounding such that the image represented by the binary matrix looks similar to our input image.

Unfortunately, human eye reception is hard to fit into mathematical formulas. There is currently no widely accepted way to measure similarity in terms of numbers, although a recent approach of Asano et al. [1] to model the digital
halftoning problem as discrepancy problem looks interesting. Nevertheless, the best way to compare different algorithms at this time seems to be by comparing their outputs through human eyes.

In this work we experiment with an algorithm recently suggested in [3]. This algorithm, though designed primarily to approximate the discrepancy problem suggested in [1], has several interesting aspects from the view-point of application. In particular, since it is a randomized algorithm, it does not produce unwanted textures.

Randomized approaches to the digital halftoning problem are not new, however, the solutions presented so far could hardly convince. Randomized rounding, i.e., comparing each entry with an independently chosen threshold from $[0, 1]$ and rounding up if and only if it is larger than the threshold, has been experimented with already in the seventies. The results, though, were poor. The images obtained were grainy and definitely inferior to those of algorithms like error diffusion or dithering.

Our algorithm, which is a variation of randomized rounding, overcomes this problem by using a different random experiment. Instead of doing the rounding independently, we use suitable dependencies that seem to reduce the number and size of grains.

After generating roundings of several both real-world and artificial images with existing algorithms and the new one, we conclude that neither algorithm can claim itself to be clearly superior to one of the others. In particular, ours is unbeaten in terms of avoiding unwanted patterns, whereas it has no chance to reach the error diffusion algorithm in terms of graininess.

We present examples showing the strengths and weaknesses of three classical algorithms, ours and independent randomized rounding. We included independent randomized rounding to demonstrate the effect of adding dependencies to the random experiment. Apart from that, independent randomized rounding continues to be not very interesting for digital halftoning applications.

We also show how these algorithms perform according to mathematical error measures, though we have to stress that all these measures only give a rough indication of what might be a good algorithm.

2 Survey of Known Algorithms

In this section we review three established algorithms as well as the approach of independent randomized rounding. Throughout the paper we denote the matrix representing the input image by $A \in [0, 1]^{m \times n}$ and the output matrix by $B \in \{0, 1\}^{m \times n}$.

2.1 Error Diffusion

A commonly used technique for the digital halftoning problem was introduced by Floyd and Steinberg [4] in 1975. They construct $B$ from $A$ by rounding the entries of $A$ one by one and distributing the error to nearby not yet computed entries of $A$. Their algorithm works as follows:
for $i := 1$ to $m$ do {
  for $j := 1$ to $n$ do {
    if $a_{i,j} < \frac{1}{2}$ then {
      $b_{i,j} := 0$
    } else {
      $b_{i,j} := 1$
    }
    $error := a_{i,j} - b_{i,j}$
    $a_{i,j+1} := a_{i,j+1} + error \times a$
    $a_{i+1,j} := a_{i+1,j} + error \times b$
    $a_{i+1,j+1} := a_{i+1,j+1} + error \times c$
  }
}

Floyd and Steinberg suggest to use the ratio $(a, b, c, d) = (\frac{7}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16})$ to
distribute the error.

In general, error diffusion shows extremely good results and maintains the
sharpness of the image. On the downside, it is an inherently serial method: The
value of $b_{mn}$ depends on all $mn$ matrix entries of $A$. This sequential structure is
believed to be the reason for ghosts produced by the error diffusion algorithm.

### 2.2 Ordered Dither

Another widely used technique is ordered dither [2, 6]. This is a thresholding
scheme where the matrix entries are rounded by comparing them with threshold
values depending on their coordinates. More precisely, to decide the rounding
a $d \times d$ dither matrix $C = c_{ij}$ with entries from $0$ to $d^2 - 1$ is used. An input
matrix $a_{ij}$ entry is rounded up ($b_{ij} := 1$) if and only if $a_{ij} \geq \frac{1}{d} c_{ij \mod d, j \mod d}$
holds. Different dither matrices have been experimented with, we use an $8 \times 8$
analogue of the one suggested by Bayer (Fig. 1).

Ordered dither is a fully parallel algorithm that has not been observed to
produce ghost, but tends to blur the images. Areas of constant brightness are
filled with a particular pattern depending on the dither matrix. These patterns
may attract unwanted attention.

\[
\begin{pmatrix}
0 & 32 & 8 & 40 & 2 & 34 & 10 & 42 \\
48 & 16 & 56 & 24 & 50 & 18 & 58 & 26 \\
12 & 44 & 4 & 36 & 14 & 46 & 6 & 38 \\
60 & 28 & 52 & 20 & 62 & 30 & 54 & 22 \\
3 & 35 & 11 & 43 & 1 & 33 & 9 & 41 \\
51 & 19 & 59 & 27 & 49 & 17 & 57 & 25 \\
15 & 47 & 7 & 39 & 13 & 45 & 5 & 37 \\
63 & 31 & 55 & 23 & 61 & 29 & 53 & 21 \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
34 & 48 & 40 & 32 & 29 & 15 & 23 & 31 \\
42 & 58 & 56 & 53 & 21 & 5 & 7 & 10 \\
50 & 62 & 61 & 45 & 13 & 1 & 2 & 18 \\
38 & 46 & 54 & 37 & 25 & 17 & 9 & 26 \\
28 & 14 & 22 & 30 & 35 & 49 & 41 & 38 \\
20 & 4 & 6 & 11 & 43 & 59 & 57 & 52 \\
12 & 0 & 3 & 19 & 51 & 63 & 60 & 44 \\
24 & 16 & 8 & 27 & 39 & 47 & 55 & 36 \\
\end{pmatrix}
\]

**Fig. 1.** An $8 \times 8$ dither matrix based on Bayer.

**Fig. 2.** The $8 \times 8$ class matrix suggested by Knuth.
2.3 Dot Diffusion

Knuth [5] suggested another solution for digital halftoning in 1987 called dot diffusion. This approach tries to combine the advantages of ordered dither and error diffusion: A parallel algorithm that maintains the sharpness of error diffusion without the risk of ghosts. Similar to ordered dither all entries are divided into $d^2$ classes according to their coordinates modulo $d$. Knuth suggests to use the following $8 \times 8$ class matrix.

The algorithm works as follows:

$$\begin{align*}
\text{for } c & := 0 \text{ to } d^2 \text{ do } \{ \\
\text{for all } (i,j) \text{ of class } c \text{ do } \{ \\
& \text{if } a_{i,j} < \frac{1}{2} \text{ then } b_{i,j} := 0 \text{ else } b_{i,j} := 1 \text{ } \
& \text{error} := a_{i,j} - b_{i,j} \\
& w := 0 \\
& \text{for all neighbors } (u,v) \text{ of } (i,j) \text{ do } \{ \\
& \text{if } \text{class}(u,v) > c \text{ then } w := w + \text{weight}(u-i,v-j) \\
& \text{if } w > 0 \text{ then do for all neighbors } (u,v) \text{ of } (i,j) \{ \\
& a_{u,v} := a_{u,v} + \frac{\text{error} \times \text{weight}(u-i,v-j)}{w} \\
& \} \} \} \}
\end{align*}$$

As can be seen above the pixel values are computed in order of their class. The error is distributed to all neighboring elements that have not been computed yet, i.e., belong to a higher class. Several weight functions can be applied. Knuth suggested using $\text{weight}(x,y) := 3 - x^2 - y^2$, which weights orthogonal neighbors twice as heavy as diagonal ones. The results of dot diffusion look grainy, but as Knuth points out, this algorithm is designed for high resolution devices where the effect of grains can be outnumbered by a higher resolution.

2.4 Randomized Rounding

All deterministic algorithms mentioned above carry the risk of producing unwanted patterns and structures. Due to their nature, randomized algorithms hardly can produce these kinds of problems. A simple approach that proved very efficient in many rounding problems is randomized rounding. In our case $B$ would be obtained from $A$ by independently rounding each entry of $A$ with probabilities $P(b_{i,j} := 1) = a_{i,j}$ and $P(b_{i,j} := 0) = 1 - a_{i,j}$.

Unfortunately, this algorithm is known to produce unfavorable results for a long time (cf. e. g. [2]). The images produces by randomized rounding look very grainy, and are not at all competitive to those produced by any of the algorithms described above. Both experiments and a moment’s thought show that randomized rounding has one nice feature though, namely that it can hardly produce any unwanted structures.

The aim of this paper is to provide a method that maintains randomness, but at the same time reduces the risk of grains in the image. As we will see, non-independent randomized rounding seems to have the desired characteristics.
3 Non-independent Randomized Rounding

The reason that randomized rounding produces grainy outputs seems to be the following: Assume to have an input image that is all light grey. Then most of the pixels in the rounded image should be white. For a fixed $3 \times 3$ submatrix, it is rather unlikely that all pixels are rounded to black ones. Looking at all $3 \times 3$ submatrices, however, it is rather likely that some of them will get completely dark in the rounding. The output image looks grainy. The idea of non-independent randomized rounding is to reduce the number and size of grains in probability.

Let us start with an easy example called joint randomized rounding upon which we will later build our algorithm.

We need some notation first. For a number $x$, we write $[x]$ for the set of all positive integers not exceeding $x$, $\lfloor x \rfloor$ is the largest integer not exceeding $x$, $\{x\}$ is the smallest integer not being less than $x$ and $\{x\} := x - \lfloor x \rfloor$ denotes the fractional part of $x$. For an arbitrary matrix $A$ we put $\Sigma A := \sum_{i,j} a_{i,j}$.

**Definition 1.** Let $a_1, a_2 \in [0, 1]$. We say that $(b_1, b_2)$ is a joint randomized rounding of $(a_1, a_2)$ if $b_1$ is a randomized rounding of $a_i$ for $i = 1, 2$ and the sum $(b_1 + b_2)$ is a randomized rounding of $(a_1 + a_2)$.

The idea of joint randomized rounding is obvious: If both $a_1$ and $a_2$ are small, say $\frac{1}{5}$, then it cannot happen that both are rounded up to one. On the other hand, each variable individually is a randomized rounding. We extend this way of rounding by adding further dependencies:

**Definition 2.** Let $A = (a_{11}, a_{12}, a_{21}, a_{22})$ be a $2 \times 2$ matrix. We say that $B = (b_{11}, b_{12}, b_{21}, b_{22})$ is a block randomized rounding of $A$ if

(i) each single entry of $B$ is a randomized rounding of the corresponding one of $A$, i. e., $P(b_{ij} = 1) = a_{ij}$ and $P(b_{ij} = 0) = 1 - a_{ij}$ for all $i, j \in [2]$.

(ii) each sum of two neighboring entries is a joint randomized rounding, i. e., for all $(i, j), (i', j') \in [2] \times [2]$ such that either $i \neq i'$ or $j \neq j'$ we have,

$$P(b_{ij} + b_{i'j'} = [a_{ij} + a_{i'j'}] + 1) = \{a_{ij} + a_{i'j'}\}$$

$$P(b_{ij} + b_{i'j'} = [a_{ij} + a_{i'j'}]) = 1 - \{a_{ij} + a_{i'j'}\}.$$

(iii) the box in total behaves like randomized rounded, i. e., we have

$$P(\Sigma B = |\Sigma A| + 1) = \{|\Sigma A|\}$$

$$P(\Sigma B = |\Sigma A|) = 1 - \{|\Sigma A|\}.$$

This is easily extended to arbitrary matrices:

**Definition 3.** Let $A \in [0, 1]^{m \times n}$ be the intensity matrix of our image. We say that $B$ is a non-independent randomized rounding of $A$, if it is computed by the following rounding scheme:

(i) for all $i \in [\frac{m}{2}], j \in [\frac{n}{2}], R := \{2i - 1, 2i\} \times \{2j - 1, 2j\}$, $B_{|R}$ is a block randomized rounding of $A_{|R}$ as in Definition 2,
(ii) if \( m \) is odd, then for all \( j \in \mathbb{N} \cup \{0\} \), \((b_{m,2j-1}, b_{m,2j})\) is a joint randomized rounding of \((a_{m,2j-1}, a_{m,2j})\) as in Definition 1,

(iii) if \( n \) is odd, then for all \( i \in \mathbb{N} \cup \{0\} \), \((b_{2i-1,n}, b_{2i,n})\) is a joint randomized rounding of \((a_{2i-1,n}, a_{2i,n})\) as in Definition 1,

(iv) if both \( m \) and \( n \) are odd, then \( b_{m,n} \) is a randomized rounding of \( a_{m,n} \).

All roundings in (i) to (iv) shall be independent.

From the definition it is clear that block randomized roundings should be less likely to contain grains than independent ones. The interesting point shown in [3] is that roundings as in Definition 2 always exist. We conclude:

**Theorem 1.** For any \( A \in [0,1]^{m \times n} \), block randomized roundings as in Definition 3 exist and can be generated in linear time.

## 4 Experimental Results

In the following section we shall demonstrate our experimental results. We applied the five algorithms described in Section 2 and 3 to several images. All image data used 1 byte per pixel resulting in an integer value between 0 and 255. We used two types of input data: Real-world images taken with a digital camera, and artificial images produced with a commercial imaging software. Naturally, the first type is more suitable to estimate how well the algorithm performs in real-world applications, whereas the second is better suited to demonstrate the particular strengths and weaknesses of an algorithm.

For reasons of space the images displayed in this paper are only small parts of the images we processed.\(^1\) These parts have a size of 160 x 160 pixels, and are displayed in 72 dpi. All printers nowadays can handle higher resolutions, of course, but the single pixels would be harder to recognize, and some unwanted effects like small white dots disappearing in a large black area would spoil the result.

### 4.1 Unwanted Structures and Textures

All known algorithms for the digital halftoning problem tend to produce some kind of structures or textures, which draw unwanted attention. Generally two kinds of textures can be observed. First there are regular patterns like snakes, crosses or labyrinths. In particular error diffusion and ordered dither algorithm tend to produce those, as can be seen in Fig. 3 and 4. These structures even can result in loss or change of image information, as can be seen in Fig. 8 and 9. In smooth transitions between dark and light areas rectangular shapes are generated, that have not been visible in the original image.\(^2\)

\(^1\) The original size images can be found at http://www.numerik.uni-kiel.de/~hes/NIRR.htm.

\(^2\) For very natural reasons, we are not able to display the original images in a printed paper. They can, however, be found on our web page.
**Fig. 3.** Error Diffusion (top left).

**Fig. 4.** Ordered Dither (top right).

**Fig. 5.** Dot Diffusion (middle left).

**Fig. 6.** Randomized Rounding (middle right).

**Fig. 7.** Non-independent Randomized Rounding (bottom).
The second form of unwanted structures are grains. Grains emerge, if in dark (respectively light) parts of the picture two or more white (respectively black) pixels touch each other and thus build a recognizable block. A single white pixel in an area of black pixels tends to look grey for the human eye since it has negligible area. A chequerboard of black and white pixels tends to look grey as well, but changing the color of a single black pixel into white makes the changed pixel shine white, as shown in Fig. 10. As the human eyes tend to react very sensitive to contrasts, these blocks of two or more pixels attract an unwanted amount of attention: The image looks grainy.

As explained in Section 3, randomized rounding is very vulnerable to this problem, which is why it is not used in practice for digital halftoning. On the other end we find error diffusion, which hardly produces any grains. It seems that algorithms that are good concerning graininess tend to produce unwanted structures and vice versa. In this sense, non-independent randomized rounding does not bad: Being by far less grainy than independent randomized rounding on the one hand, it is unlikely to produce unwanted structures on the other.
4.2 Performance

Certain applications as high resolution printing require parallel algorithms to be able to split processing. Inherently serial methods like error diffusion do not allow parallel computation. Ordered dither and dot diffusion allow parallel processing if implemented correctly.

Both randomized algorithms we used are suited for parallel computation as well. The rounding of a pixel, a 2 × 1 or a 2 × 2 submatrix can be independently done in constant time. The roundings can be assigned to different processors in an arbitrary manner.

4.3 Error Bounds

The visual reception of the human eye is very complex and therefore hard to model. It reacts less sensitive to absolute signal levels than to contrast. In certain cases, like the checkerboard, the human eye tends to blur white and black pixels into grey as long as they are in a regular pattern. On the other side patterns and textures like snakes, ghost, labyrinths or grains catch unwanted attention and distract the viewer from the initial picture information.

Having said this, there are still a number of attempts to catch the visual quality of halftoning by mathematical models. We decided to analyze four different error measures regarded in the literature with respect to two artificial and two real-world images. The errors are stated in Fig. 13 to 16. All measures given are scaled on a [0, 1] pixel value. For the randomized algorithms we ran the process 1000 times and took the average. Our analysis showed that the errors in are highly concentrated around the mean. Therefore we claim that they are reliable and that more tries would not improve the results.
<table>
<thead>
<tr>
<th>Method</th>
<th>Total</th>
<th>$3 \times 3$</th>
<th>$2 \times 2$</th>
<th>$1 \times 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Diffusion</td>
<td>0.000402</td>
<td>0.4355</td>
<td>0.3261</td>
<td>0.3204</td>
</tr>
<tr>
<td>Ordered Dither</td>
<td>0.000722</td>
<td>0.6072</td>
<td>0.3292</td>
<td>0.3220</td>
</tr>
<tr>
<td>Dot Diffusion</td>
<td>0.000232</td>
<td>0.6333</td>
<td>0.4142</td>
<td>0.3208</td>
</tr>
<tr>
<td>Randomized Rounding</td>
<td>0.000675</td>
<td>0.9255</td>
<td>0.6190</td>
<td>0.3213</td>
</tr>
<tr>
<td>Non-independent RR</td>
<td>0.000328</td>
<td>0.6382</td>
<td>0.4603</td>
<td>0.3213</td>
</tr>
</tbody>
</table>

**Fig. 13.** Errors in shadeX.

<table>
<thead>
<tr>
<th>Method</th>
<th>Total</th>
<th>$3 \times 3$</th>
<th>$2 \times 2$</th>
<th>$1 \times 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Diffusion</td>
<td>0.000290</td>
<td>0.4422</td>
<td>0.3334</td>
<td>0.3177</td>
</tr>
<tr>
<td>Ordered Dither</td>
<td>0.000654</td>
<td>0.6102</td>
<td>0.3283</td>
<td>0.3229</td>
</tr>
<tr>
<td>Dot Diffusion</td>
<td>0.000090</td>
<td>0.6025</td>
<td>0.4020</td>
<td>0.3176</td>
</tr>
<tr>
<td>Randomized Rounding</td>
<td>0.000658</td>
<td>0.9205</td>
<td>0.6156</td>
<td>0.3216</td>
</tr>
<tr>
<td>Non-independent RR</td>
<td>0.000342</td>
<td>0.6277</td>
<td>0.4539</td>
<td>0.3219</td>
</tr>
</tbody>
</table>

**Fig. 14.** Errors in abcd.

<table>
<thead>
<tr>
<th>Method</th>
<th>Total</th>
<th>$3 \times 3$</th>
<th>$2 \times 2$</th>
<th>$1 \times 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Diffusion</td>
<td>0.000133</td>
<td>0.4748</td>
<td>0.3640</td>
<td>0.3574</td>
</tr>
<tr>
<td>Ordered Dither</td>
<td>0.000251</td>
<td>0.7318</td>
<td>0.3576</td>
<td>0.3637</td>
</tr>
<tr>
<td>Dot Diffusion</td>
<td>0.000028</td>
<td>0.6755</td>
<td>0.4614</td>
<td>0.3567</td>
</tr>
<tr>
<td>Randomized Rounding</td>
<td>0.000466</td>
<td>1.0184</td>
<td>0.6859</td>
<td>0.3632</td>
</tr>
<tr>
<td>Non-independent RR</td>
<td>0.000221</td>
<td>0.6971</td>
<td>0.5121</td>
<td>0.3635</td>
</tr>
</tbody>
</table>

**Fig. 15.** Errors in car.

<table>
<thead>
<tr>
<th>Method</th>
<th>Total</th>
<th>$3 \times 3$</th>
<th>$2 \times 2$</th>
<th>$1 \times 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Diffusion</td>
<td>0.000252</td>
<td>0.4618</td>
<td>0.3144</td>
<td>0.3144</td>
</tr>
<tr>
<td>Ordered Dither</td>
<td>0.000463</td>
<td>0.6683</td>
<td>0.3323</td>
<td>0.3163</td>
</tr>
<tr>
<td>Dot Diffusion</td>
<td>0.000082</td>
<td>0.6745</td>
<td>0.4390</td>
<td>0.3146</td>
</tr>
<tr>
<td>Randomized Rounding</td>
<td>0.000442</td>
<td>0.9298</td>
<td>0.6237</td>
<td>0.3153</td>
</tr>
<tr>
<td>Non-independent RR</td>
<td>0.000190</td>
<td>0.6627</td>
<td>0.4808</td>
<td>0.3153</td>
</tr>
</tbody>
</table>

**Fig. 16.** Errors in room.
**Total Error:** The total error is the absolute difference of the average pixel value of the source and the output image:

\[
Total\ Error := \frac{1}{mn} \sum_{i \in [n], j \in [m]} |a_{ij} - b_{ij}|
\]

This indicates the capability of an algorithm to maintain the average brightness of the image.

1 × 1 Error: The 1 × 1 error is the average of the absolute errors in each pixel:

\[
1 \times 1\ Error := \frac{1}{mn} \sum_{i \in [n], j \in [m]} |a_{ij} - b_{ij}|
\]

2 × 2 Error: The next two error measures serve to compare close neighborhoods of pixels. Asano et al. introduced the error measure on 2 × 2 submatrices in [1] and reported that a low error with respect to this measure indicates a good halftoning.

\[
2 \times 2\ Error := \frac{1}{mn} \sum_{i \in [n], j \in [m]} \left| \sum_{k,l \in [2]} a_{i+k-1,j+l-1} - \sum_{k,l \in [2]} b_{i+k-1,j+l-1} \right|
\]

In general, ordered dither performs best regarding the 2 × 2 error. Non-independent randomized rounding has slightly worse error values than the deterministic algorithms, but with obvious improvements over randomized rounding.

3 × 3 Error: The 3 × 3 error is calculated in a similar manner as the 2 × 2 error, but with respect to 3 × 3 submatrices.

**Interpretation of results:** Concerning the total error and the error with respect to 1 × 1 boxes, we doubt that our results show any significant differences. Error diffusion, dot diffusion and non-independent randomized rounding seem to be superior in terms of the total error, but actually all algorithms even including randomized rounding do quite well: A total error of less than 0.001 is hardly to be spotted by the spectator.

The errors with respect to 2 × 2 and 3 × 3 boxes yield more interesting results. Whereas error diffusion and ordered dither have a significant advantage concerning 2 × 2 boxes (ordered dither even slightly better), error diffusion takes the lead concerning 3 × 3 boxes leaving ordered dither, dot diffusion and non-independent randomized rounding in a similar range behind. This might indicate that further research in the direction of [1] could yield new insight in what actually is a good error measure. It seems that the size of substructures regarded does make a difference.

Returning to the evaluation of the algorithms it seems that error diffusion in general performs best concerning the errors we computed, with the remaining algorithms except independent randomized rounding not far behind. Finally let us remark that the same statement holds for visual quality as does for the error measures: For each criterion and each algorithm there are images that support one this particular algorithm and there are others where the algorithm performs badly. The human reception seems to be by far to complex to be understood either with algorithms or through error measures.
5 Conclusions

In this paper we compare three well-established algorithms for the digital halftoning problem together with randomized rounding and a new approach of non-independent randomized rounding. We included randomized rounding mainly to estimate the effect of the dependencies included in the new approach, as otherwise independent randomized rounding is known to be unsuitable for digital halftoning.

Our experiments show several interesting results. First, the idea of reducing the graininess of non-independent randomized rounding by adding suitable dependencies worked out very well. The outputs of our algorithms are by far less grainy than those obtained by independent randomized rounding. However, they do not reach the error diffusion algorithm, that seems to be the unbeaten champion in terms of avoiding grains.

A second feature of our algorithms is that it produces no unwanted patterns. This seems to be caused by the randomness (the same holds for independent randomized rounding as well). All known deterministic algorithms tend to produce unwanted structures, in particular the error diffusion algorithm is vulnerable to this.

Concerning several mathematically defined error measures, our algorithm is in line with the three established approaches. Our algorithm has linear time complexity like all five algorithms regarded. Like ordered dither and dot diffusion it allows parallel implementations, which is an advantage over error diffusion.

From the view-point of application, we feel that the current state of affair, that no algorithm can claim itself superior or inferior to the others, persist if our one is added to the group of algorithms under consideration.

From the view-point of design of algorithms for the digital halftoning problem, our results suggest that the combination of randomized elements and structural ones is a promising approach.

References