アルゴリズム　87−11
(2002. 11. 8)

ディスクレパンシー基準によるディジタルハーフトーニング：自動評価手法と最適化手法

定兼邦彦、タッキシェビヒ　ナディア、徳山豪（東北大学）

概要　グレイイメージ画像を２値画像で近似することを(モノクロ画像の）ディジタルハーフトーニングと言う。本論文ではハーフトーニング画像の自動評価と最適基準によるハーフトーニングアルゴリズムの設計問題を取り扱う。まず、ディスクレパンシー基準による品質評価関数を考え、ハーフトーニング画像の自動評価のための高速アルゴリズムを与える。次に、このアルゴリズムを用いて評価関数についての比較実験を行い、更に、この実験において良好な性質を示した領域誤差関数に基づき、それを最小化するための実用的なアルゴリズムの設計を行った。

Discrepancy-Based Digital Halftoning: Automatic Evaluation and Optimization

Kunihiko Sadakane[1], Nadia Takki-Chebihi[1], Takeshi Tokuyama[1]
[1] GSIS Tohoku University, sada,nadia,tokuyama)@dais.is.tohoku.ac.jp

Abstract. Digital halftoning [6] is a problem of computing a binary image approximating an input gray image. We consider two problems on digital halftoning: mathematical evaluation of a halftoning image and design of optimization-based halftoning algorithms. First, we propose an efficient automatic evaluation system of halftoning images by using quality evaluation functions based on discrepancy measures. Then, we give experimental results on the evaluation system: they infer that the discrepancy corresponding to a regional error is a good evaluation measurement. Finally, we design heuristic algorithms in order to minimize this regional discrepancy measure.

1 Introduction

Digital halftoning is a well-known technique for computing a binary image approximating an input gray (or color) image , so that the binary image looks similar to the input one. The main motivation of halftoning process is to keep impression of gray-level variation of the original image in displaying it on binary devices such as laser printers and fax machines. Therefore there is a need to design a halftoning algorithm with a good visual quality.

So far, a large number of techniques have been presented in this field [6]. However those approaches don't propose reasonable criteria for evaluating how a halftoning image is similar to its original one and how its quality is good. Up to now, the most popular criterion to judge the quality is human vision system; this is inconvenient, since human's judgment depends on individual sense of beauty. Therefore, it is desired to establish an automatic evaluation system, and to handle the digital halftoning problem fully mathematically or algorithmically [1]. Consequently, the following questions should be answered in order to design a good digital halftoning system:

- How to give a reasonable mathematical definition of a good halftoning image ?
- How to evaluate quality of a halftoning image automatically?
- How can we compute a good halftoning image with respect to the above mentioned quality evaluation?

That is to say, we need to convert the digital halftoning problem as a rounding problem with a good optimization criterion. In order to evaluate quality of halftoning image automatically, we need to select a suitable quality measurement. Discrepancy is a hopeful candidate of mathematical measurement concept representing quality of halftoning images [1, 2, 11]. Intuitively, we consider a family $\mathcal{F}$ of regions in the pixel grid plane and a function f on $\mathcal{F}$ whose value at $R \in \mathcal{F}$ indicates the "*difference*" between an input picture and its output halftoning image within R.

In this paper, we give a prototype system for evaluating a halftoning output based on several discrepancy measures utilizing an efficient algorithm for computing discrepancy values. By applying the system to outputs of popular halftoning methods, we examine which discrepancy measures are suitable for quality evaluation.

Our experimental results using the evaluation system infer that the regional discrepancy on $k \times k$ rectangles is nicely compatible with human's judgment.

Thus, we next focus on devising a halftoning algorithm to output an image with small regional discrepancy. Unfortunately, computing the halftoning image minimizing the regional discrepancy for this family is NP-hard, and even its approximation is theoretically hard [3, 2].

Therefore, we provide several heuristic algorithms based on sequence rounding technique, for which we transform the pixel grid plane into a one-dimensional array by filling the grid plane with ordering curves and consider each curve as a sequence. We show experimental results of the proposed algorithms, in terms of halftoning outputs and their quality evaluations.

2 Matrix rounding problem and quality evaluation

A digital image is a two-dimensional array of small square regions known as pixels. In the case of a monochromatic (also known as a gray level) image, the brightness of each pixel is represented by a numeric value. We formulate the digital halftoning problem into a matrix rounding problem as follows: Let $A = (a_{ij})_{i,j=0,\ldots,n-1}$, be an input matrix of size $n \times n$, representing a gray level image, each pixel a_{ij} has a real value in the range [0,1], with 0 representing black, 1 representing white and values in between representing shades of gray. Let $B = (b_{ij})_{i,j=0,\ldots,n-1}$ be the output binary matrix of the same size, representing halftoning output image, so that each pixel b_{ij} uses only two values 0 and 1. In this formulation digital halftoning converts the real matrix A to a binary matrix B; thus B is obtained by *rounding* entries of the matrix A suitably.

2.1 One dimensional discrepancy

To specify a good quality of one-dimensional rounding, one needs to define an optimization criterion. Let $A = (a_0, a_1, \ldots, a_{n-1})$ be a sequence of real numbers in the range $[0,1]$ and let $B = (b_1, b_2, \ldots, b_n)$ be a binary sequence. Given an integral subinterval I of $[0, n-1]$, $|\sum_{i \in I}(b_i - a_i)|$ is the absolute difference between A and B within I. We fix a family $\mathcal{F}$ of intervals, we define a distance between A and B based on the l_∞-discrepancy by:

$$Dist^\infty(A,B) = \max_{I \in \mathcal{F}} \left| \sum_{i \in I} a_i - \sum_{i \in I} b_i \right|$$

We can compute I_{max} attaining the maximum absolute difference in $O(n)$ time by using a simple scanning algorithm. This is a classical topic in algorithm and programming, and the scanning algorithm is introduced as Kadane's scanning algorithm in the famous Programming Pearl article by Bentley [5].

2.2 Measurement for matrix rounding

Although $Dist^\infty(A, B)$ is a good distance between sequences, we need to measure the difference between two matrices. In a halftoning system, an output image B must look similar to the original image so that difference between A and B can be ignored by human's eye as negligible noise; in other words, it should be avoided that B has patterns meaningful for human's eye (e.g., line segments or curve segments) that are not in A. Such patterns are often called *unexpected patterns.*

For detecting an unexpected pattern that resembles to a horizontal (or vertical) line segment, we can use one-dimensional discrepancy considered in [9]. Indeed, we can detect such an unexpected pattern by applying the scanning algorithm for each column and row of the image.

Thus, our idea is to consider $k \times k$ subregions in the pixel grid, and judge that an output image is a good halftoning if it resembles to the input image in every such subregion according to a given optimization criterion. Therefore, for automatically judging whether a given output image B is good or not, we define a quality measurement function f defined on the set $\mathcal{F}_{k \times k}$ of all $k \times k$ subregions and indicating discrepancy between A and B, and then design an efficient algorithm to enumerate regions R with large f-values.

2.3 Discrepancy measures in a $k \times k$ region.

We define six objective functions based on discrepancy measures for evaluating errors within a region $R = I \times J \in \mathcal{F}_{k \times k}$, where I and J represents intervals of lengths k of row indices and column indices, respectively.

- *A*bsolute regional error: *ARE*

  $$ARE(R) = |\sum_{(i,j) \in R} (a_{i,j} - b_{i,j})|$$

  it gives the difference of total brightness between input and output within the region.

- *S*quared pixel error: *SPE*

  $$SPE(R) = \sum_{(i,j) \in R} (a_{i,j} - b_{i,j})^2$$

  determines the aggregated deviation between each pixel's gray value and its binary output.

- *S*quared row error: *SROE*

  $$SROE(R) = \sum_{i \in I} (\sum_{j \in J} a_{i,j} - b_{i,j})^2$$

measures the aggregated deviation between the input gray level of each row and its output in R. We expect that it can capture horizontal stripe textures. If its value is large, we can detect unexpected horizontal stripes in R, especially of unexpected wide horizontal line segments.

- *S*quared column error: *SCOE*

$$SCOE(R) = \sum_{j \in J}(\sum_{i \in I} a_{i,j} - b_{i,j})^2$$

measures the aggregated deviation within R between the input gray level of each column. We expect that it can capture vertical stripe textures.

- *S*quared (downward and upward) diagonal errors: *SDDE* and *SADE*

SDDE and *SADE* represent diagonal and off-diagonal features, respectively. We expect that they can capture diagonal/off-diagonal stripe textures. For $R = [1, k] \times [1, k]$, they are defined by

$$SDDE(R) = \sum_{-k+1 \leq y \leq k-1} (\sum_{i-j=y,(i,j) \in R} a_{i,j} - b_{i,j})^2$$

and

$$SADE(R) = \sum_{2 \leq y \leq 2k} (\sum_{i+j=y,(i,j) \in R} a_{i,j} - b_{i,j})^2$$

2.4 Scanning algorithm for computing evaluation functions

Given a halftoning image, among regions in $\mathcal{F}_{k \times k}$, we want to list up regions for which the above functions have large values. We want to investigate the relation between local quality of the worst $k \times k$ region and the global quality of the image for the human's eye, and also find out which of the six functions plays as the best indicator function for quality of an output.

Proposition 1 *We can compute ARE, SPE, SROE and SCOE for all the regions in* $\mathcal{F}_{k \times k}$ *in* $O(n^2)$ *time using* $O(n)$ *working space.*

Unfortunately, *SDDE* and *SADE* are more difficult to compute, since the diagonal segments in a $k \times k$ matrix have different lengths (from 1 to k), and we need to consume $O(k)$ time instead of $O(1)$ to update the information if we design a similar algorithm as above.

Proposition 2 *SDDE and SADE for them can be computed in* $O(kn^2)$ *time using* $O(n)$ *working space.*

We note that if we consider $k \times k$ rotated squares by 45 degree in the pixel grid instead of axis parallel squares, we can compute SDDE and SADE for them in $O(n^2)$ time although we have not done experiment on that version.

2.5 Evaluation of known halftoning algorithms

By using our scanning algorithm we evaluate the quality of some well-known halftoning.

2.5.1 Threshold rounding

A naive method to obtain a binary matrix B approximating A is to compare each real entry $a_{ij} \in [0, 1]$ to a fixed threshold value, let say 0.5, then determine the output pixel in binary. The method minimizes the squared pixel error *SPE*. However, its output (Figure 1) gives impression of the most awful quality since a gray region is transformed into totally white or totally black region.

2.5.2 Randomized rounding

This simple process rounds independently each visited pixel to 1 with probability a_{ij}, and to 0 with probability $(1 - a_{ij})$. Unfortunately, an output (Figure 2) of this method suffers from apparent grainess that makes the image blurred and it's hard for human to capture the outlines of objects in the image.

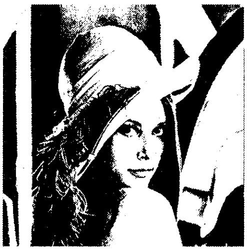


Figure 1: Threshold rounding

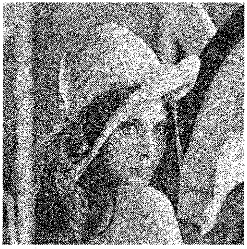


Figure 2: Randomized rounding

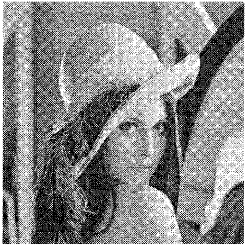


Figure 3: Ordered dither

2.5.3 Ordered dither

This approach partition the image into submatrices and round each submatrix by comparing its entries to a threshold matrix of same dimension. Its output (Figure 3) is much better for human's eye than those of threshold rounding and randomized rounding. A defect of this method is that it generates visible texture inherited from the dither matrix, and thus its output image gives an artificial impression.

2.5.4 One dimentional error diffusion

The one-dimensional error diffusion algorithm computes a binary sequence $B = (b_0, b_1, \dots, b_{n-1})$ from $A = (a_0, a_1, \dots, a_{n-1})$ greedily for $j = 0, 1, 2, \dots, n-1$ such that the inequality $\left|\sum_{i=0}^{j}(a_i - b_i)\right| \leq 0.5$ holds for each j. We process each row of the matrix by using the above algorithm. Its output (Figure 4) is suffering from vertical wave (or stripe) patterns and and vertical linear scars. Its due to the application of the same greedy algorithm for each row and thus adjacent output rows have similar periodical feature. This causes unexpected scars and vertical wave patterns.

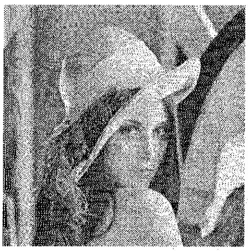


Figure 4: One dim. error diffusion

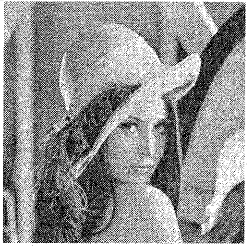


Figure 5: Two dim. error diffusion

2.5.5 Two dimensional error diffusion.

Two-dimensional error diffusion is a neighbor halftoning process given by Floyd and Steinberg [7]. The algorithm scans a matrix row-wise from its top row to the bottom one, and rounds entries greedily. The processing consists of threshold rounding for each pixel and an error computation, that is propagated to the neighboring pixels not yet processed with repect to fraction coefficients. Floyd-Steinberg suggested taking $(\alpha, \beta, \gamma, \delta)$=(7/16,3/16,5/16,1/16). As seen in Figure 5, two-dimensional error diffusion leads to higher halftoned image quality, and contouring artifacts are minimized. Although it still produces some vertical scars and artifacts looking like zebra stripes, its visual quality is best among the methods mentioned above.

2.6 Automatic evaluation of typical halftoning methods

We illustrate in section 5 Table 4 experimental results on the validity of our quality evaluation system by applying it to the outputs of the halftoning algorithms described in the previous subsection. It shows the maximum values over all regions in $\mathcal{F}_{5\times5}$ for each of our measurement functions, where "Row" and "Col" represents maximum of 1-dimensional discrepancies of rows and columns, respectively. SPE is not include in the table due to space limitation. It takes the value 7 over almost all described algorithms except for "Threshold rounding" its value is 6. Clearly, among these functions, ARE is most compatible with human's judgment (Table: 3).

The maximum values might be attained at few singular parts of the pictures; thus, one may worry that the maximum values given in Table 4 may fail to describe the visual feature of the outputs. We consider L_1 and L_2 measures which will be discussed in the next subsection for removing this anxiety.

We present the experimental results of our evaluation system in section 5. They are more detailed in the full version [10].

2.7 Variations of ARE

We have observed that ARE is very important for evaluating quality of halftoning measure. Thus, we examine ARE for different values of k, and also consider variants of ARE. We have considered the maximum value of $ARE(R)$ over all regions $R \in \mathcal{F}_{k\times k}$. This is called the L_∞ regional measure, and denoted by ARE_k^∞ from now on; that is,

$$ARE_k^\infty = \max_{R\in\mathcal{F}_{k\times k}} ARE(R).$$

Similarly, we define

$$ARE_k^1 = \sum_{R\in\mathcal{F}_{k\times k}} |ARE(R)|/|\mathcal{F}_{k\times k}|$$

and

$$ARE_k^2 = [\sum_{R\in\mathcal{F}_{k\times k}} |ARE(R)|^2/|\mathcal{F}_{k\times k}|]^{1/2}.$$

They basically correspond to L_1 and L_2 regional measures, respectively.

For the L_∞ ARE measure shown in Table 3, 2D diffusion method is best for the all ranges of $5 \leq k \leq 50$. However, for the L_1 and L_2 measurements given in Tables 1 and 2 respectively, phase transition occurs around $k = 30$. If the resolution of the picture becomes finer, ARE for a larger k becomes more important, since the resolution of human vision is fixed. Thus, for a future halftoning system, the authors guess that a method which minimizes ARE for a larger k will be more important.

3 Minimizing ARE by using global roundings

However, it is known to be NP-hard [3] to compute the binary matrix B minimizing ARE_k^∞ even for $k = 2$, and hence we need to design a heuristic or approximation method.

In this section, we modify the one-dimensional error diffusion algorithm by using the concept of *global rounding* given by Sadakane et. al. [9]. Consider a rounding $B = (b_0, b_1, \ldots, b_{n-1})$ of $A = (a_0, a_1, \ldots, a_{n-1})$. Given an integral subinterval I of $[0, n-1]$, $|B(I) - A(I)| = |\sum_{i \in I}(b_i - a_i)|$ is called the interval error of the rounding within I. For a given family $\mathcal{F}$ of intervals over a sequence, the rounding B is called *$\mathcal{F}$-global rounding* if $|A(I) - B(I)| < 1$ holds for every $I \in \mathcal{F}$.

A key fact is that there are exactly $n + 1$ global roundings (under a nondegeneracy condition) if we consider the set of all subintervals as the interval family. Moreover, we can design an efficient algorithm to enumerate all of them in $O(n^2)$ time [9]. Our strategy is, instead of 1-D error diffusion output, we choose a rounding from the set of all possible global roundings at each row, and try to break the synchronization, hoping to reduce scars and wave patterns consequently. By selecting a global rounding randomly in each row, we can reduce the generation of vertical stripe patterns as shown in the output image Fig. 6 from the one dimensional error diffusion output (Fig. 4).

Neglecting long intervals, we define I_t-global rounding where we consider the set of all subintervals of lengths at most $t < n$. The set of all I_k-global rounding forms the set of all source-sink paths of a DAG of size $O(tn)$, and hence we can select an I_t-global rounding in an uniformly random fashion; moreover, we can give side constraints or optimization conditions so that we can find the optimal I_t-global rounding by applying a shortest-path algorithm. The output images (for $n = 512$ and $t = 9$) are given in Figure 6, and we can see that they have few vertical scars. The computation time for all of n rows is $O(tn^2)$, which is less than $0.5t$ seconds in our experiment. Unfortunately, we observe wave patterns, and the waves are larger than that of 2-D error diffusion; thus, they give impression that the outputs are slightly rough.

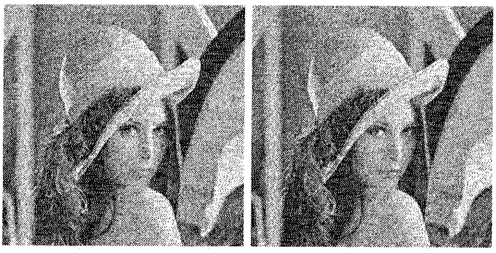


Figure 6: I_n-global rounding (left) and I_9 global rounding (right)

We evaluate quality of outputs of the method based on the I_t-global rounding for different values of t illustrated in [10]. In Table 4 we shows the results of evaluation functions for $t = 9$ and $t = n = 512$.

The values of discrepancy measures tend to decrease slightly if we increase t as interpreted in [10], although the visual difference (via human eyes judgment) between the quality of output images does not significantly depend on the choices of t if $t \geq 7$.

The values of ARE for $k = 5$ do not depend on t much; however, for a larger k, ARE_k depends on t heavily, as seen in the following tables 1, 2 and 3. We can observe that, for a larger k, the values are considerably smaller than the method given in the previous section if we take a large t. In precise for $k \geq 20$ ARE values are improved comparing with 2-D error diffusion method if t is large enough.

4 Rounding based on ordering curves

An alternative approach is to apply sequence rounding along an ordering curve (often called grid filling curve [12, 13]) that gives a sequential ordering of the pixels so that the consecutive pixels in the ordering is adjacent to each other in the grid. We combined the idea of ordering curve with I_t-global rounding method. The method takes advantage of characteristics of an ordering curve to reduce a two-dimensional problem to one-dimensional problem.

A *Hilbert curve* is a curve to fill pixel grid plane recursively. For our experiment we considered a non-recursive curve H_2 having 16 vertices each at the center of the sixteenth of the unit square. This is illustrated in Figure 7. We partition the pixel plane into horizontal bands of size $4 \times n$. We order the entries of each band following the H_2 curve. Doing so, we constitute a sequence of length $4n$ for each band. The combination of H_2 and I_n-global rounding gives a good halftoning result shown in Figure 8 having low discrepancy measures. This algorithm outputs an image which takes a relatively small value for every evaluation function. One weak point of this algorithm is its $O(n^3)$ time complexity, but it can be reduced by using I_t-global rounding for a smaller t; see Figure 9 for $t = 9$.

Tables 1, 2 and 3 show the evaluation quality of I_t-global rounding along Hilbert curve for $t = 9$ and $t = 4n$. We observe that if t increase the values of discrepancy measures tend to decrease [10]. For a $t \geq 20$ and $k \geq 30$ ARE^1_k and ARE^2_k are improved comparatively with 2-D error diffusion method.

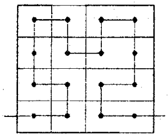


Figure 7: 4×4 Hilbert curve

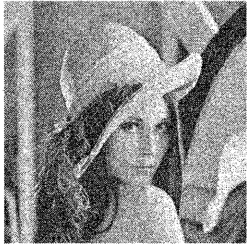


Figure 8: I_{4n}-global rounding along $H2$

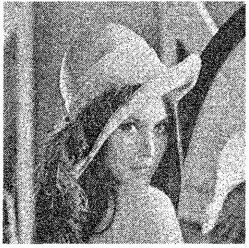


Figure 9: I_9-global rounding along $H2$

We also tried to order entries of an input matrix following a curve shaped as a union of diagonal zigzags of width 3. For notational convenience let DZ_3 denote this ordering curve. Figure 10 shows the partition of the picture into a set of sequences filling the pixel grid plane. Figure 13 is a halftoning image by the I_9-global rounding along the DZ_3 curve. Unfortunately, it suffers from downward diagonal stripe patterns.

4.1 Global rounding minimizing the sum of ARE

We have selected the sequence rounding from the set of all global roundings in a randomized fashion. However, we can control the selection by choosing the best one according to a given criterion. Indeed, we want to compute the I_t-global rounding along an ordering curve minimizing ARE^2_k.

Therefore we implemented an algorithm in which the $\mathcal{F}$-global rounding criterion and regional discrepancy criterion are simutaneously treated. Such an optimized rounding along an ordering curve substantially reduces the regional discrepancy as a solution for the matrix rounding problem.

The problem seems to be difficult theoretically, and we give a heuristic algorithm for the curve DZ_3 in order to test whether such kind of optimization is effective or not. Figure 11 illustrates the behavior of our heuristic algorithm.

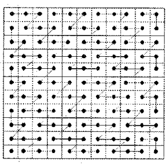
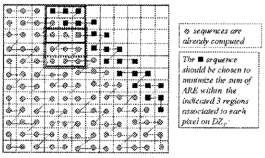


Figure 10: DZ_3 Ordering curve



◈ sequences are already computed
The ■ sequence should be chosen to minimize the sum of ARE within the indicated 3 regions associated to each pixel on DZ_3

Figure 11: Regions in $\mathcal{F}_{3\times3}$ associated with pixels of a sequence following DZ_3

Our heuristic is as follows: For simplicity, we assume that $n = 3m+2$ for a natural number m. Indeed, $512 = 3 \times 170 + 2$. For an integer $s \in [-m, m]$, let U_s be the tridiagonal region such that its central diagonal consists of the matrix entries $a_{i,j}$ satisfying that $j - i = 3s$.

Let $\mathcal{F}^i = \{R | R \in \mathcal{F}_{3\times3}, R \subset \cup^i_{j=-m} U_j\}$. Starting with $i = -m$, we compute the I_t global rounding of U_i along DZ_3 minimizing $\sum_{R \in \mathcal{F}_i} ARE(R)$ under the condition that the roundings of entries in $\cup^{i-1}_{j=0} U_j$ has been already computed.

The computation of the rounding of U_i can be done as follows: The set of I_t roundings of the sequence $DZ_3 \cap U_i$ (of length at most $3n$) can be represented as a set of shortest path of a directed acyclic graph G with $O(tn)$ nodes (see Figure 12 for an example). The

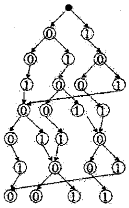


Figure 12: The $\mathcal{I}_3$-global rounding graph for an input sequence (0.4,0.4,...,0.4)

structure of G is given in Sadakane et. al. [9].

Given the graph G representing the set of I_t global roundings, we can compute the global rounding of $DZ_3 \cap U_i$ minimizing $\sum_{R \in \mathcal{F}_i} ARE(R)$ as follows: We keep track of square regions which are in $\mathcal{F}_i$ but not in $\mathcal{F}_{i-1}$. Let S be the set of such regions. For each node v of G, we give a value $Sum(v)$, which is the total sum of ARE values of regions in S whose all entries are rounded according to the labels of the path from the root to v; we do not add the ARE values of regions that contains an entry corresponding to the edges below the level of v. We sweep the leveled directed acyclic graph G from the root to leaves, and every time the rounding on a path newly determines ARE value of a square region, we update v. If two paths join at v, we take the parent so that the smaller value of $Sum(v)$ is attained. The suffix of length $3k$ of the two paths has the same suffix of length t, and since $t \geq 2k+1$, the suffix only influence to the non-determined ARE values. When the process is completed, we take the leaf with the smaller Sum value, and retrieve the path by using backtracking. It takes $O(tn)$ computation time for this process for an i. Thus, in total, we obtain the rounding of the $n \times n$ matrix in $O(tn^2)$ time.

The output halftoning image is given in Figure 14. The problem of the figure is that it has many small diagonal patterns, and we cannot say that optimization gives significant improvement of visual quality. Moreover, unfortunately, the gain of the optimization is not very large, we improved our objective function ARE_3^{∞} from 0.86 to 0.79. As shown in Tables 1, 2 and 3, ARE values are mostly improved from the output of I_9 global rounding along the curve, but not as good as I_n global rounding.

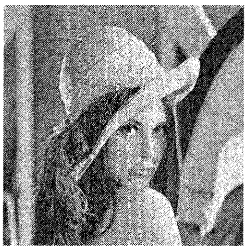


Figure 13: I_9-global rounding along DZ_3

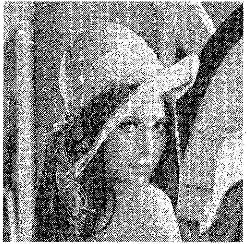


Figure 14: With optimization

4.2 Comparison of halftoning algorithms

The following three tables 1, 2 and 3 show ARE_k^p errors for $p = 1, 2, \infty$ and $k = 5, 10, 20, 30, 40, 50$. For larger k, I_n global rounding, I_{4n} global rounding along H_2, and I_n global rounding along DZ_3 perform well. This suggests that these method will work well for a finer pictures.

Table 4 shows the performance comparison on other measurement functions. We see that I_{4n} (or I_9) rounding along H_2 gives good scores for every quality evaluation function except the computation time. The CPU times for computing these image range from 0.6 to 107.1 seconds. Our algorithms are clearly slower than known methods, but not very slow if t is small.

Method	ARE_5^1	ARE_{10}^1	ARE_{20}^1	ARE_{30}^1	ARE_{50}^1
Threshold	7.34	27.21	96.80	196.80	459.11
Randomized	1.84	3.68	7.32	10.84	17.83
Dither	0.98	1.19	2.15	3.70	6.33
1D diffusion	0.74	1.12	1.69	2.15	2.90
2D diffusion	0.61	0.85	1.45	2.19	4.06
I_n rounding	0.67	0.99	1.43	1.75	2.26
I_9 rounding	0.72	1.27	2.89	5.59	14.47
I_{4n} along H_2	0.75	1.04	1.45	1.80	2.34
I_9 along H_2	0.80	1.31	3.15	6.47	17.75
I_n along DZ_3	0.91	1.30	1.84	2.28	2.93
I_9 along DZ_3	0.93	1.44	2.75	5.00	13.01
I_9-DZ_3 opt.	0.81	1.23	2.16	3.51	8.07

Table 1: Comparison of L_k^1 ARE values

Method	ARE_5^2	ARE_{10}^2	ARE_{20}^2	ARE_{30}^2	ARE_{50}^2
Threshold	7.87	29.66	108.39	224.13	534.91
Randomized	2.31	4.63	9.18	13.70	22.75
Dither	1.24	1.50	2.77	4.67	7.98
1D diffusion	0.97	1.48	2.26	2.89	3.89
2D diffusion	0.76	1.08	1.84	2.76	5.11
I_n rounding	0.86	1.25	1.81	2.20	2.84
I_9 rounding	0.92	1.64	3.78	7.06	17.01
I_{4n} along H_2	0.93	1.33	1.89	2.31	3.00
I_9 along H_2	0.99	1.69	4.02	7.75	19.30
I_n along DZ_3	1.14	1.63	2.32	2.87	3.68
I_9 along DZ_3	1.16	1.82	3.54	6.18	14.46
I_9-DZ_3 opt.	1.33	1.58	2.79	4.52	9.59

Table 2: Comparison of L_k^2 ARE values

5 Concluding Remarks

Based on our experiment, it is very important to select a suitable measurement function. We have succeeded to reduce ARE_k^p values (for $p = \infty, 1$, or 2) for $k > 20$ as illustrated in Figures 15 and 16, and thus we hope

Method	ARE_5^∞	ARE_{10}^∞	ARE_{20}^∞	ARE_{30}^∞	ARE_{50}^∞
Threshold	12.29	48.23	186.67	412.70	1127.96
Randomized	11.00	21.16	42.93	67.13	105.98
Dither	5.00	7.42	13.17	21.52	36.56
1D diffusion	4.71	8.78	15.45	19.49	25.54
2D diffusion	3.24	5.98	10.05	13.40	20.81
I_n rounding	4.36	6.16	8.13	10.02	11.97
I_9 rounding	4.42	9.16	20.16	41.97	86.65
I_{4n} along H_2	3.69	8.87	12.25	14.89	16.79
I_9 along H_2	3.98	9.01	21.94	42.61	86.76
I_n along DZ_3	4.50	6.87	10.22	13.83	17.38
I_9 along DZ_3	5.30	9.02	22.80	40.24	76.75
I_9-DZ_3 opt.	5.25	9.87	16.80	30.31	49.52

Table 3: Comparison of L_k^∞ ARE values

Method	Row	Col	$SROE$	$SCOE$	$SDDE$	$SADE$
Threshold	100	206	21	32	30	20
Randomized	37	38	31	28	23	25
Dither	107	134	14	15	25	24
1D diffusion	0.99	159	4	36	25	25
2D diffusion	16	68	13	33	22	22
I_n rounding	0.99	40	4	35	22	23
I_9 rounding	9	154	4	34	22	23
I_{4n} along H2	25	25	19	15	22	22
I_9 along H2	29	35	18	16	22	23
I_n along DZ_3	31	53	11	34	26	16
I_9 along DZ_3	27	43	12	35	26	16
I_9-DZ_3 opt.	42	30	11	32	26	16

Table 4: Values of other measurement functions

that our method using global rounding works well for halftoning a fine image; we will do experiments on such fine images in our future work.

ARE works well, but it is necessary to give a more effective measurement function to devise an evaluation system that simulate human's eye more precisely. It is necessary that ARE value should be low; however, optimizing ARE doesn't always give good looking output. Nevertheless, if we will find a more effective measurement function, we expect that we will be able to design a better halftoning method by using an algorithm for optimizing the measurement function. Instead of considering partitioning of the grid into ordering curves, we could use a covering of the grid by curves. That is, if two or more curves overlap at a pixel, we may choose the rounding of the pixel from the roundings given on the curves by majority and/or coin-flip. In our future work, we will implement our evaluation system by using other types of region families, for example, the 2-laminated family proposed by Asano et al.[2]. Also, we will try to detect unexpected scars and waves automatically and sensitively.

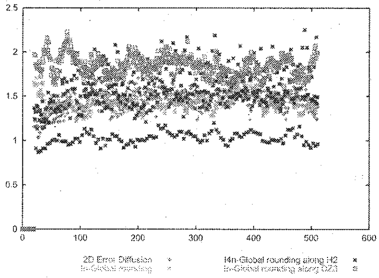


Figure 15: Distribution of ARE_{20}^1

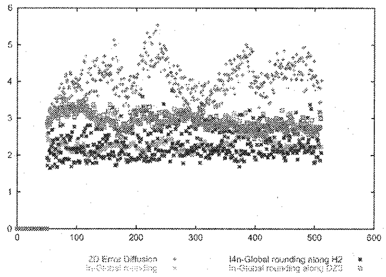


Figure 16: Distribution of ARE_{50}^1

References

[1] T. Asano: "Digital Halftoning: Challenges of Algorithm Engineers," *IEICE TRANS. Fundamentals*, vol. E02-D, No. 5, May 2002.

[2] T. Asano, N. Katoh, K. Obokata, T. Tokuyama: " Matrix Rounding under the L_p-Discrepancy Measure and Its Application to Digital Halftoning," *Proc. 13th ACM-SIAM Symp. on Discrete Algorithms*, pp.896–904, 2002.

[3] T. Asano, T. Matsui, and T. Tokuyama: "Optimal Roundings of Sequences and Matrices," *Nordic Journal of Computing*, pp. 241–256, vol. 7, No. 3, 2000.

[4] J. Beck and V. T. Sös: "Discrepancy Theory," in *Handbook of Combinatorics* Volume II, (ed. R.Graham, M. Grötschel, and L Lovász), Elsevier, 1995.

[5] J. Bentley: "Programming Pearls," *CACM*, vol. 27, pp. 865–871, 1984.

[6] Donald E. Knuth: "Digital Halftones by Dot Diffusion,"*ACM Transactions on Graphics*, Vol.6, pp. 245–273, 1987.

[7] R. W. Floyd and L. Steinberg: "An adaptive algorithm for spatial gray scale," *SID 75 Digest, Society for Information Display*, pp. 36–37, 1975.

[8] P. Raghavan and C. Thompson: "Randomized Rounding," *Combinatorica*, vol. 7, pp. 365–374, 1987.

[9] K. Sadakane, N. Takki-Chebihi, T. Tokuyama: "Combinatorics and Algorithms on Low-Discrepancy Roundings of a Real Sequences," *Proc. 28th ICALP, LNCS*, vol. 2076, pp. 166-177, 2001.

[10] K. Sadakane, N. Takki-Chebihi, T. Tokuyama: "Discrepancy-Based Digital Halftoning: Automatic Evaluation and Optimization,"*Interdisciplinary Information Sciences GSIS Tohoku University*. In preparation for pubilication.

[11] V. Rödl and P. Winkler: "Concerning a matrix approximation problem," *Crux Mathmaticorum*, pp. 76–79, 1990.

[12] L. Velho, J. Gomes: "Digital Halftoning with Space Filling Curves," *Proc. of SIGGRAPH '91*, pp.81-90, 1991.

[13] I.H Witten, M. Neal: "Using peano curves for bilevel display of continuous-tone images," *IEEE Computer Graphics and Applications*, pp.47-52, 1982.