JAIST Repository
https://dspace.jaist.ac.jp/

Title	食い違い度理論に基づいたデジタルハーフトーニングにおける点配置問題に関する研究
Author(s)	大島，穣
Citation	
Issue Date	2004-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1784
Rights	
Description	Supervisor:浅野 哲夫，情報科学研究科，修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

食い違い度理論に基づいたデジタルハーフトーニングにおける点配置問題に関する研究

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

大島 穣

2004 年 3 月

修 士 論 文

食い違い度理論に基づいたデジタルハーフトーニングにおける点配置問題に関する研究

指導教官　浅野哲夫 教授

審査委員主査　浅野哲夫 教授
審査委員　平石邦彦 教授
審査委員　金子峰雄 教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

210011 大島 穣

提出年月 2004 年 2 月

Copyright © 2004 by Ohshima Minoru

概 要

デジタルハーフトーニングとは連続階調をもつ画像を 2 値の画像に変換する技術である．これは印刷機が入力画像にできるだけ類似している画像を出力するための重要な技術である．本研究では，デジタルハーフトーニング技術と密接に関係している単位正方形内にある一定数の点を一様に分布させる点配置問題を正方格子のような離散平面上での食い違い度理論 $(discrepancy\ theory)$ に基づいて考える．従来，計算幾何学の分野で研究されてきた食い違い度理論に基づいた正方格子での点配置問題は漸近的な解析以外にはなされておらず，計算機で実装可能な問題のサイズにおいては必ずしも最適な点配置を求めているとはいえなかった．本研究では，デジタルハーフトーニングなど現実的に使用可能な小さな範囲での食い違い度理論に基づいた離散平面上の点配置問題の漸近解析について考察する．また，現実的に最適な点配置を求めるアルゴリズムを提案し，デジタルハーフトーニング技術へ応用する．

目 次

第 1 章　序論 **1**
1.1　本研究の背景と目的 1
1.2　本論文の構成 2

第 2 章　食い違い度理論 **3**
2.1　食い違い度の定義 3
2.2　他の点配置問題との比較 4

第 3 章　食い違い度理論に基づいた既存の手法 **6**
3.1　離散平面上での食い違い度の定義 6
3.2　ランダム丸めを用いた手法 6
3.2.1　独立なランダム丸め 7
3.2.2　同時ランダム丸め 7
3.2.3　連続ブロックランダム丸め 8
3.3　既存の手法による実験結果 9

第 4 章　提案手法 **18**
4.1　提案手法１ 18
4.2　提案手法２ 19

第 5 章　操作順序：スペースフィリング曲線 **20**
5.1　単純なスペースフィリング曲線から変形させる方法 21
5.1.1　WFCC を用いた手法 21
5.2　全域木を利用した方法 23
5.2.1　RSFC を用いた手法 25
5.2.2　集合ユニオンファインド木を用いた手法 27
5.3　スペースフィリング曲線の出力結果 29

第 6 章　実験結果 **33**
6.1　準備 33
6.2　出力結果と考察 34
6.2.1　入力画像：グラデーション画像 34
6.2.2　入力画像：一定の明るさの画像 36

6.2.3　入力画像：風景画像 38

第 7 章　まとめ **41**

7.1　結論 41

7.2　今後の課題 41

図 目 次

2.1 軸並行な長方形の領域 R 4
2.2 食い違い度 $O(\sqrt{n})$ の点集合 5
2.3 食い違い度 $O(\log n)$ の点集合：Van der Corput 集合 5

3.1 グラデーション画像を入力とした既存手法による出力結果 10
3.2 一定の明るさの画像を入力とした既存手法による出力結果 11
3.3 風景画像を入力とした出力結果：原画像 12
3.4 風景画像を入力とした出力結果：オーダードディザ法 13
3.5 風景画像を入力とした出力結果：エラーディフュージョン法 14
3.6 風景画像を入力とした出力結果：独立なランダム丸め法 15
3.7 風景画像を入力とした出力結果：同時ランダム丸め法 16
3.8 風景画像を入力とした出力結果：連続ブロックランダム丸め 17

5.1 例：スペースフィリング曲線 20
5.2 WFCC のアルゴリズム：Step1 22
5.3 WFCC のアルゴリズム：Step2 22
5.4 WFCC のアルゴリズム：Step4,Step5 23
5.5 格子平面上における点集合の表現方法 24
5.6 全域木を用いたスペースフィリング曲線 24
5.7 辺の延長可能な方向 25
5.8 RSFC で生成した全域木 (頂点数：64×64) 26
5.9 集合族 {{1,2,4,6,9},{3,5,7},{8}} を表現するユニオンファインド木 27
5.10 集合ユニオンファインド木生成の高速化 28
5.11 集合ユニオンファインド木 (頂点数：64×64) 29
5.12 WFCC を用いて生成したスペースフィリング曲線 30
5.13 RSFC を用いて生成したスペースフィリング曲線 31
5.14 集合ユニオンファインド木を用いて生成したスペースフィリング曲線 . . . 32

6.1 グラデーション画像を入力とした出力結果 35
6.2 一定の明るさ (輝度値１６) の画像を入力とした出力結果 37
6.3 風景画像：WFCC を用いた手法による出力画像 38
6.4 風景画像：RSFC を用いた手法による出力画像 39

6.5　風景画像：ユニオンファインド木を用いた手法による出力画像 40

第1章　序論

1.1　本研究の背景と目的

現在，爆発的なパーソナルコンピュータの普及とネットワークエリアの拡大により，最新の情報を新聞などからだけでなくインターネットから簡単に手に入るようになってきている．それでもなお，インターネットから得た情報やコンピュータで作成した文章，デジタルカメラなどで得たデジタル画像などを保存性の高い紙に記録するプリンタは今でも重要なインターフェイスの一つである．また，近年ではデジタルカメラの普及が急速に広がっていると同時にデジタルカメラ自体の高性能化も著しく，高画質のデジタル画像を扱う機会が増えてきている．そのため，デジタル画像の質をなるべく劣化することなくプリント出力させるような，インクジェットプリンタなどの高画質化も重要な問題である．

これまでのインクジェットプリンタの高画質化はハードウェア面での飛躍的な向上が大きい．機械的に，より精密に細かく点を置くことによって高い解像度で表現できるようになり，より高画質な画像が出力可能となった．しかしながらこれ以上の高解像度はあまり必要とされることはないという段階まで向上してきており，プリンタのさらなる高画質化は画像処理技術によるソフトウェア面での向上が必要である．

インクジェットプリンタの基本原理は色の 3 原色と黒の 4 色 (CMYK) のインクを基に紙にプリントする方法である．つまり，各色のインクをその点に置くか置かないかの組み合わせによって様々な色を表現することになる．最近ではさらにインクの色の数を増やすことによってさらなる高画質化を図っているが，それでもインクの色の数には限界があり，フルカラーでプリントすることはできない．そのため，限られたインクをうまく組み合わせて，連続階調をもつ入力画像を 2 値の画像に変換するデジタルハーフトーニング技術は，印刷機やプリンターが入力画像にできるだけ近い画像を出力するための重要な技術である．つまり，デジタルハーフトーニングが必要となるのは多階調のデジタル画像にある中間色を 2 値の階調で表現するときである．これまでに単純 2 値化法やオーダードディザ法，エラーディフュージョン法など様々なデジタルハーフトーニングのアルゴリズムが提案されており，現在のプリンタ技術に利用されている．しかし，どのアルゴリズムにおいても出力画像に特有の模様が現れたり輪郭部分がぼやけてしまうといった欠点を持つ．このように入力画像にできるだけ類似した画像を出力するにはまだいくつかの問題が残っている．

これらの問題を考えるうえで，2 値の画像は「点を打つ」または「打たない」のどちらかであるので，デジタルハーフトーニングを計算幾何学の分野の一つである点配置問題

(単位正方形内に一定量の点を一様に分布させる問題) に置き換えることが可能である．また，これまでのデジタルハーフトーニング手法のほとんどで見られる特有の模様が表れる問題を解決するには，点配置の不規則性と一様分布性に着目して研究されてきた食い違い度理論 (*discrepancy theory*) が非常に有用である．近年，Doerr らがランダム丸めに基づいたいくつかのアルゴリズムを開発し，またそのアルゴリズムを食い違い度理論に基づいた理論的解析も行っている [2][3]．Doerr らのアルゴリズムは，各々の処理において，処理領域内の誤差値を常にある一定値以下に保ちながら，その領域内の画素を確率的に丸める手法である．しかしながら，Doerr らのアルゴリズムは各々の処理を施す度に食い違い度が蓄積されるために，入力画像のサイズに比例して全体の食い違い度が大きくなってしまうといった欠点を持っている．

本研究では，Doerr らのアルゴリズムに誤差伝播とランダムな走査順序を取り入れ，特有の模様が表れない，かつ入力画像にできるだけ類似している 2 値画像を出力するアルゴリズムを開発する．また，それらのアルゴリズムを実装し，見た目などの評価や既存手法に対する比較を行った．

1.2 本論文の構成

本論文の構成は以下のようになる．第 2 章では，食い違い度理論について述べる．第 3 章では，食い違い度理論に基づいた既存の手法を挙げ，それらの出力画像と代表的なデジタルハーフトーニング手法の出力画像も含めた解説および評価を述べる．第 4 章では，第 3 章で述べた食い違い度理論に基づいた既存の手法を改良した 2 つの提案手法を述べる．第 5 章では，提案手法で用いられるスペースフィリング曲線について述べる．第 6 章では，提案手法を実装し実験で得られた結果を示すとともに，それぞれの実験結果の評価を述べる．最後に第 7 章では，まとめと今後の研究課題について述べる．

第2章　食い違い度理論

食い違い度理論 ($discrepancy\ theory$) は単位正方形内にある一定量の点を一様に分布させる点配置問題を考える上で，非常に有用な理論である．この理論には次のような代表的な問題がある．

- 単位正方形内に，ある一定量の点を最も一様に分布させる方法は何か．
- そのような任意の分布の不規則度 ($irregularity$) はどのくらいの大きさか．

これらの問いを定式化するには，与えられた分布の不規則度を数量化しなければならない．つまり，食い違い度理論とは，不規則度の数量化に基づいた点集合の定量的評価に関する理論である．このような問題は，最初，1930 年代後半に取り組まれた．2 次元平面状の基本的な食い違い度理論の満足する解は 1960 年代後半に得られたが，より高次元の問題は今日に至るまで解かれていない．また，従来からこの理論は連続平面上での議論が主であり，離散平面上での議論は漸近的な解析以外にはなされていない．

本章では，連続平面上における食い違い度の定義とさまざまな点配置に対する評価について述べる．

2.1　食い違い度の定義

食い違い度の定義にはさまざまなものがあるが，ここでは連続平面上における軸平行な長方形に対する点集合の食い違い度の定義を紹介する．

単位正方形 $[0,1]^2$ を U とし，U 内に任意に分散された n 個の点の集合を P とする．このとき全ての軸平行な長方形 $[a_1,b_1)\times[a_2,b_2)\subseteq U$ の集合を考え，これを $\mathcal{R}$ とする．$R\in\mathcal{R}$ に対して，長方形 R 内に含まれる点集合 P の点の数と長方形 R 内に含まれる点の期待値との食い違い度 $D(P,R)$ は，

$$D(P,R)=|n\cdot area(R)-|P\cap R||$$

で定義される．ただし，

$$area(R):R\cap U\ \text{の面積}$$

である (図 2.1)．つまり，実際に R に含まれている点の数が近似的にその期待値に近づけ

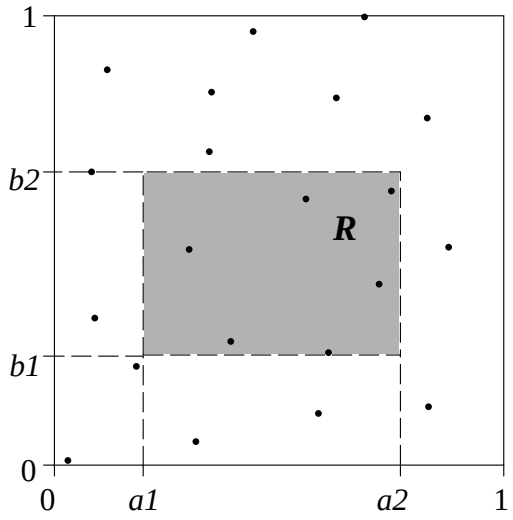


図 2.1: 軸並行な長方形の領域 R

ば食い違い度の値が小さくなり，点配置が一様に分布しているといえる．

このとき，軸平行な長方形に対する点集合 P の食い違い度とは，全ての可能な軸並行な長方形 R に対する食い違い度の最小上界のことであり，

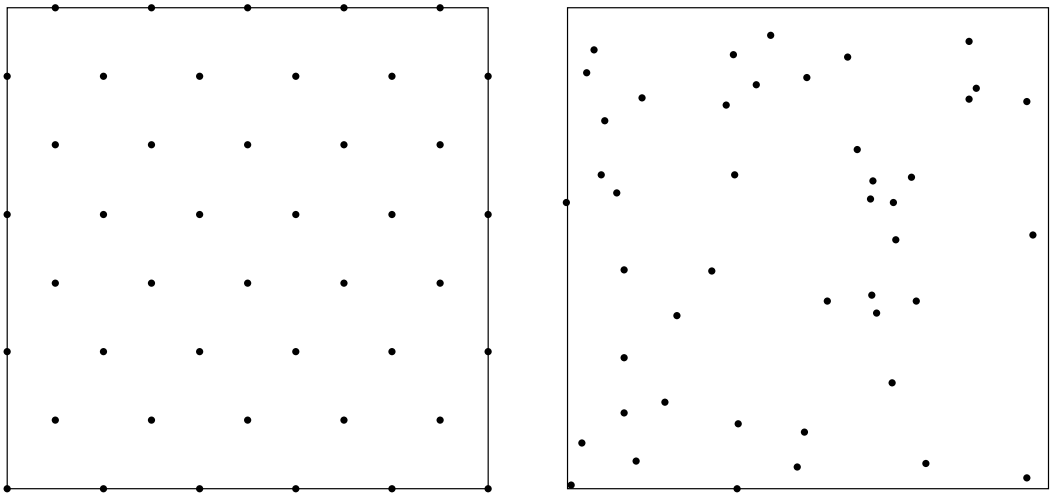
$$D(P, \mathcal{R}) = \sup_{R \in \mathcal{R}} D(P, R)$$

と定義される．

2.2 他の点配置問題との比較

他の点配置問題として，ある決められた空間内に定数個の点を任意の 2 点間距離の最小値が最大となるように配置する問題 (2 点間最小距離最大化問題) がある．しかし，この問題による一様分布性の評価基準は食い違い度理論による一様分布性の評価基準とは極めて異なる．たとえば，図 2.2(a) に示しているような三角格子状の点の集合は単位正方形内で 2 点間最小距離が最大である点集合であるが，食い違い度理論からの視点では極めて悪い点集合である．点の個数を n としてこの点集合を食い違い度の定義に当てはめると，およそ $\sqrt{n}$ の食い違い度がある．また，単に乱数を用いて配置された点の集合 (図 2.2(b)) の食い違い度は $O(\sqrt{n})$ であることが知られている．

食い違い度理論では一般に食い違い度が $O(\log n)$ 以下のとき，良い点集合といわれる．食い違い度が $O(\log n)$ である代表的な点配置の一つとして，Van der Corput 集合 [1] と呼ばれるものがある．図 2.3 に一例を示す．



(a) 三角格子状の点集合　　(b) 乱数を用いて配置された点集合

図 2.2: 食い違い度 $O(\sqrt{n})$ の点集合

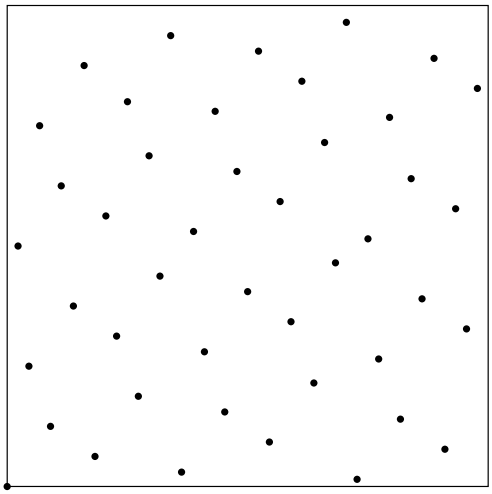


図 2.3: 食い違い度 $O(\log n)$ の点集合：Van der Corput 集合

第3章　食い違い度理論に基づいた既存の手法

前章では連続平面上での食い違い度理論について述べた．しかし，デジタルハーフトーニングなどに応用するには，画像の各ピクセルに対応できるような正方格子状の離散平面上で考えなければならない．そこで，この章では正方格子のような離散平面上での食い違い度を定義し，その定義に基づいた既存の手法を幾つか紹介する．

3.1　離散平面上での食い違い度の定義

R を $m \times n$ 行列内の任意の 2×2 行列，$\mathcal{R}$ を $m \times n$ 行列内にある全ての 2×2 行列の集合とする．$\mathcal{R}$ を基準とする食い違い度 $d_{\mathcal{R}}$ を次のように定義する．$m \times n$ 行列 $A \in [0,1]^{m \times n}$ と出力行列 $B \in \{0,1\}^{m \times n}$ との間の食い違い度 $d_{\mathcal{R}}$ は

$$d_{\mathcal{R}}(A,B) := \frac{\sum_{R \in \mathcal{R}} |A_R - B_R|}{|\mathcal{R}|}$$

で定義される．ただし，

$$A_R \ : \ A \text{ 内の指定された行列 } R \text{ の要素の和 ,}$$
$$B_R \ : \ B \text{ 内の指定された行列 } R \text{ の要素の和}$$

である．また， B^* を $d_{\mathcal{R}}(A,B^*)$ を最小にする行列 (最適解) とする．

さらに，ある正の実数 x に対して，

$$\Pr(X = \lfloor x \rfloor + 1) = x - \lfloor x \rfloor,$$
$$\Pr(X = \lfloor x \rfloor) = 1 - (x - \lfloor x \rfloor)$$

を満たすとき，確率変数 X を x の**ランダム丸め**と呼ぶ.

3.2　ランダム丸めを用いた手法

B. Doerr らは上で定義された食い違い度によるランダム丸めを用いて以下の 3 つの手法を開発し，デジタルハーフトーニングに応用した．また，それぞれの手法についての食い違い度の解析，出力画像の見た目の評価も行なった [2, 3].

3.2.1 独立なランダム丸め

行列 B の各々の要素 $b_{ij}(i = 0, 1, \cdots, n,\ j = 0, 1, \cdots, m)$ が行列 A の要素 a_{ij} のランダム丸めであり，かつ全てのこれらの丸めが互いに独立であるとき， B を A の**独立なランダム丸め**と呼ぶ．これを用いた手法は，画像処理の分野で白色雑音 $(white\text{-}noise)$ による閾値処理法としても知られている．

定理 3.2.1 *[2, 3].* 独立なランダム丸めの食い違い度の期待値 $E\big(d_{\mathcal{R}}(A, B)\big)$ は

$$E\big(d_{\mathcal{R}}(A, B)\big) \leq 0.82944,$$
$$E\big(d_{\mathcal{R}}(A, B)\big) \leq 0.75 + d_{\mathcal{R}}(A, B^*)$$

である．

3.2.2 同時ランダム丸め

$a_1, a_2 \in [0, 1]$ とする．以下の条件を満たすとき，$(b_1, b_2) \in \{0, 1\}^2$ を (a_1, a_2) の**同時ランダム丸め**と呼ぶ．

1. 全ての $i \in \{1, 2\}$ に対して， b_i は a_i のランダム丸めである．
2. $b_1 + b_2$ は $a_1 + a_2$ のランダム丸めである．

つまり，全ての $a_1, a_2 \in [0, 1]$ に対して， $a_1 + a_2 \leq 1$ のとき

$$(b_1, b_2) = \begin{cases} (1, 0) & \text{確率 } a_1, \\ (0, 1) & \text{確率 } a_2, \\ (0, 0) & \text{確率 } 1 - a_1 - a_2. \end{cases}$$

また，$a_1 + a_2 > 1$ のとき

$$(b_1, b_2) = \begin{cases} (1, 0) & \text{確率 } 1 - a_2, \\ (0, 1) & \text{確率 } 1 - a_1, \\ (1, 1) & \text{確率 } a_1 + a_2 - 1. \end{cases}$$

と表すことができる．

定理 3.2.2 *[2, 3].* 同時ランダム丸めを用いた手法の食い違い度の期待値 $E\big(d_{\mathcal{R}}(A, B)\big)$ は

$$E\big(d_{\mathcal{R}}(A, B)\big) \leq 0.7111,$$
$$E\big(d_{\mathcal{R}}(A, B)\big) \leq 0.625 + d_{\mathcal{R}}(A, B^*)$$

である．

3.2.3 連続ブロックランダム丸め

ボックス $X = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix} \in [0,1]^{2\times 2}$ とする．以下の条件を満たすとき，ボックス $Y = \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} \in \{0,1\}^{2\times 2}$ を X の**ブロックランダム丸め**と呼ぶ.

1. 全ての $i,j \in \{1,2\}$ に対して，y_{ij} は x_{ij} のランダム丸めである．すなわち，
$$\Pr(y_{ij} = 1) = x_{ij},$$
$$\Pr(y_{ij} = 0) = 1 - x_{ij}.$$

2. ボックス Y 内の縦または横に並んだ 2 つの要素がそれに対応するボックス X 内の 2 つの要素の同時ランダム丸めである．すなわち，全ての $(i,j),(i',j') \in \{1,2\}$ ($i \neq i'$ か $j \neq j'$ のどちらか一方) に対して，
$$\Pr(y_{ij} + y_{i'j'} = \lfloor x_{ij} + x_{i'j'} \rfloor + 1) = x_{ij} + x_{i'j'} - \lfloor x_{ij} + x_{i'j'} \rfloor,$$
$$\Pr(y_{ij} + y_{i'j'} = \lfloor x_{ij} + x_{i'j'} \rfloor) = 1 - (x_{ij} + x_{i'j'} - \lfloor x_{ij} + x_{i'j'} \rfloor)$$
が成り立つ.

3. ボックス自身がランダム丸めを満たす. つまり，
$$\Pr\left(\sum_{i,j\in\{1,2\}} y_{ij} = \left\lfloor \sum_{i,j\in\{1,2\}} x_{ij} \right\rfloor + 1\right) = \sum_{i,j\in\{1,2\}} x_{ij} - \left\lfloor \sum_{i,j\in\{1,2\}} x_{ij} \right\rfloor,$$
$$\Pr\left(\sum_{i,j\in\{1,2\}} y_{ij} = \left\lfloor \sum_{i,j\in\{1,2\}} x_{ij} \right\rfloor\right) = 1 - \left(\sum_{i,j\in\{1,2\}} x_{ij} - \left\lfloor \sum_{i,j\in\{1,2\}} x_{ij} \right\rfloor\right).$$

$A \in [0,1]^{m\times n}$ とする．以下の条件を満たすとき， B を A の**連続ブロックランダム丸め**と呼ぶ.

1. 全ての $i \in \left\{1,2,\cdots,\frac{m}{2}\right\}, j \in \{1,2,\cdots,n-1\}, R := \{2i-1,2i\} \times \{j,j+1\}$ に対して， B_R は A_R のブロックランダム丸めである.

2. m が奇数のとき，全ての $j \in [n-1]$ に対して， $(b_{m,j}, b_{m,j+1})$ は $(a_{m,j}, a_{m,j+1})$ の同時ランダム丸めである.

3. b_{ij} は全ての $b_{i'j'}$ $(\lceil i/2 \rceil \neq \lceil i'/2 \rceil)$ と互いに独立である.

定理 3.2.3 *[2, 3].* 連続ブロックランダム丸めを用いた手法の食い違い度の期待値 $E\left(d_{\mathcal{R}}(A,B)\right)$ は
$$E\left(d_{\mathcal{R}}(A,B)\right) \leq 0.5463,$$
$$E\left(d_{\mathcal{R}}(A,B)\right) \leq 0.3125 + d_{\mathcal{R}}(A,B^*)$$
である.

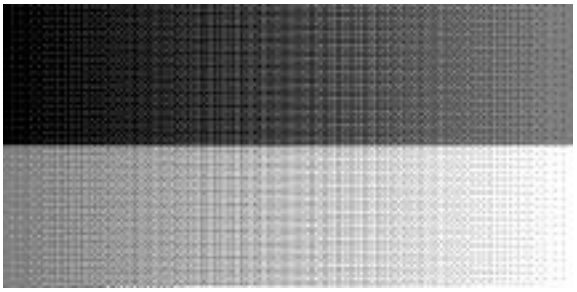
3.3　既存の手法による実験結果

前節で紹介したランダム丸めを用いた手法をデジタルハーフトーニングに応用して実験を行った．入力画像は 256 階調で表現されたグラデーション画像 (1024×128pix)，明るさが一定 (輝度値 16) の画像 (256×256pix)，風景画像 (1000×700pix) の 3 つとした．グラデーション画像を使用した理由は，中間色の表現と滑らかさを測るためであり，明るさが一定の画像を使用した理由は，点の一様分布性と不規則性を測るためである．図 3.1, 3.2 は 2 つの原画像（256 階調で表現されたグラデーション画像と明るさが一定の画像) とそれぞれの手法を適用して出力された画像である．図 3.3 〜 3.8 は風景画像とそれぞれの手法を適用して出力された画像である．また，既存のデジタルハーフトーニング手法と比較するためにオーダードディザ法，エラーディフュージョン法による出力画像も載せておく．

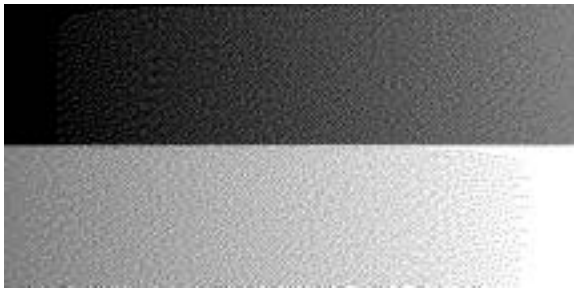
ランダム丸めを用いた手法による実験結果の画像を原画像と見た目の比較すると，原画像との類似度は独立なランダム丸め法が一番低く，連続ブロックランダム丸め法が一番高い．これに対してそれぞれの食い違い度の期待値は，前節で述べられたように独立なランダム丸め法が一番大きく，連続ブロックランダム丸め法が一番小さい．つまり，ランダム丸めの処理を施す際，処理時に互いに影響を及ぼし合っているピクセルの数が増えるほど食い違い度の期待値が小さくなることがわかる．

明るさが一定の画像に着目すると，Doerr らの手法による出力画像には他のデジタルハーフトーニング手法の出力画像に表れるような特有の模様が表れていない．しかしグラデーションにおける出力画像に注目すると，連続ブロックランダム丸め法の出力画像は輝度 86〜128,128〜170 で粒状の点模様が目立ち，滑らかさに欠けた画像になっている．

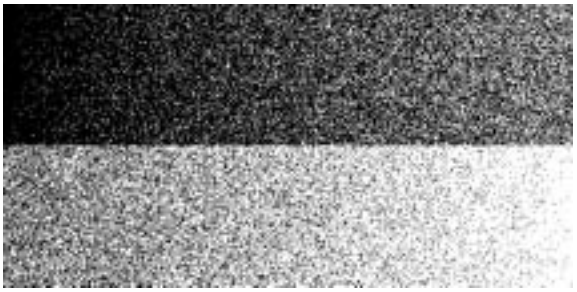
(a) 原画像



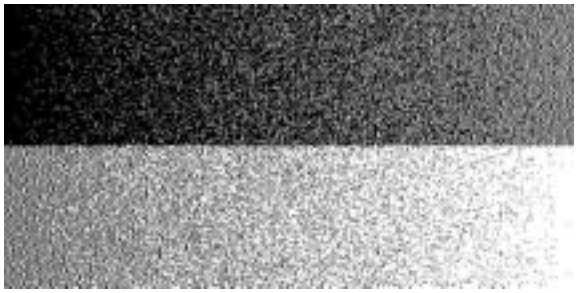
(b) オーダードディザ法



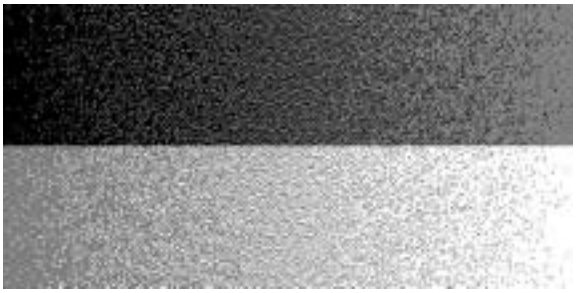
(c) エラーディフュージョン法



(d) 独立なランダム丸め法

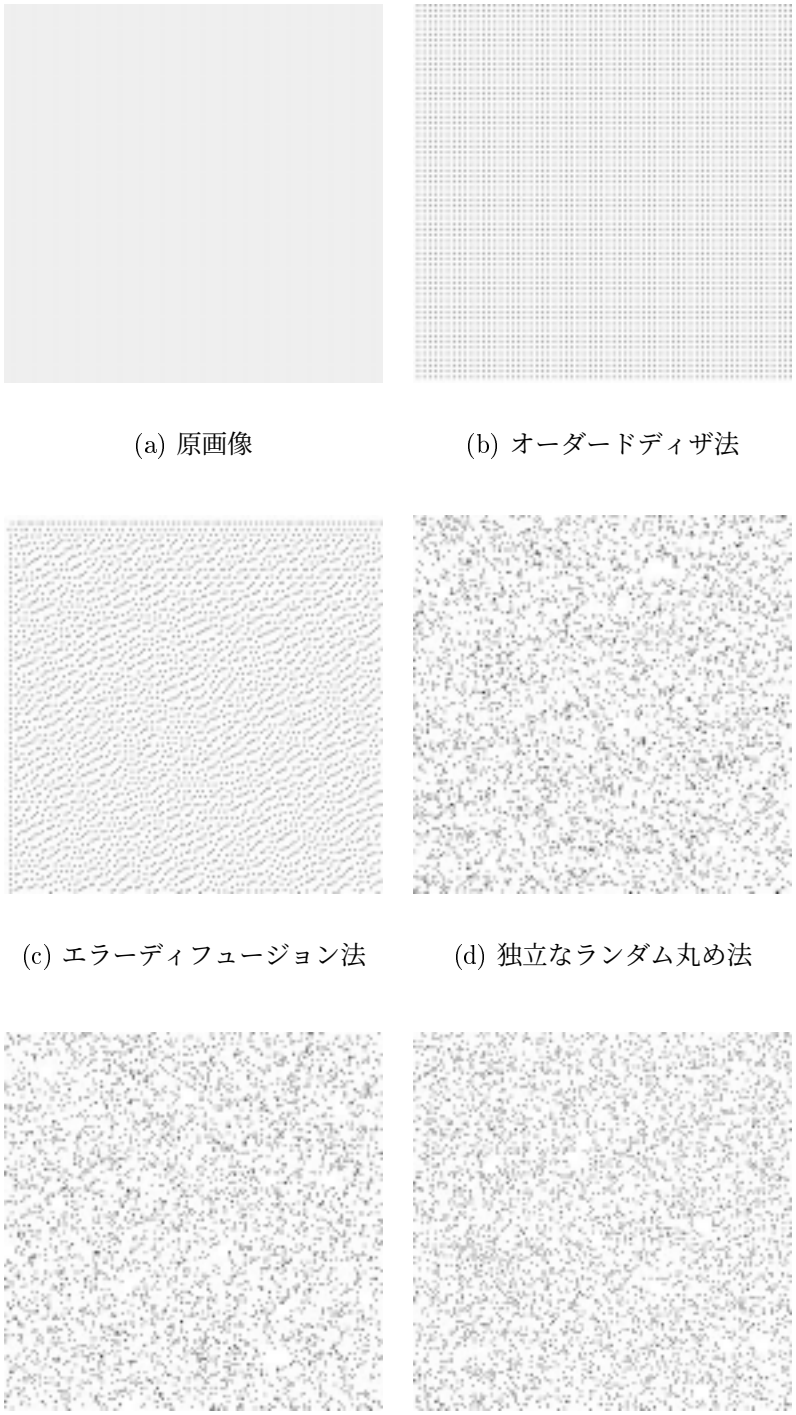


(e) 同時ランダム丸め法



(f) 連続ブロックランダム丸め法

図 3.1: グラデーション画像を入力とした既存手法による出力結果



(a) 原画像　(b) オーダードディザ法

(c) エラーディフュージョン法　(d) 独立なランダム丸め法

(e) 同時ランダム丸め法　(f) 連続ブロックランダム丸め法

図 3.2: 一定の明るさの画像を入力とした既存手法による出力結果

図 3.3: 風景画像を入力とした出力結果：原画像

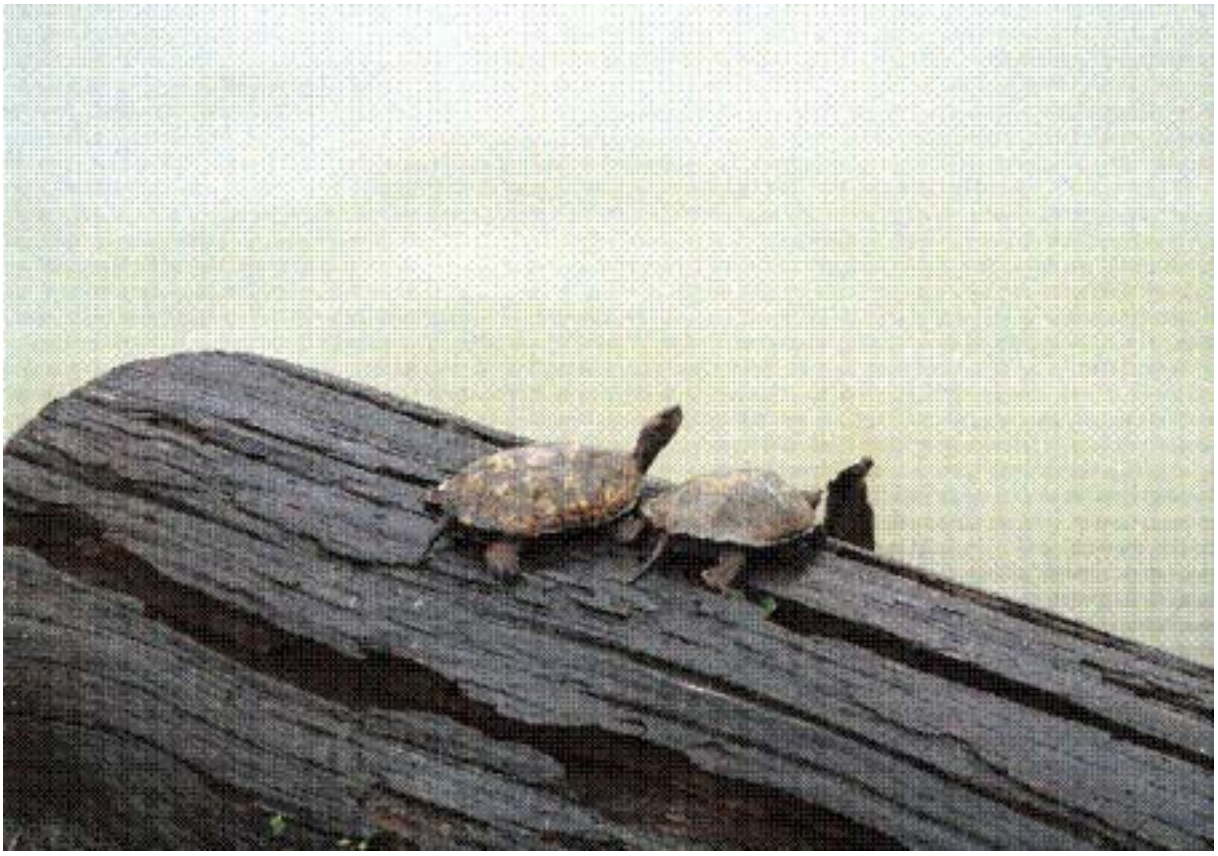


図 3.4: 風景画像を入力とした出力結果：オーダードディザ法

図 3.5: 風景画像を入力とした出力結果：エラーディフュージョン法

図 3.6: 風景画像を入力とした出力結果：独立なランダム丸め法

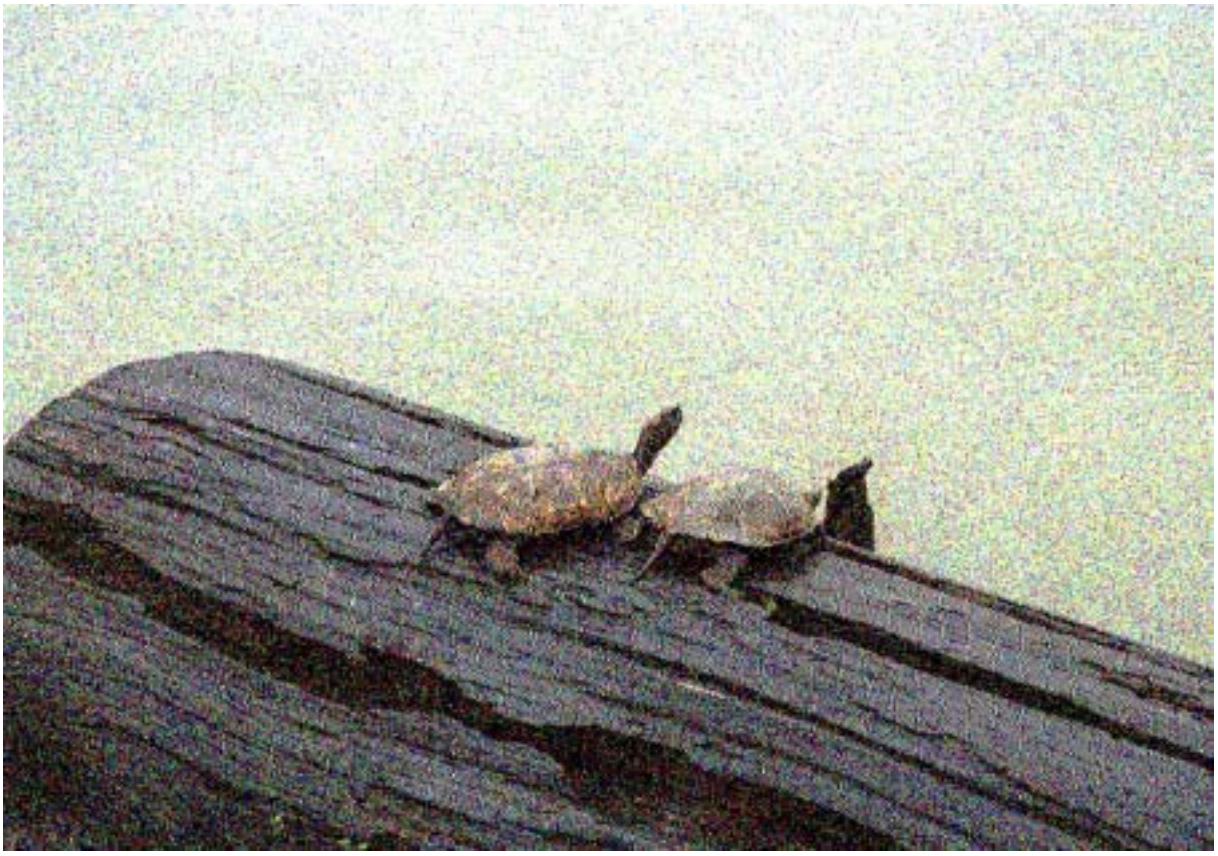


図 3.7: 風景画像を入力とした出力結果：同時ランダム丸め法

図 3.8: 風景画像を入力とした出力結果：連続ブロックランダム丸め

第4章　提案手法

前章までに紹介されたこれまでのランダム丸め法では，各々のピクセルの処理を施す際に誤差を他の未処理のピクセルに伝播していない．そのため，各々のピクセルの処理を施す度に食い違い度 $d_{\mathcal{R}}$ が蓄積され，入力画像のサイズに比例して全体の食い違い度が大きくなってしまう．また，処理を施す対象が 1 ピクセル，または 2×1 の 2 ピクセルといった，とても小さい領域であるため，出力画像に粒状の点模様が目立ったり滑らかさに欠ける，といった欠点が見られる．

粒状の点模様が目立ったり滑らかさに欠ける原因として，各々の処理において誤差を伝播していないことが考えられる．しかし，規則性を持つ走査順序で誤差伝播を処理すると問題を生じることがある．それは，明るさが一定である領域を処理すると規則的な模様が発生することである．この問題を回避するためには走査順序にランダム性を採り入れれば良い．そこで，以下の 2 つの点を考慮して Doerr らのアルゴリズムを改良した新しい手法を 2 つ提案する．

- 処理済みピクセル全体の誤差を次に処理を施す領域に伝播する
- 一方向のみの走査ではなくランダムに方向を変えながら処理を行うために，ランダム性を持つスペースフィリング曲線を用いる

4.1　提案手法１

ピクセル 1 つ 1 つを独立なランダム丸めで処理し，処理済の領域全体の誤差を次に処理を施すピクセルに伝播させる．そして，誤差伝播を受けたピクセルを独立なランダム丸めで処理する．つまり i 番目までの処理済のピクセル全体の誤差を $d_i = \sum_{j=1}^{i}(b_i - a_i)$ と仮定すると， a_{i+1} のランダム丸めは

$$b_{i+1} = \begin{cases} 1 & \text{確率 } a_{i+1} - d_i \\ 0 & \text{確率 } 1 - (a_{i+1} - d_i) \end{cases}$$

となる．

走査順序については，ランダム性をもつスペースフィリング曲線を用いて領域全体をランダムに方向を変えて走査させるようにする．ランダム性をもつスペースフィリング曲線についてはさまざまな曲線パターンが存在し，その生成方法は次章で述べる．

4.2 提案手法２

提案手法１を拡張したものである．大きな変更点は２点ある．

ひとつ目はピクセルを２つずつ同時ランダム丸めで処理を施すことである．つまり i 番目までの処理済のピクセル全体の誤差を $d_i = \sum_{j=1}^{i} \left(b_{1_j} + b_{2_j} - a_{1_j} - a_{2_j}\right)$ と仮定すると，$\left(a_{1_{i+1}}, a_{2_{i+1}}\right)$ の同時ランダム丸めは $a_{1_{i+1}} + a_{2_{i+1}} - d_i \leq 1$ のとき

$$\left(b_{1_{i+1}}, b_{2_{i+1}}\right) = \begin{cases} (1,0) & \text{確率 } a_1 - d_i/2, \\ (0,1) & \text{確率 } a_2 - d_i/2, \\ (0,0) & \text{確率 } 1 - (a_1 + a_2 - d_i), \end{cases}$$

また，$a_{1_{i+1}} + a_{2_{i+1}} - d_i > 1$ のとき

$$\left(b_{1_{i+1}}, b_{2_{i+1}}\right) = \begin{cases} (1,0) & \text{確率 } 1 - a_2 + d_i/2, \\ (0,1) & \text{確率 } 1 - a_1 + d_i/2, \\ (1,1) & \text{確率 } a_1 + a_2 - d_i - 1 \end{cases}$$

となる．

このように，ピクセルを 2 つずつ同時ランダム丸めで処理を施すことによって，理論的に食い違い度の期待値が小さくなるので，中間色をさらに細かく表現することができると考えられる．

ふたつ目は，スペースフィリング曲線の走査順序における処理方法である．Doerr らのアルゴリズムでは常に走査方向が右方向に固定されている．しかしながら，スペースフィリング曲線は上下左右の 4 方向に走査するため，Doerr らの同時ランダム丸め法のように常に縦に隣接した 2 ピクセルを処理し続けることは不可能である．そのため，ピクセルを２つずつ処理できるようにスペースフィリング曲線における処理方法を改良しなければならない．

提案手法 2 では 4×4 ピクセルごとに分割してスペースフィリング曲線を生成しているので，その曲線の走査では 2×2 ピクセルごとの処理として考えることができる．つまり 2×2 ピクセル内での 2 回の同時ランダム丸めを 1 つの処理とまとめることができるので，2×2 ピクセル内での同時ランダム丸めの方法のみを考えればよい．2×2 ピクセル内において，走査方向が右方向 (または左方向) の場合は，縦に隣接した 2 ピクセルを右方向 (または左方向) に同時ランダム丸めを 2 回処理することと決めておく．また同様に，走査方向が上方向 (または下方向) の場合は横に隣接した 2 ピクセルを上方向 (または下方向) に同時ランダム丸めを 2 回処理することと決めておく．各々の処理における誤差伝播方法は提案手法 1 と同様に，次に処理を施す領域に誤差伝播を行う．

これらの手法を用いた実験結果については，第 6 章で説明する．

第5章　操作順序：スペースフィリング曲線

スペースフィリング曲線 (*space filling curve*) とは，縦横の格子辺により構成される格子平面において，縦の格子辺，横の格子辺に挟まれた各小領域の中央の点 (これを格子点と呼ぶ) を，一つの漏れも重複もなく全てたどることのできる曲線のことを言う (図 5.1).

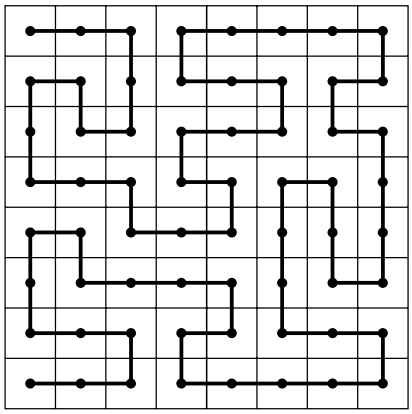


図 5.1: 例：スペースフィリング曲線

スペースフィリング曲線は，1890 年にイタリアの数学者ペアノによって初めてその一例を発見されて以来，様々な生成方法が研究されてきた．また，スペースフィリング曲線は，単に学術上の興味に留まるものではなくいくつかの分野で応用され成果を上げている [4]．代表的なスペースフィリング曲線としてヒルベルト曲線，ペアノ曲線，シェルピンスキー曲線など多くの生成方法があるが，どの生成方法にもある規則性が存在し，デジタルハーフトーニングに応用しても特有の模様が現れてしまう．そのため，本研究ではランダム性をもつスペースフィリング曲線の形を生成する手法を使用する．

ここではランダム性をもつスペースフィリング曲線の生成方法として，

- 単純なスペースフィリング曲線からランダムに変形させる方法
- ランダム性を持つ全域木を生成し，その周りをたどってスペースフィリング曲線を生成する

の 2 つに大別して説明する.

5.1 単純なスペースフィリング曲線から変形させる方法

ここでは，入力画像のサイズに対して一度スネークパターンの形をしたスペースフィリング曲線を作成し，その曲線を少しずつランダムに変形させて新しいランダム性のあるスペースフィリング曲線を生成する方法を紹介する.

5.1.1 WFCC を用いた手法

Asano ら [4] が提案した WFCC(Well Formed Cell Coloring) を用いたスペースフィリング曲線の生成法を説明する.

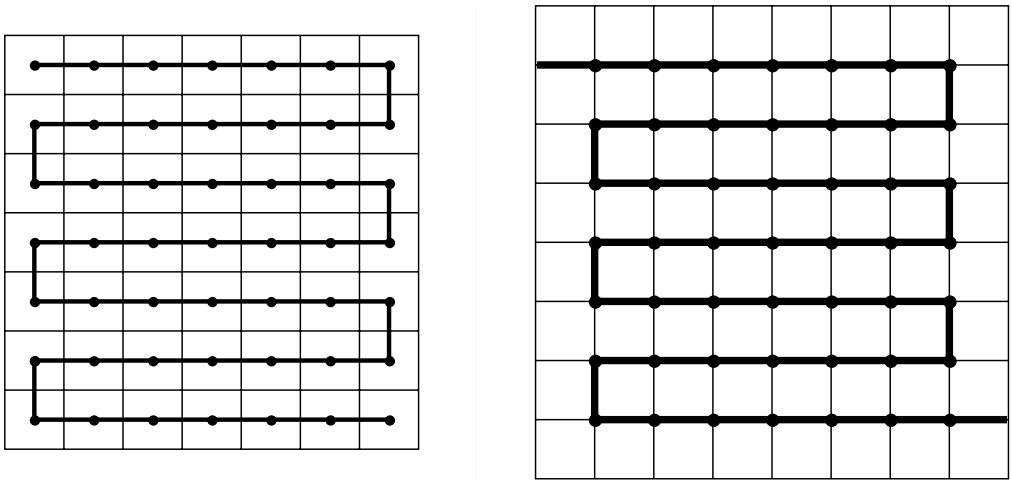
step1 図 5.2(a) のように与えられた入力画像の各ピクセルの中心点を格子点として置き換え，縦横の直線により構成される新しい格子平面状 (図 5.2(b)) で考える．スペースフィリング曲線の形はスネークパターンの形とする．このとき，スペースフィリング曲線が各格子点と格子辺を辿っていることがわかる.

step2 新しく構成された格子平面において，縦の格子辺と横の格子辺に囲まれた小平面をセルと呼ぶことにする．スペースフィリング曲線の左側のセルを黒，右側のセルを白に塗ってゆくと図 5.3 が得られる.

step3 そして，スペースフィリング曲線の状態を維持させながら，ある規則に基づいてランダムに黒のセルと白のセルを数回交換 (フリップ) していく.

step4 黒白のセルでランダムに敷き詰められた格子平面から，左側が黒のセル，右側が白のセルとなるようにスペースフィリング曲線を描く (図 5.4(a)).

step5 step4 で得られた格子平面を元の入力画像に当てはめる (図 5.4(b)).



(a) 入力画像 (変換前)　　(b) 格子平面 (変換後)

図 5.2: WFCC のアルゴリズム：Step1

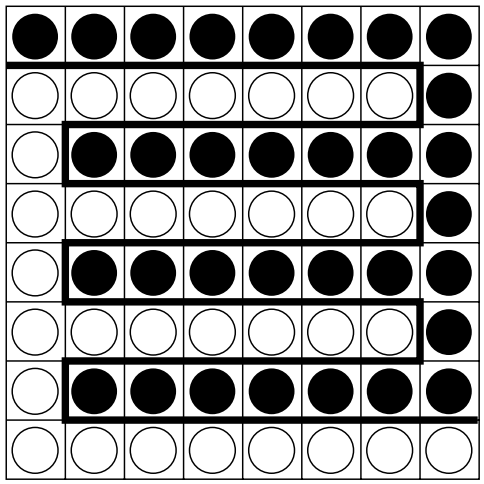
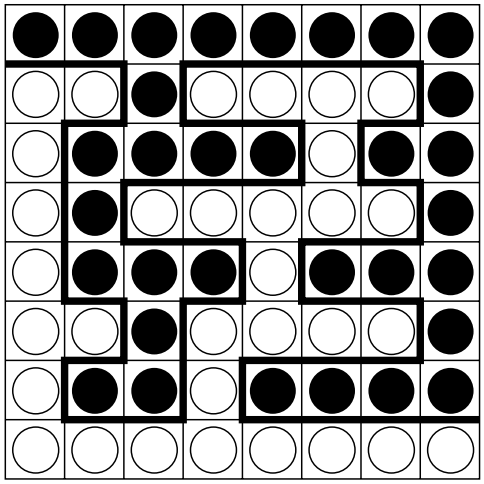
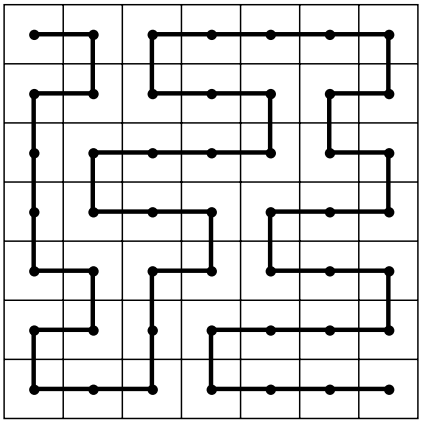


図 5.3: WFCC のアルゴリズム：Step2



(a) 数回のフリップ後の格子平面



(b) 生成されたランダム性を持つスペースフィリング曲線

図 5.4: WFCC のアルゴリズム：Step4,Step5

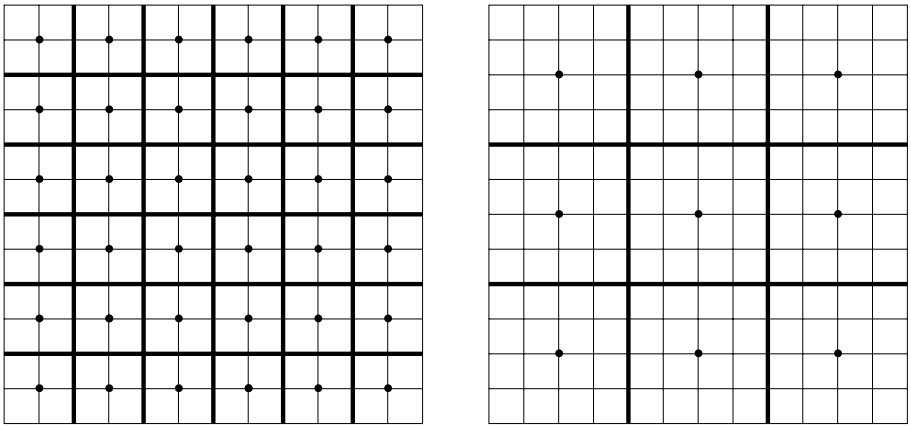
ここで問題となるのが，何回のフリップ回数でランダム性を持つスペースフィリング曲線が生成されるかである．フリップの回数については，宍戸の実験から大体入力画像のサイズ分の回数で規則性が見られなくなることがわかる [6]．たとえば，入力画像のサイズが 256×256 ピクセルとすると 65536 回のフリップ回数となる．

5.2 全域木を利用した方法

スペースフィリング曲線のもう一つの生成方法として，全域木を利用した方法がある．まず始めに，入力画像に対して 2×2 ピクセルごとに分割し，それぞれの 2×2 ピクセルの中心に点を打つ (図 5.5(a))．(提案手法 2 の場合は 4×4 ピクセルごとに分割させ，4×4 ピクセルの中心に点を打つ (図 5.5(b)))．

その点の集合から全域木を生成し全域木の周りをたどることによって，スペースフィリング曲線が生成される (図 5.6)．

この全域木の生成について，2 つの生成方法を紹介する．



(a) 提案手法 1 の場合　　(b) 提案手法 2 の場合

図 5.5: 格子平面上における点集合の表現方法

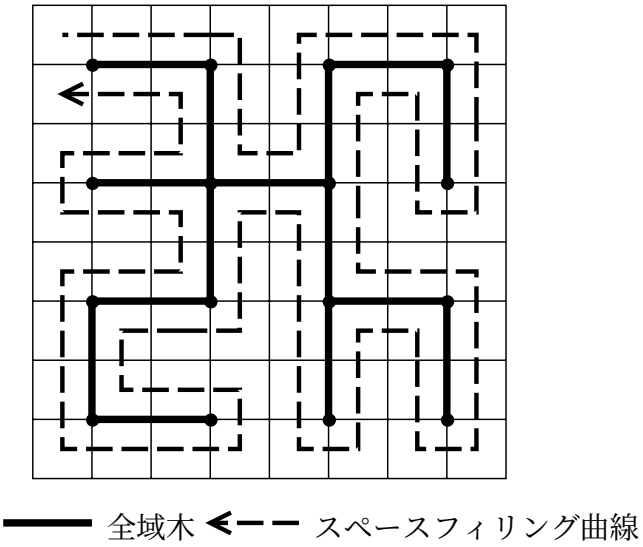


図 5.6: 全域木を用いたスペースフィリング曲線

5.2.1 RSFC を用いた手法

RSFC(Random Space Filling Curve) を用いた手法は，点集合のある 1 つの点を根 (root) として，そこから枝 (edge) を延ばしていく手法である．RSFC によって生成される全域木はとても深い枝を持っているといった特徴を持つ．

全域木の生成方法は以下の手順で生成される．

step1 左上コーナーの点を始点 (全域木の根) とする．

step2 現在の位置から木構造を保ちながら可能な限り枝分かれせずに辺を延ばしていく．木構造を保つとは，サイクルを含まないかつ，連結性を保っていることである．

step3 これ以上，枝分かれせずに辺を延ばすことができなくなったのならば，新たに枝分かれをして辺を延ばすことが可能な点まで戻る．

step4 始点 (根) まで戻ってきたならば step5 へ進む．そうでなければ step2 へ戻る．

step5 全域木が生成されたので，全域木の周りに沿ってスペースフィリング曲線を作成する．

全域木を生成する際，格子状の点集合なので枝分かれは多くても 3 方向である (図 5.7)．また，辺を延ばす時の方向の候補は均一の確率で選んでいるが，直進方向あるいは左右方向に重みを持った確率で方向を選び，全域木を生成する方法も考案した．

それぞれの全域木の出力結果を図 5.8 に示す．直進方向に重みを持った全域木 (図 5.8(b)) では，直進が多いためにエラーディフュージョン法のように誤差伝播処理特有の模様が表れてしまう．また，左右方向に重みを持った全域木 (図 5.8(c)) では，ギザギザ状の形をした枝が所々に現れており, ランダム性に欠けている．

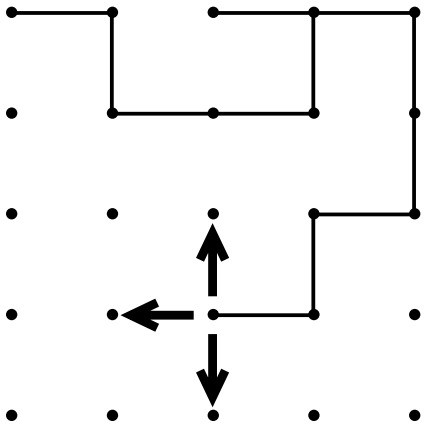
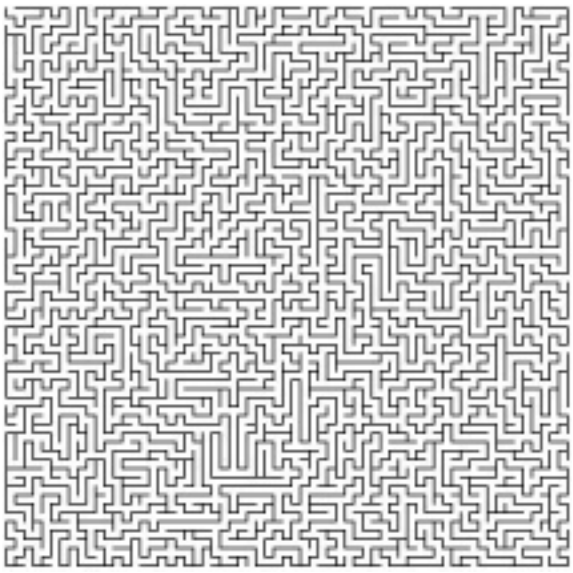
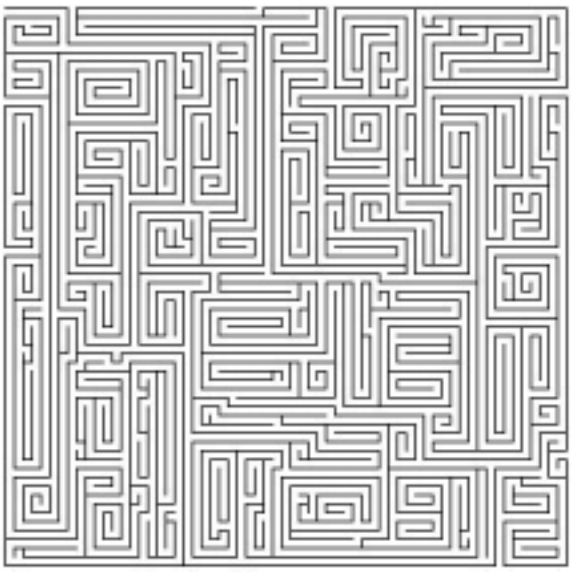


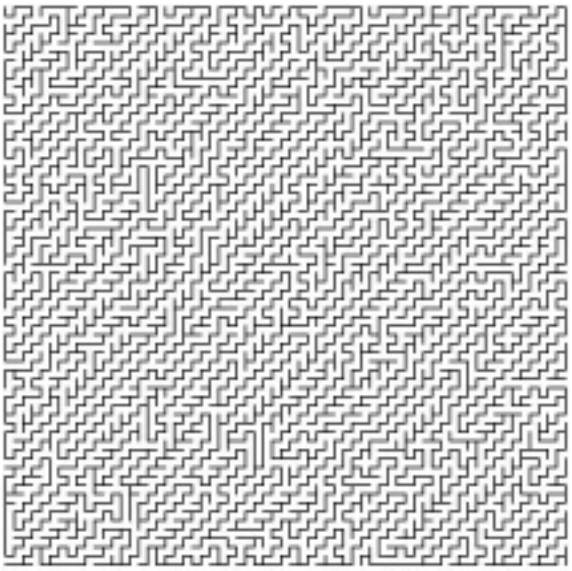
図 5.7: 辺の延長可能な方向



(a) 等確率に枝分かれした全域木



(b) 直線方向に重みを持った全域木



(c) 左右方向に重みを持った全域木

図 5.8: RSFC で生成した全域木 (頂点数：64×64)

5.2.2 集合ユニオンファインド木を用いた手法

任意の集合から全域木を高速に生成する手法の一つである集合ユニオンファインド木 ($set\ union\ find\ tree$) を用いて，ランダム性のある全域木を生成する．[7]

集合ユニオンファインド木では互いに素な集合の族を扱い以下の 2 つの操作を実行して全域木を生成する．

- $find(x)$: 要素 x を含む集合の名前を返す.
- $union(x, y)$: 要素 x を含む集合 X と要素 y を含む集合 Y を合併する．ただし $X \wedge Y = \emptyset$ を仮定している．集合 X ， Y は集合族から除去され集合 $X \wedge Y$ が集合族に入る．

集合の各要素 x は木のノードに対応 (要素 x とノードとを同一視する) し，各ノード v ではその親 $p(v)$ が記憶されている．さらに，根の要素を集合の代表元，すなわち，集合名と考える (図 5.9).

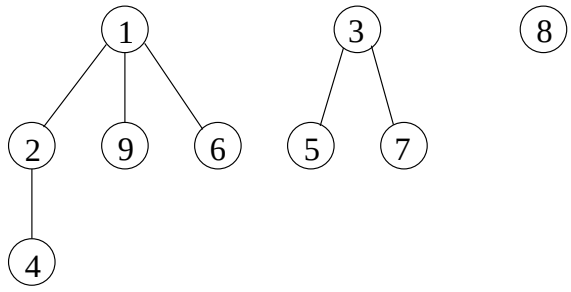


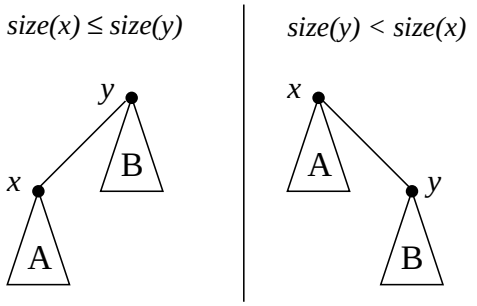
図 5.9: 集合族 {{1,2,4,6,9},{3,5,7},{8}} を表現するユニオンファインド木

$find(x)$ を行なうためには，単に $p(p(x)), \ldots$ と順に親をたどっていき根を求めればよい．また， $union(x, y)$ を行なうには，最初 $xx := find(x)$ ， $yy := find(y)$ を行い，次に， $p(yy) := xx$ (yy の親を xx) とすればよい.

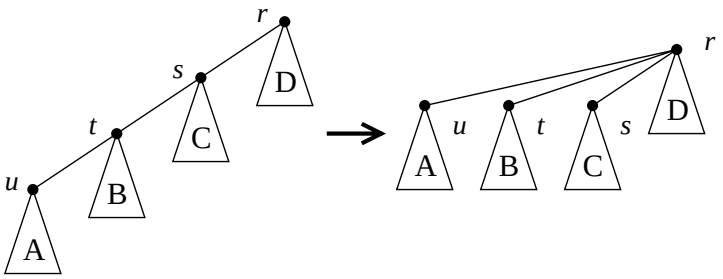
このままでは計算時間が大きいため，サイズによる合併とパス圧縮という工夫を用いて高速化を図る.

- サイズによる合併 : 根 x, y に対して $union(x, y)$ を行なうとき， $size[x] \leq size[y]$ なら $p(x) := y$ とし， $size[x] > size[y]$ ならば $p(y) := x$ とする (図 5.10(a)).
- パス圧縮 : $find(x)$ で x から根までたどるとき，根を除くそのパス上の全てのノードに対して親を根にする (図 5.10(b)).

基本的には，サイズによる合併では木の深さがなるべく大きくならないようにしており，パス圧縮では以降の $find$ を効率よく行なえるように前もって準備しているといえる.



(a) サイズによる合併



(b) パス圧縮

図 5.10: 集合ユニオンファインド木生成の高速化

したがって，サイズによる合併とパス圧縮を用いれば，$find$ の操作は最悪でも $O(\log n)$ の手間，$union$ の操作は最悪でも $O(1)$ の手間で行なうことができる．$union, find$ の操作を合わせて m 回 ($union$ は高々 $n-1$ 回) 行なうものとすると，全体でならした意味で，各操作 1 回当たり $O(\alpha(m,n))$ の手間で実行できることがわかっている．なお，$\alpha(m,n)$ はアッカーマン逆関数 (*functional inverse of Ackermann function*) と呼ばれるもので，通常の問題では，$\alpha(m,n) \leq 4$ ($n < 2^{16} = 65536$ ならば $\alpha(m,n) \leq 3$) と考えてよい．

このアルゴリズムから生成される全域木の出力結果を図 5.11 に示す．

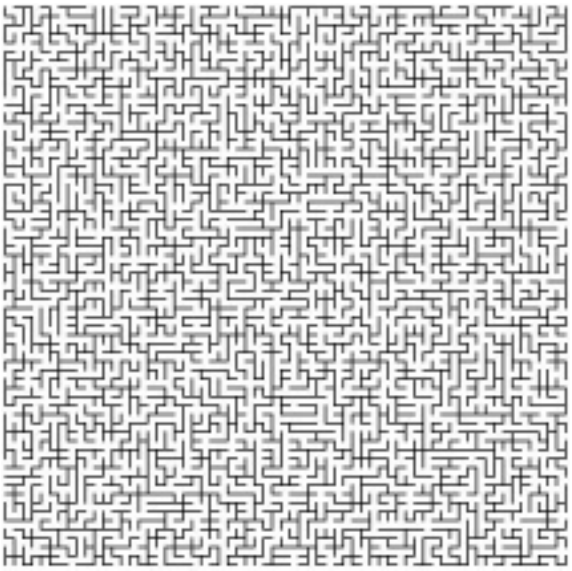


図 5.11: 集合ユニオンファインド木 (頂点数：64×64)

5.3 スペースフィリング曲線の出力結果

これまでに述べられた 3 つのランダム生を持つスペースフィリング曲線の出力結果を以下に示す (図 5.12, 5.13, 5.14)．これらのスペースフィリング曲線のサイズは 128×128 である．また，RSFC を用いた手法については，等確率で進行方向を決定する場合を採用した．

WFCC を用いて生成したスペースフィリング曲線 (図 5.12) は，若干横方向に長い直線が残っている．また，計算時間が入力サイズ n に対して $O(n^2)$ であるので，この曲線を用いた実用化は難しい．

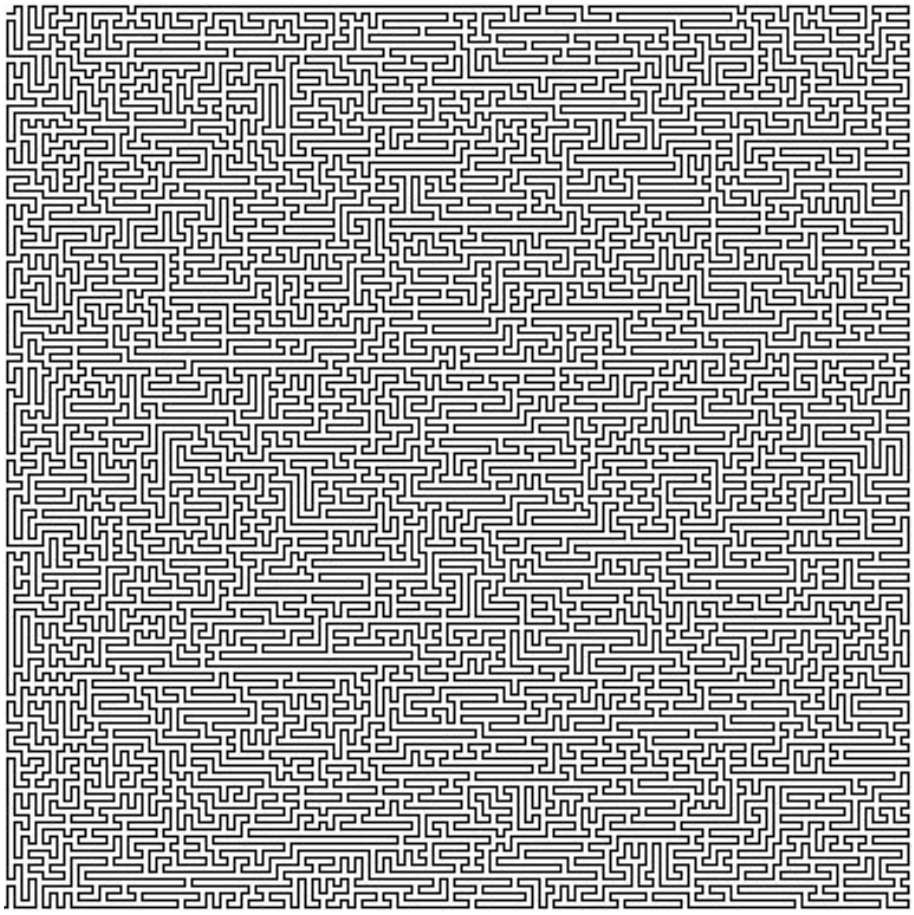


図 5.12: WFCC を用いて生成したスペースフィリング曲線

RSFC を用いて生成したスペースフィリング曲線 (図 5.13) は，いたるところにパイ生地の断面のような模様が見られる．計算時間については入力サイズ n に対して $O(n)$ である．

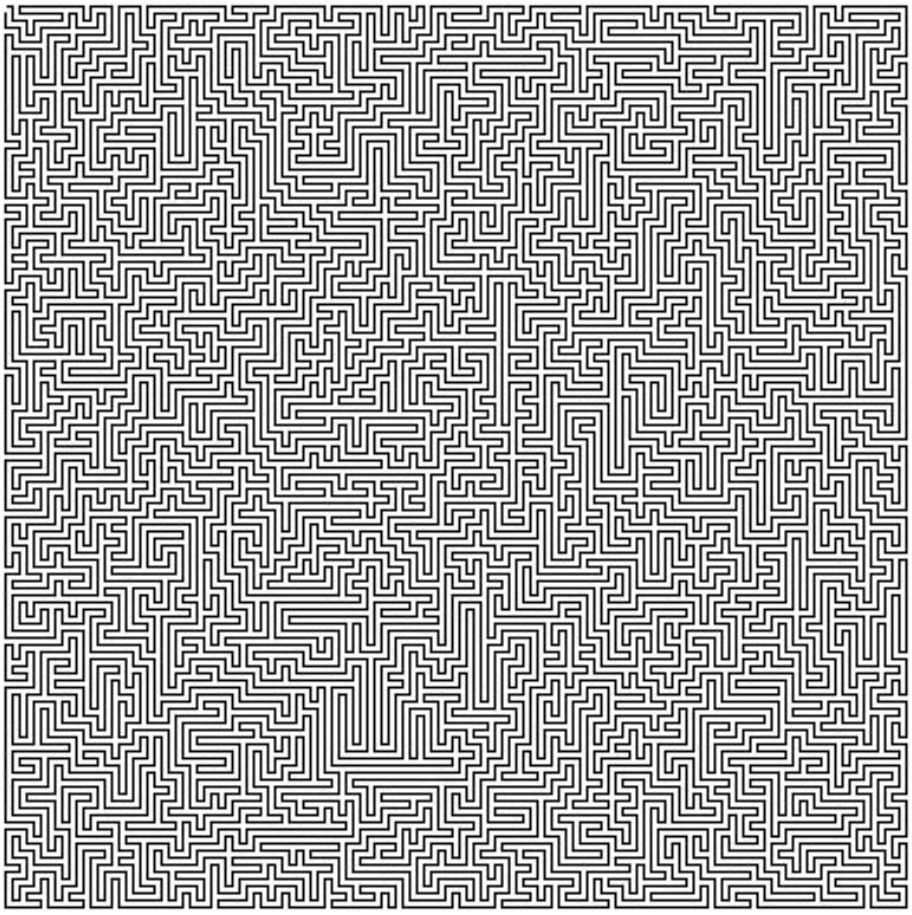


図 5.13: RSFC を用いて生成したスペースフィリング曲線

集合ユニオンファインド木を用いて生成したスペースフィリング曲線(図5.14)は，WFCCやRSFCを用いた手法の時に表れていた特有のパターンが見られない．計算時間については前節で述べたように，入力サイズ n に対して $O(\alpha(m,n)n)(m<n)$ である．

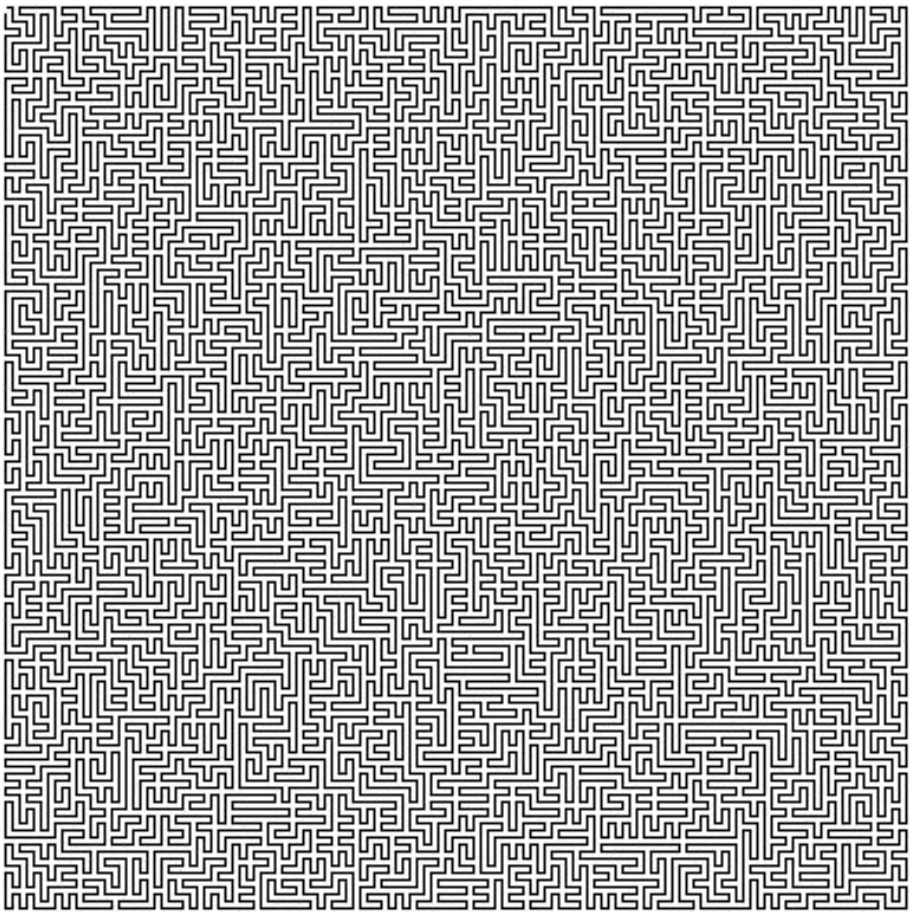


図 5.14: 集合ユニオンファインド木を用いて生成したスペースフィリング曲線

第6章　実験結果

第5章で紹介したいくつかのランダム性を持つスペースフィリング曲線を用いて提案手法1，2をそれぞれ実装した．また既存の手法も実装し，提案手法との比較に用いた．

6.1　準備

入力となる原画像として，以下の3つのRGBそれぞれに0から255までの連続階調で表されているPPM画像を用いる．

- グラデーション画像 (1024×128pix)

  緩やかな階調変化における画像の滑らかさや模様の有無について測ることができる．

- 一定の明るさ (輝度16) の画像 (256×256pix)

  どの領域を取ってきても明るさは一定であるので，点の一様分布性について測ることができる．

- 風景画像 (1000×700pix)

  池の滑らかな階調変化における画像の滑らかさ，木や亀の輪郭部分や模様の再現性について測ることができる．

(既存手法を実装して得られた出力結果は第3章の実験結果に掲載 (図3.1〜3.8) しているので，この章では省略する．)

また，それぞれの入力の画像に対する出力画像は0または255しか持たないPPM画像とする．

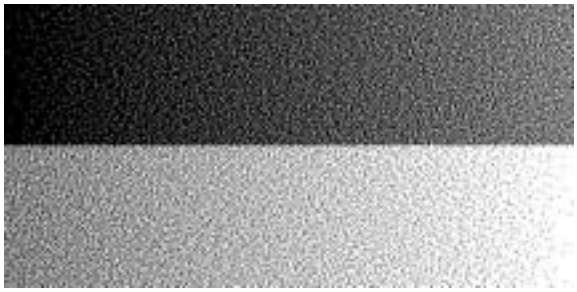
6.2　出力結果と考察

6.2.1　入力画像：グラデーション画像

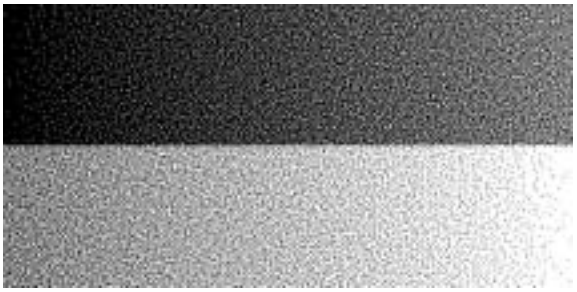
グラデーション画像を入力画像としたそれぞれの手法の出力結果は図 6.1 のとおりである.

連続ブロックランダム丸め法で問題となっていた滑らかさについては，どの新しい手法の出力画像においても改善されている.

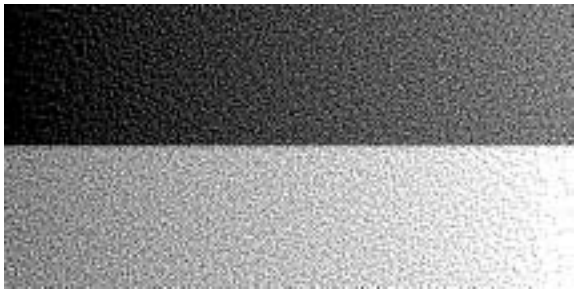
提案手法 1 と提案手法 2 との比較については，どの手法においても提案手法 2 のほうが階調変化を細かく表現しており，滑らかな画像を出力している．WFCC を用いたスペースフィリング曲線は横方向に長い直線が残っていたため，WFCC を用いた手法の出力画像 (図 6.1(a),6.1(b)) には誤差を伝播した時に表れる斜めの線模様が表れている．RSFC を用いた手法の出力結果 (図 6.1(c),6.1(d)) では，明るさが非常に明るい領域または非常に暗い領域で誤差を伝播したときに表れる斜めの線模様が若干表れている．ユニオンファインド木を用いた手法の出力結果 (図 6.1(e),6.1(f)) では，特に目立った模様などは表れていない.



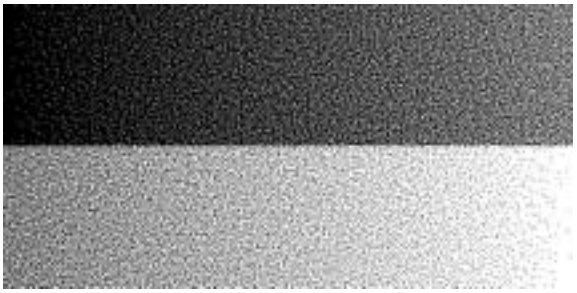
(a) 提案手法 1：WFCC を用いた手法



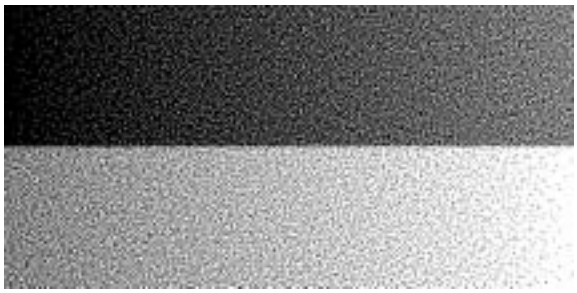
(b) 提案手法 2：WFCC を用いた手法



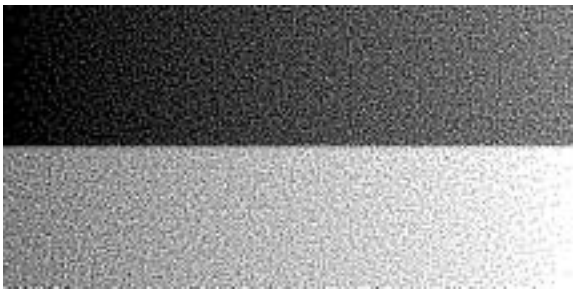
(c) 提案手法 1：RSFC を用いた手法



(d) 提案手法 2：RSFC を用いた手法



(e) 提案手法 1：ユニオンファインド木を用いた手法



(f) 提案手法 2：ユニオンファインド木を用いた手法

図 6.1: グラデーション画像を入力とした出力結果

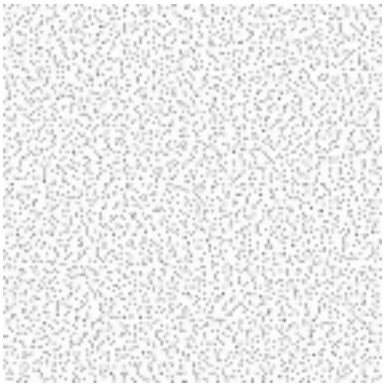
6.2.2　入力画像：一定の明るさの画像

一定の明るさの画像を入力画像としたそれぞれの手法の出力結果は図 6.2 のとおりである.

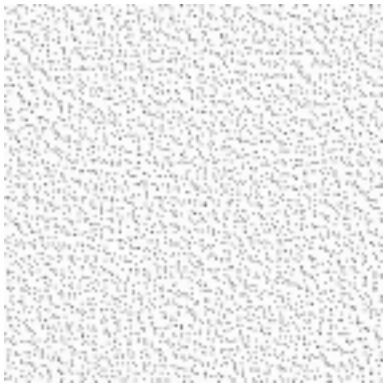
提案手法全ての出力画像の点分布については，Doerr らの手法の出力画像 (図 3.2(d), 3.2(e), 3.2(f)) で現れていた大きな空白の領域 (点が存在しない領域）が見られない．したがって，点の一様分布性については，どの新しい手法においても Doerr らの手法よりも向上していることが言える.

それぞれの手法の出力結果については特に大きな違いは見られないが，RSFC を用いた手法の出力画像 (図 6.2(c), 6.2(d)) では特有の模様が他の手法より多く表れている．しかし，エラーディフュージョン法の出力画像 (図 3.2(c)) で表れたようなはっきりとした模様は見られない.

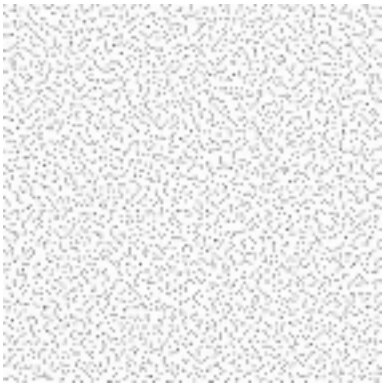
(a) 提案手法 1：WFCC を用いた手法



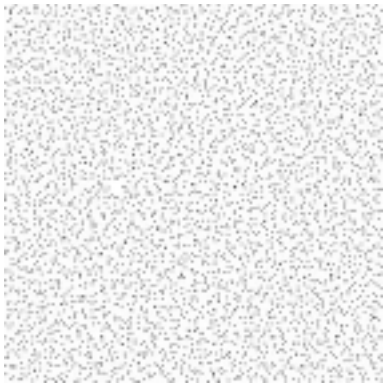
(b) 提案手法 2：WFCC を用いた手法



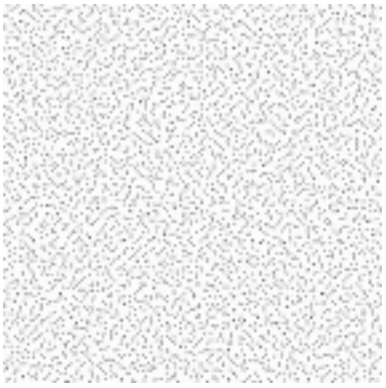
(c) 提案手法 1：RSFC を用いた手法



(d) 提案手法 2：RSFC を用いた手法



(e) 提案手法 1：ユニオンファインド木を用いた手法



(f) 提案手法 2：ユニオンファインド木を用いた手法

図 6.2: 一定の明るさ (輝度値１６) の画像を入力とした出力結果

6.2.3　入力画像：風景画像

風景画像を入力画像としたそれぞれの手法の出力結果は図 6.3〜6.5 のとおりである.

提案手法全ての出力画像では依然細かい粒状の点模様が残っているが，Doerr らの手法の出力画像 (図 3.6, 3.7, 3.8) より滑らかに表現されており，木や亀の細部の輪郭部分や模様のなどの再現性についても向上している.

また，エラーディフュージョン法の出力画像 (図 3.5) で表れたような斜めの線模様は見られない.

図 6.3: 風景画像：WFCC を用いた手法による出力画像

図 6.4: 風景画像：RSFC を用いた手法による出力画像

図 6.5: 風景画像：ユニオンファインド木を用いた手法による出力画像

第7章　まとめ

7.1　結論

Doerr らのアルゴリズムに誤差伝播法などを取り入れたことによって，Doerr らの手法の出力画像に表れていた粒状の粗い模様を大幅に減らすことに成功した．また，従来の手法で表れていたような特有の模様は表れていない．しかしながら，依然細かい粒状の点模様が残っており，従来の手法に比べて多少の画質の劣化が見られる．これは一度に処理を行う領域が小さいためであり，その領域を大きくすることで画質が向上すると考えられる．

スペースフィリング曲線の生成方法については様々なランダム性を持つ曲線を生成することができた．本研究で用いたスペースフィリング曲線に対しては，生成にかかる時間，曲線の形，出力画像での模様の有無を考慮するとユニオンファインド木を用いた生成法が適していると考えられる．

7.2　今後の課題

本研究では，提案手法の出力結果と既存手法の出力結果を人間の見た目によって比較しながら評価を行った．本研究はデジタルハーフトーニング手法を幾何学的な理論に基づいて評価し，良い点配置を求めることも目的としていた．しかし，提案手法では処理済全体の誤差を伝播しており，またランダム性を持つ走査順序であったため，任意の領域の食い違い度の期待値を計算するのは困難であった．そのため，食い違い度理論に基づいた解析は行われていない．したがって，食い違い度理論に基づいた解析は今後の課題の重要な 1 つである．

また，Doerr らが用いた解析方法を適用することだけでなく，食い違い度理論に基づいた別の解析方法を用いて評価することについても考慮する必要がある．

新しいデジタルハーフトーニング手法の開発については，今後も Doerr らが用いた解析方法を適用するならば，食い違い度理論の基づいた解析の困難さの原因となっていた誤差伝播とランダム性を持つ走査順序を取り除いた手法も考慮しなければならないと思われる．

謝辞

本研究を進めるにあたり，終始御指導をいただきました浅野哲夫教授に深く感謝いたします．また，ゼミなどで大変熱心な御指導をして下さった元木光雄助手，ならびに浅野研究室の皆様に感謝いたします．

参考文献

[1] J. Matoušek. *Geometric Discrepancy : An Illustrated Guide*, Springer, 1999(Algorithms and Combinatorics;18).

[2] B. Doerr and H. Schnieder. Non-independent Randomized Rounding and an Application to Digital Halftoning, in Proc. of *ESA2002*, pp. 399–410, 2002.

[3] B. Doerr. Non-independent Randomized Rounding and an Application to Digital Halftoning, in Proc. of *SODA2003*, pp. 506–507, 2003.

[4] T. Asano, N. Katoh, H. Tamaki, and T. Tokuyama. Convertibility among Grid Filling Curves, in Proc. of *ISAAC'98* (Int. Symp. on Algorithms and Computation), pp.307-316, Taejon, Korea, December 1998.

[5] 中尾 俊行. 食い違い度理論に基づいた離散平面上の点配置問題に関する研究, 北陸先端科学技術大学院大学修士論文, 2002.

[6] 宍戸 健太郎. 格子充填曲線の生成およびその応用に関する研究, 北陸先端科学技術大学院大学修士論文, 2000.

[7] 浅野 孝夫 , 今井 浩. 計算とアルゴリズム, オーム社, 2000.