LETTER

Digital Halftoning Algorithm Based on Random Space-Filling Curve*

Tetsuo ASANO[†], *Member*

SUMMARY This letter introduces a new digital halftoning technique based on error diffusion along a random space-filling curve. The purpose of introducing randomness is to erase regular patterns which tend to arise in an image area of uniform intensity. A simple algorithm for generating a random space-filling curve is proposed based on a random spanning tree and maze traversal. Some experimental results are also given.

key words: *digital halftoning, error diffusion, random space filling curve*

1. Introduction

Digital halftoning is a well-known technique in image processing to convert an image having several bits for brightness levels into a binary image consisting of black and white dots. Up to now, a large number of methods and algorithms for digital halftoning have been proposed (see e.g., [4]–[7], [9]). The error-diffusion method [5] among them is known to be good enough in quality of its output images and efficiency in ordinary serial machines. It scans each pixel in a raster manner and determines its output binary level by comparing its gray level with some average intensity level. The difference of its intensity level from the average level as the threshold is distributed as error over its neighboring pixels which have not been examined yet. Unfortunately, the regularity of error distribution is sometimes recognized as regular patterning in large uniform regions. Therefore, to have better quality we need to incorporate some randomness for error distribution.

An idea here is to use random space-filling curve instead of raster scan. Recently, it has been observed that error diffusion along some space-filling curves such as Peano curve [4] and Hilbert curve [11] sometimes achieve better quality compared to the traditional error diffusion based on a raster scan. In [10] parallel implementation of cluster-dot technique using a space-filling curve is proposed. One drawback of the methods comes from the fact that such space-filling curves are usually defined recursively on a square grid plane. Thus, there is some difficulty when they are applied to rectangular images. Another drawback is found in its quality of an output image due to its recursive structure. Since it is recursively defined, each quarter of an image is completely separated and their boundaries are often recognized in the resulting binary image.

Our idea is to use a space-filling curve based on random choices to avoid regular patterning. We have studied space-filling curves from different standpoints [1], [2], [8]. Some experimental results are included to show the effectiveness of those ideas.

2. Random Space Filling Curve

Given a lattice plane G, a space-filling curve on G is a curve which visits every lattice point on G exactly once. Since the shape of the curve itself is not important, it is sometimes represented as a permutation of lattice points of G. Many space-filling curves such as Hilbert and Peano curves are non-selfcrossing although this property is not a necessary condition for a curve to be space-filling. A number of space-filling curves are defined in addition to those famous curves (see for example, [8]).

The idea of using space-filling curves for digital halftoning is not new. Velho and Gomes [11] use space-filling curves for cluster-dot dithering. Zhang and Webber [10] give a parallel halftoning algorithm based on space-filling curves. Asano, Ranjan and Roos [2] formulate digital halftoning as a mathematical optimization problem and obtain an approximation algorithm based on space-filling curves. So, the digital halftoning techniques based on space-filling curves seem to be promising. However, one of their serious disadvantages is that there is some difficulty when the size of an input image is not a power of 2 since most of recursively defined space-filling curves such as Hilbert and Peano curves are defined for square lattice planes of sizes of powers of 2. One advantage of the random space-filling curve proposed in this paper is that it can be defined even for irregular-shaped lattice planes under some reasonable conditions. A more precise description for an irregular-shaped lattice plane will be given later.

Another disadvantage of using a recursive space-filling curve comes from the shapes of the curves. Suppose that we draw space-filling curves by connecting vertices which are consecutive in the order specified by the space-filling curves. Then, a straight gap is defined

Manuscript received July 6, 1998.

Manuscript revised October 15, 1998.

[†] The author is with the School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa-ken, 923–1292 Japan.

[*] A part of this paper was presented at IEEE International Conference on Image Processing, Lausanne, Switzerland, 1996.

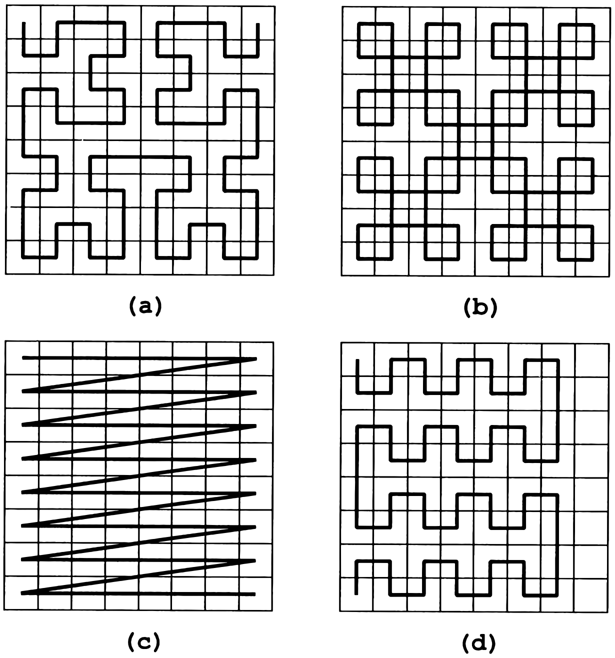

Fig. 1 Representative space-filling curves. (a) Hilbert curve, (b) Sierpinski curve, (c) Raster scan, and (d) Serpentine rack.

to be a horizontal or vertical line segment which does not intersect the space-filling curve. It should be noted that those gaps could be barriers against error propagation. Thus, long straight gaps are sometimes easily recognized in the resulting image. Figure 1 shows several representative space-filling curves which are defined recursively. It is easy to see that each space-filling curve contains long straight gaps. Especially, the serpentine rack has gaps which are as long as the side of the entire plane. On the other hand, since the random space-filling curve frequently changes its direction, it seldom contains long gaps. This is one of the advantages of the random space-filling curve.

Then, how can we generate random space-filling curves? First of all, can we guarantee the existence of such space-filling curves? And how can we incorporate randomness into space-filling curves?

In [3] a general scheme for defining a class of space-filling curves based on a grammar is presented. Although this suggests a way of constructing space-filling curves looking random, this kind of approaches bear some limit as far as they are based on a grammar. That is, such a space-filling curve exists only when the sizes of a rectangular grid are powers of some constants. On the other hand, this letter discusses a class of space-filling curves defined on a rectangular lattice plane of any even sizes. Consider a lattice plane looking like a checker board as shown in Fig. 2. If it consists of odd number of cells (small squares), the number of black cells is different from that of white cells. Thus, there is no space-filling curve starting and ending at cells of different colors. The reason is as follows. Colors of cells alternate in any space-filling curve. Therefore, the length of the space-filling curve must be even if the starting and ending cells have the same color. It contradicts to the fact that there are an odd number of cells in total.

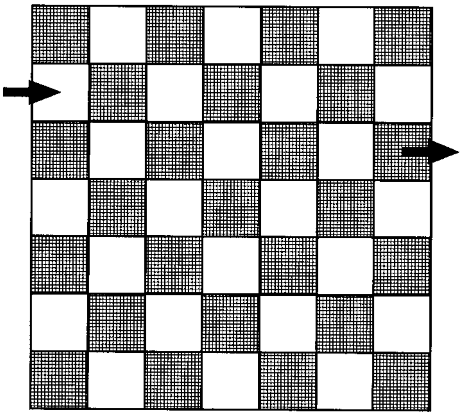

Fig. 2 Is there any space-filling curve with starting and target positions specified on a lattice graph consisting of an odd number of lattice points?

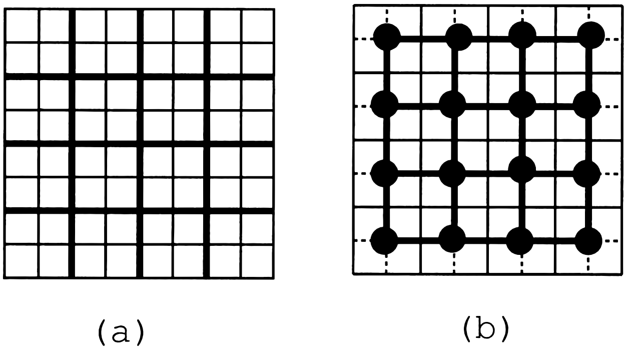

Fig. 3 Partitioning an image matrix into 2 × 2 small submatrices (to the left) and its associated lattice graph (to the right).

So, the questions above are not really trivial. In this letter we present a simple incremental algorithm for generating space-filling curves based on random choices at each step.

Let $A = (a_{ij})$, $i = 0, 1, \ldots, 2n-1, j = 0, 1, \ldots, 2m-1$ be a two-dimensional array with sides of even lengths. We first partition the entire array into 2 × 2 small arrays. We define a cell b_{ij} $(0 \le i \le n-1,\ 0 \le j \le m-1)$ to consist of elements $a_{2i,2j}, a_{2i+1,2j}, a_{2i,2j+1}, a_{2i+1,2j+1}$ (see Fig. 3 (a)).

Now, we can imagine a lattice graph whose vertices are those b_{ij}'s and two vertices are joined by an edge if they are horizontally or vertically adjacent to each other. In Fig. 3 (b) those vertices and edges are represented by black disks and solid bold lines, respectively.

Here we want to find a random spanning tree of the lattice graph. For the purpose we choose any vertex on the external boundary of the lattice graph as the root of the spanning tree. Then, we implement a depth-first search on the lattice graph starting at the root. To incorporate randomness, at each step during the search we choose the next adjacent vertex randomly if two or three unvisited vertices are adjacent to the current vertex. Formally, the algorithm is described as follows:

Generating a Random Spanning Tree

```
$G$ := a lattice graph associated with an input image;
$V$ := its vertex set;
Choose a vertex $u$ on the external boundary of $G$;
Let $T$ be a tree which has one node $u$ as a root;
Call a recursive procedure rdfs($u$);
procedure rdfs($u$){
    Mark $u$;
    $C(u)$ := a set of unmarked vertices adjacent to $u$;
    while($C(u)$ is not empty){
        Remove a vertex $v$ randomly out of $C(u)$;
        Add the edge $(u, v)$ to $T$;
        Recursively call rdfs($v$);
    }
}
```

An example of a random spanning tree is shown in Fig. 4 (a).

Now it is easy to see that the resulting tree serves as a connected wall which defines a maze on the rectangular grid plane. Thus, if we follow the wall while keeping one hand touching the wall, we can traverse the entire image. See Fig. 4 (b) in which the traverse associated with the spanning tree is depicted by dotted lines.

The above definition of a random space-filling curve on a rectangular lattice plane can be generalized to an irregular-shaped plane. It may have holes. The condition for a lattice region to satisfy is the following; It is a collection of 2×2 small lattice regions. We say that two such small regions are fully adjacent to each other if they share their horizontal or vertical sides of length two. Then, those small lattices must form a single connected component.

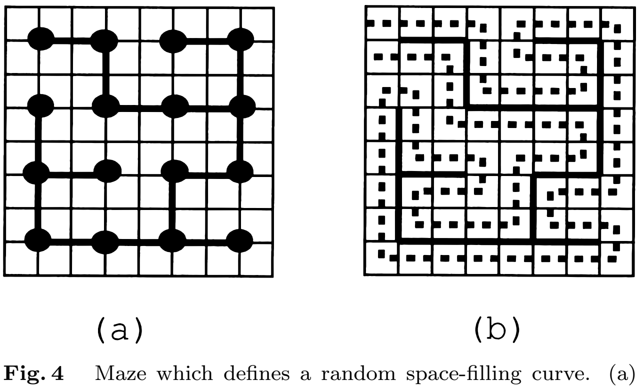

Fig. 4 Maze which defines a random space-filling curve. (a) Spanning tree. (b) Traverse along the tree.

3. Error Diffusion along RSFC

Once a random space-filling curve (or RSFC for short) is fixed, we can implement error diffusion along the curve. More concretely, we scan pixels in the order defined by the curve. For each such pixel, we compare its intensity level with some predetermined threshold level to determine its binary level. Then, the rounding error is diffused to adjacent pixels which have not been scanned yet. In the Floyd-Steinberg algorithm [5] based on a raster scan the error is distributed to adjacent pixels based on the following matrix:

```
0/16  0/16  0/16
0/16        7/16
3/16  5/16  1/16
```

Since the rounding error is propagated to those adjacent pixels whose output levels have not been determined, we have 0 weights for those four pixels in the upper and left parts. 1/16 is the normalizing factor of the weights.

In our case based on a random space-filling curve a set of pixels to which the error is propagated is not fixed. So, each entry could have a non-zero value. We have tested several weight matrices against standard test patterns in the SIDBA library, and chose the following weight matrix based on experimental evaluation.

```
1  5  3
7     7
3  5  1
```

At each pixel p we first find those pixels adjacent to p that have not been scanned yet, and then compute the sum of the weights associated with those pixels to determine the normalizing factor at p. Note that the algorithm behaves exactly in the same fashion if space filling-curve is raster scan. According to our experience on experiments, the error propagating ratio defined above is somewhat too large. So, in our program we used 30% larger value of the total weights as the normalizing factor. In other words we propagated only about 76% of the rounding error generated at each pixel to its adjacent pixels. This prohibits too long error propagation since the influence decreases exponentially in the distance.

Figures 5 and 6 show output images due to ordinary error diffusion method based on raster scan, and error diffusion along a random space-filling curve defined above. Regular patternings are recognized in Fig. 5 especially in its upper parts while such patterns are not included in Fig. 6. Moreover, details are somewhat more clearer in our result in Fig. 6. See for example the boundary of the sleeve of the right hand.

4. Concluding Remarks

In this paper we have proposed a new algorithm for digital halftoning based on a random space-filling curve induced by a random spanning tree. Using space-filling curves for halftoning is not new, but we experimentally showed that randomness is important to use space-filling curves as a guide for halftoning. Space filling curves have been extensively investigated for various applications in mind [2], [8], [10], [11]. To the author's best knowledge, this is the first efficient algorithm for generating random space-filling curve.

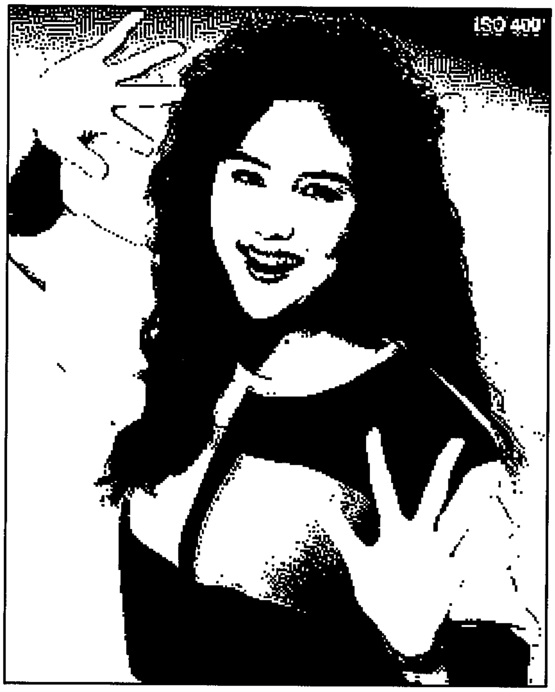

Fig. 5 Output image by ordinary error diffusion method.

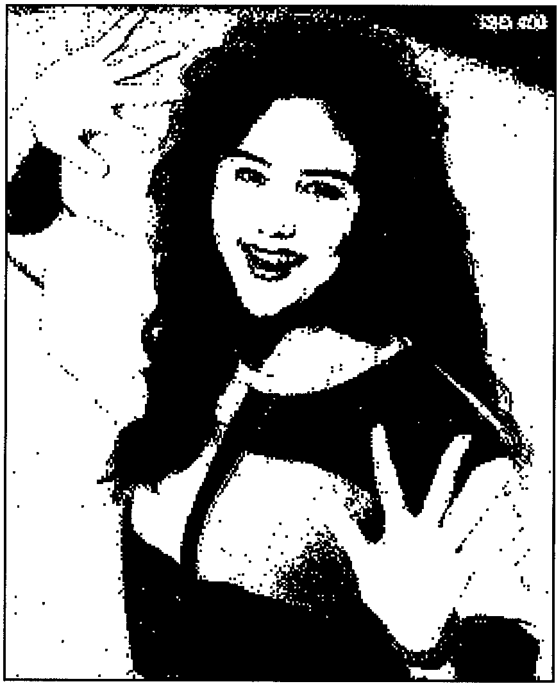

Fig. 6 Output image by error diffusion along a random space-filling curve.

Acknowledgement

The basic idea in this paper was devised by Takashi Hashimoto, a former student of Osaka Electro-Communication University. The author would like to thank him for his great contribution to this work. This work was partly supported by Grant in Aid for Scientific Research of the Ministry of Education, Science and Cultures of Japan.

References

[1] T. Asano, N. Katoh, H. Tamaki, and T. Tokuyama, "Convertibility among grid filling curves," in Proc. International Conf. on Algorithms and Computation, pp.307–316, Taejon, Korea, Dec. 1998.
[2] T. Asano, D. Ranjan, and T. Roos, "Digital halftoning algorithms based on optimization criteria and their experimental evaluation," IEICE Trans., vol.E79-A, no.4, pp.524–532, April 1996.
[3] T. Asano, D. Ranjan, T. Roos, E. Welzl, and P. Widmayer, "Space-filling curves and their use in the design of geometric data structure," Theoretical Computer Science, vol.181, pp.3–15, 1997.
[4] A.J. Cole, "Halftoning without dither or edge enhancement," The Visual Computer, vol.7, pp.232–246, 1991.
[5] R.W. Floyd and L. Steinberg, "An adaptive algorithm for spatial gray scale," SID 75 Digest, Society for Information Display, pp.36–37, 1975.
[6] R. Geist, R. Reynolds, and D. Suggs, "A Markovian framework for digital halftoning," ACM Trans. Graphics, vol.12, no.2, pp.136–159, 1993.
[7] D.E. Knuth, "Digital halftones by dot diffusion," ACM Trans. Graphics, vol.6, no.4, pp.245–273, 1987.
[8] T. Roos, T. Asano, D. Ranjan, E. Welzl, and P. Widmayer, "Space filling curves and their use in the design of geometric data structures," Theoretical Computer Science, vol.181, pp.3–15, 1997.
[9] R. Ulichney, "Digital halftoning," The MIT Press, Cambridge, 1988.
[10] Y. Zhang and R.E. Webber, "Space diffusion: An improved parallel halftoning technique using space-filling curves," Proc. of SIGGRAPH '93, pp.305–312, 1993.
[11] L. Velho and de M. Gomes, "Digital halftoning with space filling curves," Proc. of SIGGRAPH '91, pp.81–90, 1991.