

## シームレステクスチャの生成法に関する研究

山田, 辰美

<https://doi.org/10.11501/3168353>

---

出版情報 : 九州芸術工科大学, 1999, 博士 (芸術工学), 課程博士  
バージョン :  
権利関係 :

## 第2章 シームレステクスチャの生成法

### 2.1 まえがき

テクスチャマッピングはCGのリアル性など質感を向上させるための重要な技術である。しかし、第1章で述べたように、基本テクスチャパターンの大きさがマッピング領域の大きさより小さい場合、基本パターンを繰り返して貼り付ける必要があり、そのため繰り返し部分の境界が不連続になる。このことが、画像品質を著しく低下させるという結果をもたらす。

本章では、この問題に対処するために、以下の条件を満たす基本テクスチャパターンの生成法を示す。

[条件1] マッピング後のテクスチャに不連続な部分が発生しない。

[条件2] マッピング後のテクスチャに基本テクスチャパターンの単純な繰り返しが見えない。

### 2.2 単純シームレステクスチャ（トーラステクスチャ）

#### 2.2.1 テクスチャマッピングパターン

まずはじめに、 $L_x \times L_y$  画素の長方形の基本テクスチャパターンが与えられ、それを使用してマッピングを行う場合の並べ方を考える。図2.1に3つの例を示す。

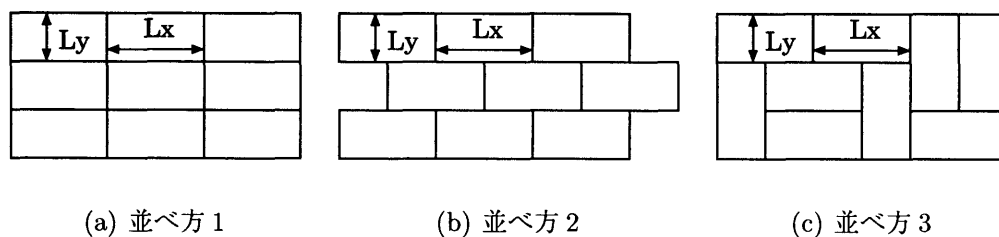
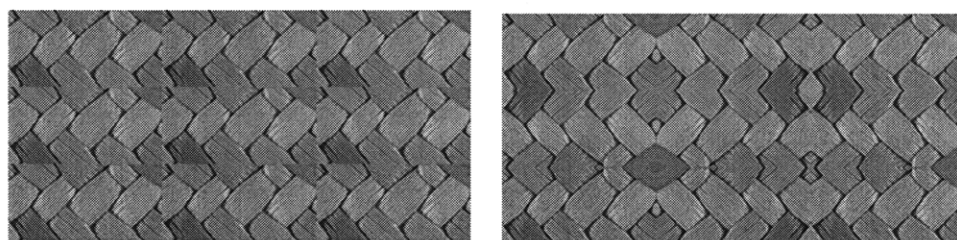


図 2.1: 基本テクスチャパターンの並べ方

図2.1(a) は与えられた基本テクスチャパターンを、縦方向と横方向に平行に並べ

た例である。この並べ方によるマッピングの例を図 2.2 に示す。図 2.2(a) を見ると、並べられたそれぞれの基本テクスチャパターンの境界が、マッピング後のテクスチャにおいて、境界線として現れていることが確認できる。この並べ方の場合、繰り返す際、基本テクスチャパターンを裏返してマッピングすることにより、境界の不連続感をかなり減ずることは出来る。この裏返しの操作を行った結果を図 2.2(b) に示す。しかしながら、この方法を用いても本質的な解決は見込めない。

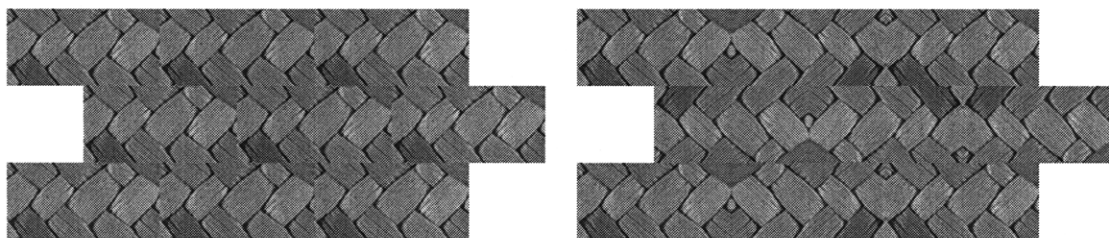


(a) 裏返しなし

(b) 裏返しあり

図 2.2: 並べ方 1 によるマッピング

図 2.1(b) は、与えられた基本テクスチャパターンを煉瓦のように、横方向に一段毎半分 ( $L_x/2$ ) ずつずらして並べたものである。この並べ方によるマッピングの例を図 2.3 に示す。図 2.3 を見ると、ただ並べた場合はもちろんのこと、裏返してマッピングを行っても縦方向の境界は緩和されるが、横方向の境界には全く効果がないことが分かる。



(a) 裏返しなし

(b) 裏返しあり

図 2.3: 並べ方 2 によるマッピング

図 2.1(c) は、与えられた基本テクスチャパターンを様々な方向に並べたものである。この並べ方によるマッピングの例を図 2.4 に示す。この並べ方に至っては、任意の基本テクスチャパターン間で境界線が更に顕著に現れることが分かる。

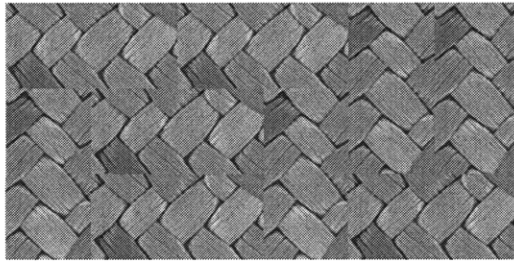


図 2.4: 並べ方3によるマッピング

### 2.2.2 境界条件

2.2.1節で任意のテクスチャパターンを基本テクスチャパターンとして並べた場合、基本テクスチャパターン間の境界で境界線が現れてしまうことを述べた。そこで、基本テクスチャパターンがどのような条件を満たせば、この境界線が現れないかを考える。まずはじめに図 2.1(a) の並べ方において、すべての基本テクスチャパターンが同一方向に並べられている場合について考えてみる。

この場合、一つの基本テクスチャパターンにおける境界の接続は、上の境界は一つ上の基本テクスチャパターンの下の境界と、下の境界は一つ下の基本テクスチャパターンの上の境界と、左の境界は一つ左の基本テクスチャパターンの右の境界と、右の境界は一つ右の基本テクスチャパターンの左の境界とそれぞれつながっている。同一の基本テクスチャパターンを繰り返して並べているので、この境界の接続のようは図 2.5 の矢印で示すようになっている。

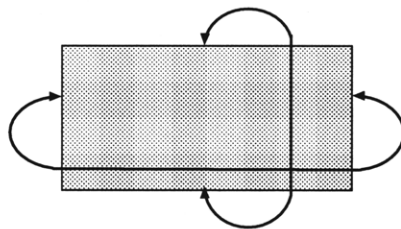


図 2.5: 境界の接続

この境界の接続の仕方から、境界上の画素が同じ値を持てば、濃度による不連続性は解消されることになる。これを擬似的に実現したのが、基本テクスチャパターンを裏返して並べる方法である。しかしながら、先にも述べたように裏返して並べる方法では本質的な解決は望めない。そこで、基本テクスチャパターンの境界で、境界線が現れないような境界における濃度の関係を考えてみる。そのような境界における濃度は図 2.6 のようになれば良い。すなわち、図 2.6 において実線、点線の矢印



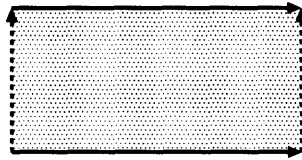


図 2.6: 境界の濃度

はそれぞれ同一の濃度を持つものである。図 2.6 の関係を満たす、基本テクスチャパターンの配置を一種のトポロジーと見なすと、これは図 2.7 に示すようなトーラス

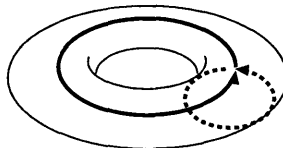


図 2.7: 基本テクスチャパターンのトポロジー

(ドーナツ) 形状を成していることがわかる。

このことから、図 2.1(a) の並べ方において、テクスチャ間で境界線が現れないためには、マッピングする基本テクスチャパターンがこのトーラス上に描かれたものと等価である必要があることがわかる。これ以降、本論文において、この条件を満たす基本テクスチャパターンのことをトーラステクスチャ、単純シームレステクスチャと呼ぶこととする。

先に述べた基本テクスチャパターンを裏返して並べる方法では本質的な解決が得られない理由を、この単純シームレステクスチャとの比較により説明する。図 2.8 と図 2.9 にそれぞれの方法によるマッピング前後のテクスチャを示す。

図 2.8(b) からわかるように、裏返しを行った場合、確かに境界部分での濃度は同じになるので、濃度的にはつながることとなる。しかしながら、基本テクスチャパターン内の基本パターン (図 2.8 では長方形) の境界への入射角が二つのテクスチャ間で鏡面反射した状態になってしまう。そのため基本パターンを損なうことになり、連続性を阻害することになる。これが、本質的な解決にならない理由である。一方、図 2.9(b) に示す単純シームレステクスチャは、図 2.9(a) に示しているトーラス上でつながっているテクスチャと同相であり、境界部分においての濃度はもちろんのこと、パターンの境界への入射角も同じになる。その結果、図 2.9(c) から分かるように連続性が保たれることとなる。

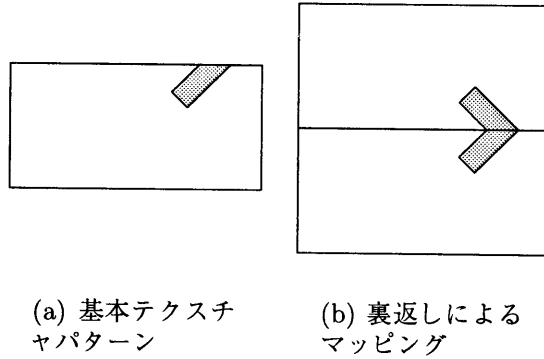


図 2.8: 裏返しによる改善

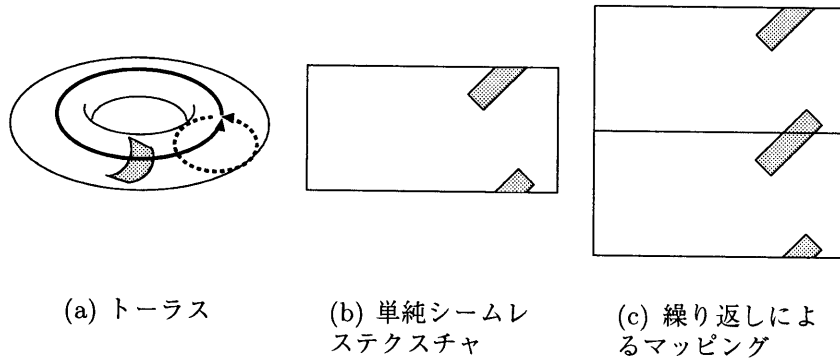


図 2.9: 単純シームレステクスチャによる改善

次に、図 2.1(b) の並べ方においてすべてのテクスチャが同一方向に並べられている場合について、図 2.1(a) の場合と同様に考えてみる。図 2.1(b) の場合の境界の接続と濃度は図 2.10 のようになる。この場合の基本テクスチャパターンについてもトポロジーを考えてみる。すると、この基本テクスチャパターンのトポロジーも図 2.11 に示すように図 2.1(a) の場合と同様にトーラス形状を成していることがわかる。つまり、図 2.1(b) を満たすテクスチャも単純シームレステクスチャであり、図 2.1(a) との違いはそのトーラスの切断位置によるものであることがわかる。

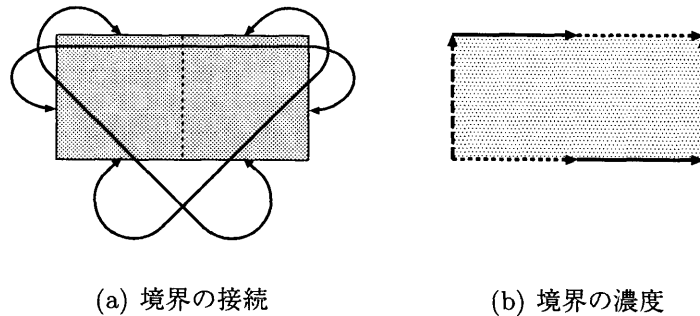


図 2.10: 図 2.1(b) の場合の境界の接続と濃度

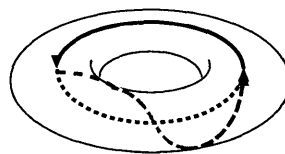


図 2.11: テクスチャのトポロジー

### 2.2.3 ランダム書き込み法による単純シームレステクスチャの生成

ここまで図 2.1(a) 及び (b) の並べ方で境界連続性を保つテクスチャは単純シームレステクスチャであれば良いことを述べてきた。次にこの単純シームレステクスチャの生成法について述べる。

単純シームレステクスチャは上下、左右のみがつながる基本テクスチャパターンでそのトポロジーがトーラスと同相であること満たすことによって実現できる。これまで、単純シームレステクスチャの説明にトポロジーを用いてきたが、実際に単純シームレステクスチャを生成する場合、トーラス上への生成は計算の複雑さ、トーラスから平面への展開時の座標位置のゆがみ等の問題があり、実用的ではない。そこで、図 2.6 や図 2.10(b) の平面上に基本パターンを複数貼り付ける方法を利用した方法（以後、ランダム書き込み法と呼ぶこととする）により単純シームレステクスチャを生成する。

#### <ランダム書き込み法>

ランダム書き込み法は、基本パターンを乱数を利用して基本テクスチャパターンに書き込むことにより単純シームレステクスチャを生成する方法である。この方法を順を追って説明する。単純シームレステクスチャの大きさは  $L_x \times L_y$  画素とし、画素位置を  $(x, y) (0 \leq x < L_x, 0 \leq y < L_y)$  で示す。書き込み基本パターンが式 (2.1)

で表される円である場合を用いて説明する.

$$\begin{aligned} \text{circle}_i(x, y) &< 0 \\ \text{circle}_i(x, y) &= (x - x_i)^2 + (y - y_i)^2 - r_i^2 \end{aligned} \quad (2.1)$$

- $(x_i, y_i)$  : 書き込み基本パターンの中心の画素位置  
 $r_i$  : 書き込み基本パターンの半径  
 $x, y, x_i, y_i$  : 整数  
 $r_i$  : 実数

手順1: 乱数により円の中心  $(x_i, y_i)$  および半径  $r_i$  を設定する. ただし, これらの値は以下の条件を満たす値とする.

$$\begin{aligned} 0 &\leq x_i < L_x \\ 0 &\leq y_i < L_y \\ r_i &< L_x/2 \quad \text{if } L_x < L_y \\ r_i &< L_y/2 \quad \text{if } L_x > L_y \end{aligned} \quad (2.2)$$

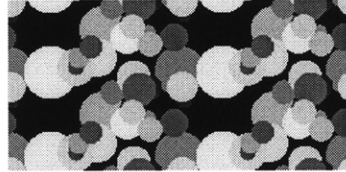
手順2: この円の中心および半径を用いて式 (2.1) で生成された円を書き込んでいく. 書き込み画素位置が単純シームレステクスチャの領域内の場合にはそのまま書き込みを行う. 書き込み画素位置が領域外にある場合には, 座標位置を領域内の画素に次の式 (2.3) により再割当てした後, 書き込みを行う.

$$\begin{aligned} x_{\text{after}} &= \begin{cases} x_{\text{before}} - L_x & \text{if } x_{\text{before}} \geq L_x \\ x_{\text{before}} & \text{if } 0 \leq x_{\text{before}} < L_x \\ x_{\text{before}} + L_x & \text{if } x_{\text{before}} < 0 \end{cases} \\ y_{\text{after}} &= \begin{cases} y_{\text{before}} - L_y & \text{if } y_{\text{before}} \geq L_y \\ y_{\text{before}} & \text{if } 0 \leq y_{\text{before}} < L_y \\ y_{\text{before}} + L_y & \text{if } y_{\text{before}} < 0 \end{cases} \end{aligned} \quad (2.3)$$

- $(x_{\text{before}}, y_{\text{before}})$  : 再割当て前の書き込み画素位置  
 $(x_{\text{after}}, y_{\text{after}})$  : 再割当て後の書き込み画素位置

手順3: 手順1,2を必要回数繰り返す.

この方法により生成した単純シームレステクスチャの例を図2.12に示す.



(a) 単純シームレステクスチャ

(b) マッピング例

図 2.12: ランダム書き込み法による単純シームレステクスチャの例

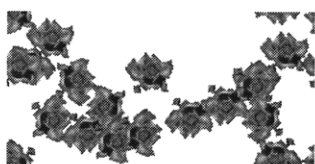
上述の方法において, 円の替りに画像を利用した例を図2.13, 図2.14, 図2.15に示す. 画像を貼り込む場合, 手順1で与えられる値は, 各書き込み基本パターンの中心位置  $(x_i, y_i)$ , およびその回転角  $\theta_i$  である. これらの例において単純シームレステクスチャの大きさは  $256 \times 128$  画素であり, 図2.13(a), および, 図2.15(a)の書き込み基本パターンの大きさは  $49 \times 39$  画素, 図2.14(a), および, 図2.15(b)の書き込み基本パターンの大きさは  $38 \times 38$  画素である. また回転角  $\theta_i$  は  $90$  度刻みで行っている. 図2.13は, 乱数の初期値の違いによるランダム書き込み法による単純シームレステクスチャの変化を示している. 図2.14は, 繰り返し回数による変化を示し, 図2.15においては複数の基本パターンを利用した場合の例を示している.

図2.10(b)を満たす単純シームレステクスチャを生成する場合には, 先に示した手順の2の式(2.3)を以下の式(2.4)に書き換えればよい.

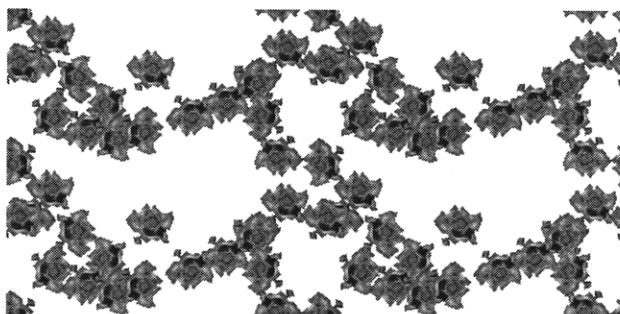
$$\begin{aligned}
 (x_{after}, y_{after}) = & \\
 \left\{ \begin{array}{ll}
 (x_{before} + L_x, y_{before}) & \text{if } x_{before} < 0, \quad 0 \leq y_{before} < L_y \\
 (x_{before}, y_{before}) & \text{if } 0 \leq x_{before} < L_x, \quad 0 \leq y_{before} < L_y \\
 (x_{before} - L_x, y_{before}) & \text{if } x_{before} \geq L_x, \quad 0 \leq y_{before} < L_y \\
 (x_{before} + L_x/2, y_{before} - L_y) & \text{if } x_{before} < L_x/2, \quad y_{before} \geq L_y \\
 (x_{before} - L_x/2, y_{before} - L_y) & \text{if } x_{before} \geq L_x/2, \quad y_{before} \geq L_y \\
 (x_{before} + L_x/2, y_{before} + L_y) & \text{if } x_{before} < L_x/2, \quad y_{before} < 0 \\
 (x_{before} - L_x/2, y_{before} + L_y) & \text{if } x_{before} \geq L_x/2, \quad y_{before} < 0
 \end{array} \right.
 \end{aligned}
 \tag{2.4}$$



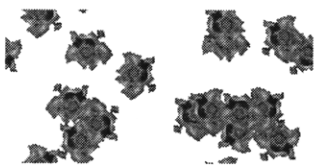
(a) 書き込み基本パターン



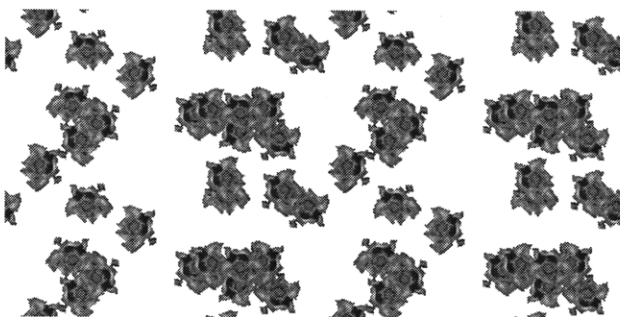
(b) 生成テクスチャ  
乱数初期値 1



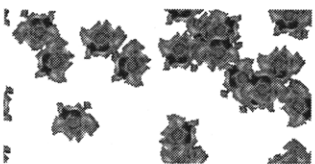
(c) (b) によるマッピング



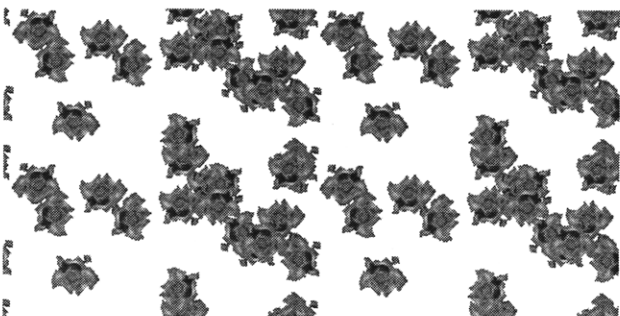
(d) 生成テクスチャ  
乱数初期値 2



(e) (d) によるマッピング



(f) 生成テクスチャ  
乱数初期値 3



(g) (f) によるマッピング

図 2.13: ランダム書き込み法による単純シームレステクスチャ(乱数による変化)

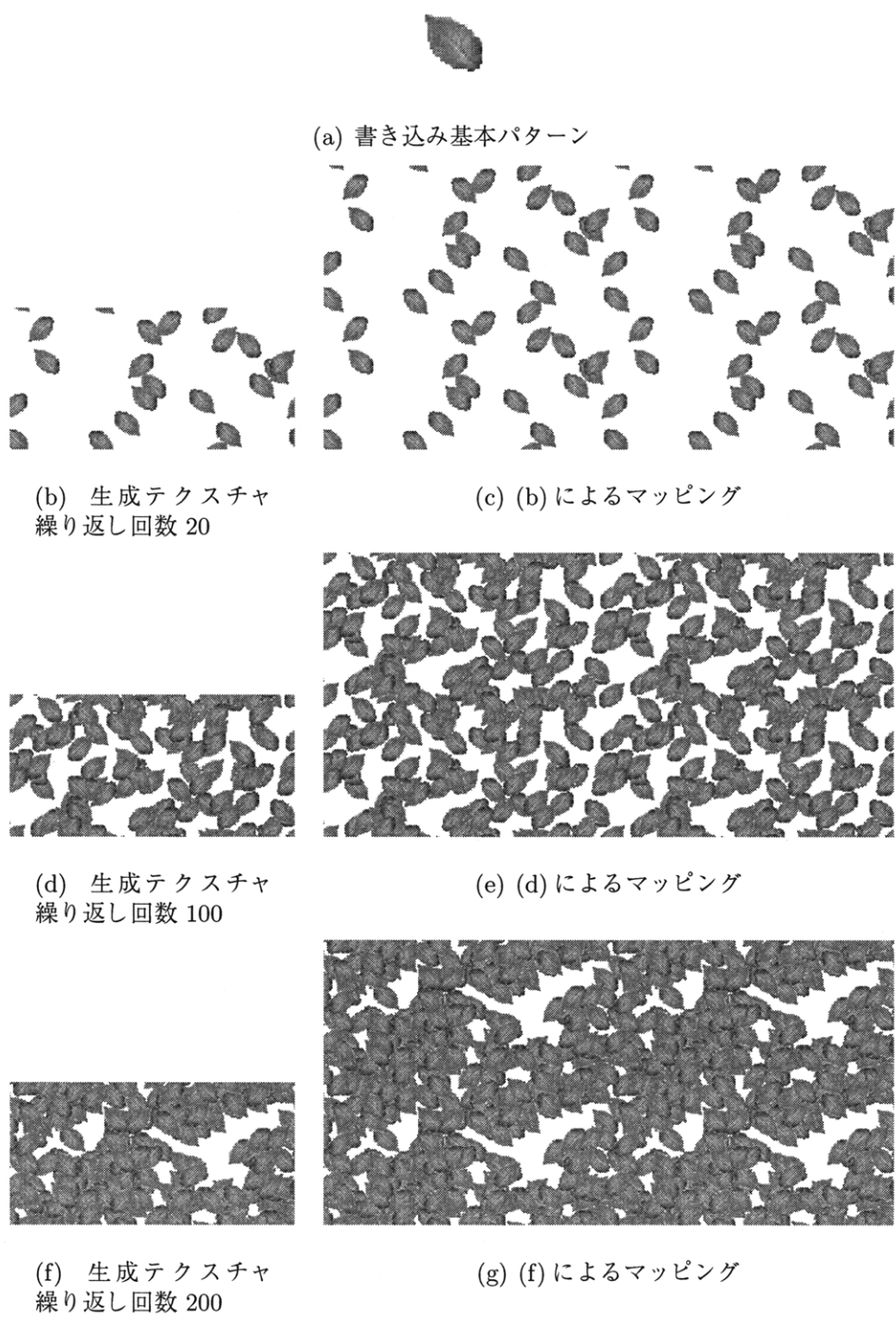


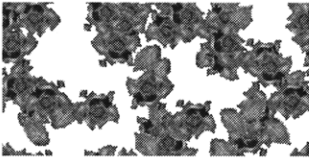
図 2.14: ランダム書き込み法によるテクスチャの例 (繰り返し回数による変化)



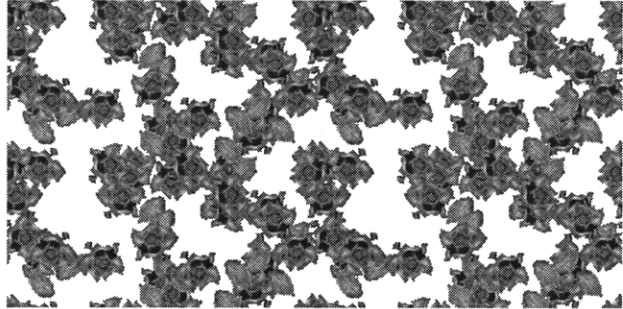
(a) 書き込み基本パターン 1



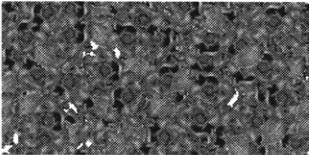
(b) 書き込み基本パターン 2



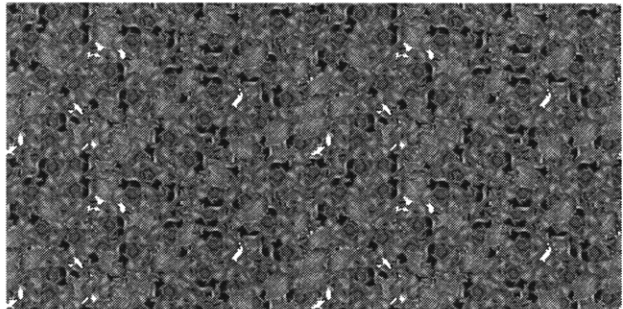
(c) 生成テクスチャ  
乱数初期値 4 繰り返し  
回数 50



(d) (c) によるマッピング



(e) 生成テクスチャ  
乱数初期値 77 繰り返し  
回数 300



(f) (e) によるマッピング

図 2.15: ランダム書き込み法による単純シームレステクスチャ(複数の書き込み基本パターンの使用)



- $(x_{before}, y_{before})$  : 再割当て前の書き込み画素位置  
 $(x_{after}, y_{after})$  : 再割当て後の書き込み画素位置

また、この例では、横方向に  $L_x/2$  だけ移動させてつなげてあるが、式(2.3)の代りに式(2.5)を利用すれば、横方向に  $L_{xoffset+}$  だけ移動させてつなげた単純シームレステクスチャの生成が可能である。

$$\begin{aligned}
 (x_{after}, y_{after}) = & \\
 \left\{ \begin{array}{ll}
 (x_{before} + L_x, y_{before}) & \text{if } x_{before} < 0, \quad 0 \leq y_{before} < L_y \\
 (x_{before}, y_{before}) & \text{if } 0 \leq x_{before} < L_x, \quad 0 \leq y_{before} < L_y \\
 (x_{before} - L_x, y_{before}) & \text{if } x_{before} \geq L_x, \quad 0 \leq y_{before} < L_y \\
 (x_{before} + L_{xoffset-}, y_{before} - L_y) & \text{if } x_{before} < L_{xoffset+}, \quad y_{before} \geq L_y \\
 (x_{before} - L_{xoffset+}, y_{before} - L_y) & \text{if } x_{before} \geq L_{xoffset+}, \quad y_{before} \geq L_y \\
 (x_{before} + L_{xoffset+}, y_{before} + L_y) & \text{if } x_{before} < L_{xoffset-}, \quad y_{before} < 0 \\
 (x_{before} - L_{xoffset-}, y_{before} + L_y) & \text{if } x_{before} \geq L_{xoffset-}, \quad y_{before} < 0
 \end{array} \right.
 \end{aligned}
 \tag{2.5}$$

- $(x_{before}, y_{before})$  : 再割当て前の書き込み画素位置  
 $(x_{after}, y_{after})$  : 再割当て後の書き込み画素位置  
 $L_{xoffset-}$  :  $L_x - L_{xoffset+}$

式(2.5)において  $L_{xoffset+} = L_x/2$  とすれば、式(2.3)と等価になり、 $L_{xoffset+} = 0$  とすれば、式(2.2)と等価になる。

図 2.16 にいくつかの例を示す。

## 2.2.4 単純シームレステクスチャの限界

これまで、単純シームレステクスチャは、基本テクスチャパターン間の境界で境界線が現れないという点において有効であること述べた。しかしながら、この単純シームレステクスチャでは、同一パターンが規則正しく並ぶので、本章の最初にあげた2つの条件のうちの条件2を満たすものではない。条件2を満たす方法として、次の二つの方法を考えることができる。

[方法1] 境界で連続し内部パターンの異なるテクスチャの複数生成

[方法2] 向きを変えて並べても境界で連続するテクスチャの生成

このうち方法1は単純シームレステクスチャの境界をまたがる部分を固定し、内部に再度テクスチャを生成することにより比較的簡単に実現できる。しかしながら、条件2の効果を視覚的に十分感じさせるためにはある程度のパターン数を必要とする。そこで方法2を検討する。しかしながら、単純シームレステクスチャはトーラスと同相な基本テクスチャパターンであり、任意の向きでの配置は構造的に不可能である。方法2は、図2.1(c)の並べ方を可能とする基本テクスチャパターンの生成を意味する。つまり、これは90度単位で任意の向きに接続可能な基本テクスチャパターンであり、単純シームレステクスチャよりも少ない枚数で条件2を実現できる有効な基本テクスチャパターンである。次の2.3で具体的実現方法について述べる。

## 2.3 一般化シームレステクスチャ

### 2.3.1 テクスチャマッピングパターン

単純シームレステクスチャの場合と同様に、テクスチャマッピングにおける並べ方を考える。ここでは、簡単のため、基本テクスチャパターンの形状は正方形とする。図2.17にマッピングの例を示す。

Texture	Texture	Texture	Texture
Texture	Texture	Texture	Texture

図 2.17: マッピングの例

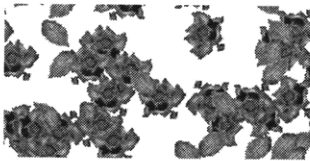
図2.17では同一の基本テクスチャパターンを、90度単位で任意の向きを変えながらマッピングを行っている。このようなマッピングが可能な基本テクスチャパター



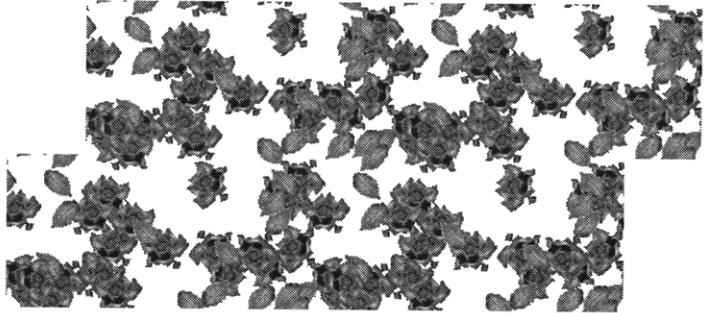
(a) 書き込み基本パターン 1



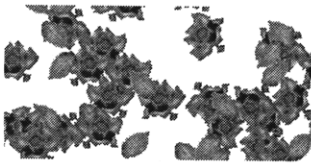
(b) 書き込み基本パターン 2



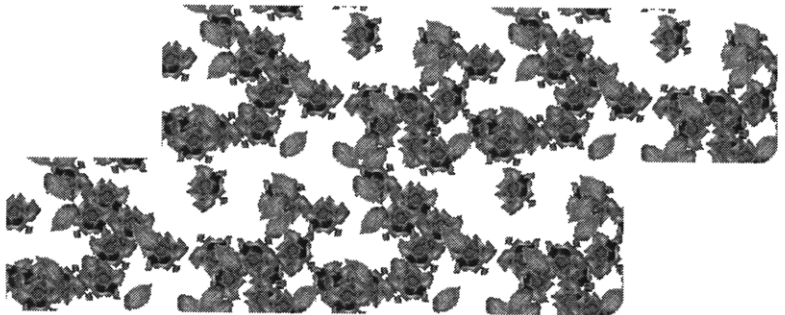
(c) 生成テクスチャ  
 $offset = L_x/4$



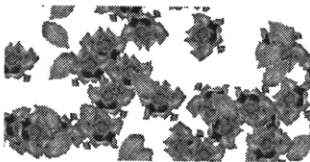
(d) (c) によるマッピング



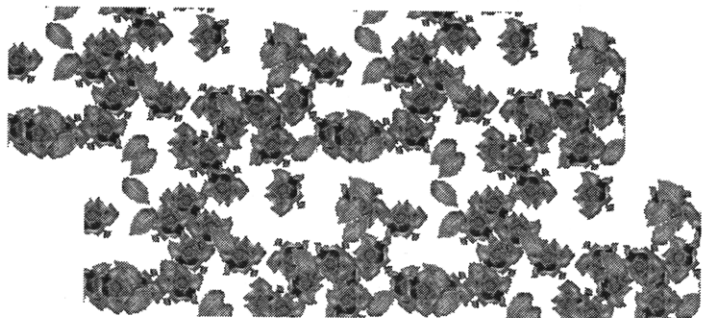
(e) 生成テクスチャ  
 $offset = L_x/2$



(f) (e) によるマッピング



(g) 生成テクスチャ  
 $offset = 3L_x/4$



(h) (g) によるマッピング

図 2.16: ランダム書き込み法による単純シームレステクスチャ(シフトの量による変化)

ンのことを，一般化シームレステクスチャと呼ぶこととする．今後，特に限定しない限りシームレステクスチャとは，この一般化シームレステクスチャを示すものとする．

### 2.3.2 境界条件

図 2.17 のシームレステクスチャの境界条件を考える．その境界の接続と濃度は図 2.18 のようになる．図 2.18(a) に，右境界線を例にとってシームレステクスチャの境

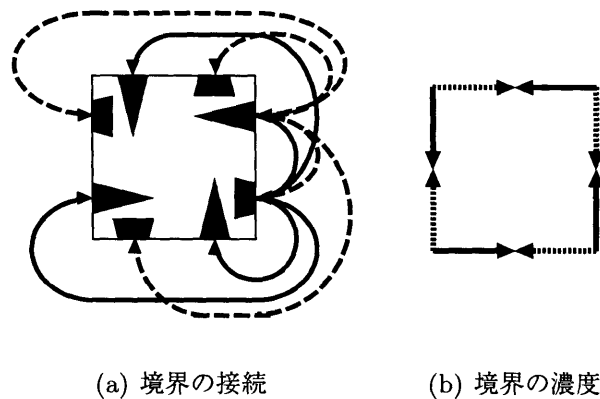


図 2.18: シームレステクスチャの境界の接続と濃度

界の接続を示している．図 2.18(b) に，境界の濃度を示している．図 2.18(b) において，4 本の実線の矢印，および，4 本の点線の矢印は同一の濃度を持つが，図 2.18(a) の関係から実線と点線の 2 種類の線を使い示している．接続関係を考えた場合，実線部分は必ず点線部分と接続し，その逆に点線部分は必ず実線部分と接続するようになっている．図 2.18 の関係から，このテクスチャのトポロジーを考えると，概念的には図 2.19 のようなものとなる．ここで概念的とことわったのは，シームレステクスチャのトポロジーと図 2.19 が正確には同じでないからである．この図 2.19 の中心にある矢印部分は境界の同一部分であるが，シームレステクスチャの図 2.18 に示した濃度関係は満足しているが，実線と点線の接続関係は正確に示されていない．図 2.20 に中心部分の関係を示す．

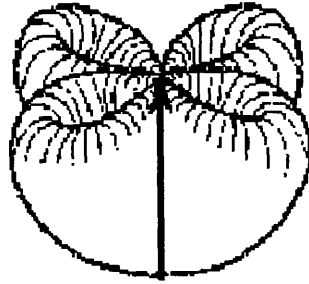


図 2.19: シームレステクスチャのトポロジーの概念図

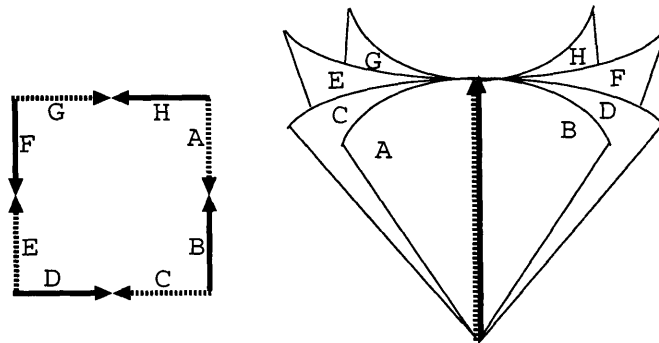


図 2.20: シームレステクスチャのトポロジー中心部

2.3.3節,2.3.4節でシームレステクスチャの生成法として, 単純シームレステクスチャでも使用したランダム書き込み法とフラクタルの midpoint displacement method を利用した方法(以後, フラクタル法)の二つを説明する.

### 2.3.3 ランダム書き込み法によるシームレステクスチャの生成

シームレステクスチャは上下, 左右に加えて90度単位の回転が可能なテクスチャで図2.18で示した平面上にテクスチャパターンを描くことにより実現することができる. 単純シームレステクスチャの場合と同様に, 書き込み基本パターンは式(2.1)で表される円を用いて書き込み手順を追って説明する. シームレステクスチャの大きさは  $L_x \times L_y (L_x = L_y)$  画素とする.

$$\begin{aligned} circle_i(x, y) &< 0 \\ circle_i(x, y) &= (x - x_i)^2 + (y - y_i)^2 - r_i^2 \end{aligned} \tag{2.1}$$

- $(x_i, y_i)$  : 書き込み基本パターンの中心の画素位置  
 $r_i$  : 書き込み基本パターンの半径  
 $x, y, x_i, y_i$  : 整数  
 $r_i$  : 実数

### <ランダム書き込み法>

手順1: 乱数により円の中心  $(x_i, y_i)$  および半径  $r_i$  を設定する。ただし、これらの値は以下の条件を満たす値とする。

$$\begin{aligned} 0 &\leq x_i < L_x \\ 0 &\leq y_i < L_y \\ r_i &< L_x/2 \end{aligned} \tag{2.6}$$

手順2: この円の中心および半径を用いて式 (2.1) で生成された円をシームレステクスチャに書き込んでいく。その際に生成された円がシームレステクスチャの領域を越えるかどうかを判定する。

$$\begin{aligned} 0 &\leq x_i + r_i < L_x \\ 0 &\leq x_i - r_i < L_x \\ 0 &\leq y_i + r_i < L_y \\ 0 &\leq y_i - r_i < L_y \end{aligned} \tag{2.7}$$

生成された円が式 (2.7) を満たす場合はそのまま書き込みを行い、一画素でも式 (2.7) を満たさない場合には生成された円を次の式によりそれぞれ4点に再割当てし、書き込みを行う。

$$\begin{aligned}
(x_{after_1}, y_{after_1}) &= \\
\left\{ \begin{array}{ll}
(x_{before}, y_{before}) & \text{if } 0 \leq x_{before} < L_x, \quad 0 \leq y_{before} < L_y \\
(x_{before} + L_x, y_{before}) & \text{if } x_{before} < 0, \quad 0 \leq y_{before} < L_y \\
(x_{before} - L_x, y_{before}) & \text{if } x_{before} \geq L_x, \quad 0 \leq y_{before} < L_y \\
(x_{before}, y_{before} + L_y) & \text{if } 0 \leq x_{before} < L_x, \quad y_{before} < 0 \\
(x_{before}, y_{before} - L_y) & \text{if } 0 \leq x_{before} < L_x, \quad y_{before} \geq L_y \\
(x_{before} + L_x, y_{before} + L_y) & \text{if } x_{before} < 0, \quad y_{before} < 0 \\
(x_{before} + L_x, y_{before} - L_y) & \text{if } x_{before} < 0, \quad y_{before} \geq L_y \\
(x_{before} - L_x, y_{before} + L_y) & \text{if } x_{before} \geq L_x, \quad y_{before} < 0 \\
(x_{before} - L_x, y_{before} - L_y) & \text{if } x_{before} \geq L_x, \quad y_{before} \geq L_y
\end{array} \right. \\
(x_{after_2}, y_{after_2}) &= (L_x - x_{after_1}, L_y - y_{after_1}) \\
(x_{after_3}, y_{after_3}) &= (y_{after_1}, L_x - x_{after_1}) \\
(x_{after_4}, y_{after_4}) &= (L_y - y_{after_1}, x_{after_1}) \tag{2.8}
\end{aligned}$$

$(x_{before}, y_{before})$  : 再割当て前の書き込み画素位置  
 $(x_{after_n}, y_{after_n})$  :  $n(n = 1 \sim 4)$  個目の再割当て後の書き込み画素位置

手順 3: 手順 1,2 を必要回数繰り返す.

これにより、ランダム書き込み法によるシームレステクスチャが生成される。しかしながら、この手順をそのまま使った場合、二つの問題が生じる。一つの問題は、境界部分では内部と比較してテクスチャの書き込み頻度が 4 倍となってしまうことであるが、この問題は、内部の書き込み指定頻度を 4 倍にすることにより比較的簡単に回避することができる。より問題となるの次の問題である。図 2.18 の 4 頂点、つまり、 $(0, 0), (L_x - 1, 0), (0, L_y - 1), (L_x - 1, L_y - 1)$  の近傍が書き込み位置に選ばれた場合に起こる。具体的にこの問題について説明する。

書き込み基本パターンが二つの境界を同時にまたがった位置に書き込まれたとする。図 2.21 の円で囲まれた部分の構成を図 2.21 右の円内に記述する。この図 2.21 右から、4 頂点近傍のテクスチャが書き込み基本パターンの 3 つの部分の合成であることがわかる。この図 2.21 のシームレステクスチャをマッピングした例を図 2.22 に示す。

図 2.22 からわかるように、この場合には書き込み基本パターンが対症的に重なりあった大きなテクスチャが規則的に並ぶこととなる。この大きなテクスチャは、4 頂

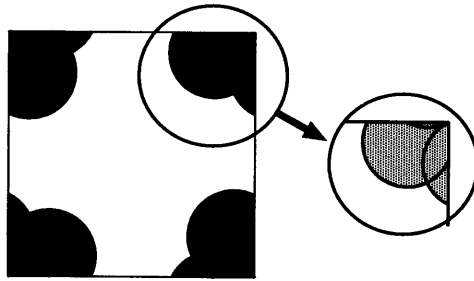


図 2.21: 二つ境界を同時にまたがる書き込み基本パターン

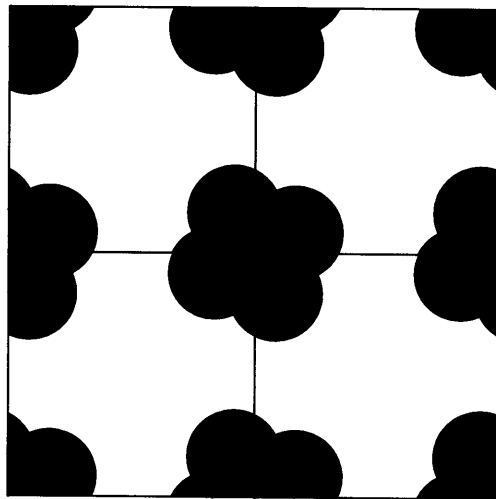


図 2.22: 図 2.21 によるマッピングテクスチャ

点を含む部分にあるために、回転操作を行っても必ず規則的に現れてしまう。これは、条件2を満たさない。この問題の回避のため、先に示した手順1に条件として式(2.9)を加え、二つの境界を同時にまたがらないように書き込み位置を制限する。

$$\left\{ \begin{array}{ll} 0 \leq x_i \pm r_i < L_x & 0 \leq y_i \pm r_i < L_y \\ x_i + r_i \geq L_x & 0 \leq y_i \pm r_i < L_y \\ x_i - r_i < 0 & 0 \leq y_i \pm r_i < L_y \\ 0 \leq x_i \pm r_i < L_x & y_i + r_i \geq L_y \\ 0 \leq x_i \pm r_i < L_x & y_i - r_i < 0 \end{array} \right.$$

(2.9)



これらの条件を満たした手順により生成したシームレステクスチャの例を図 2.23 に示す。

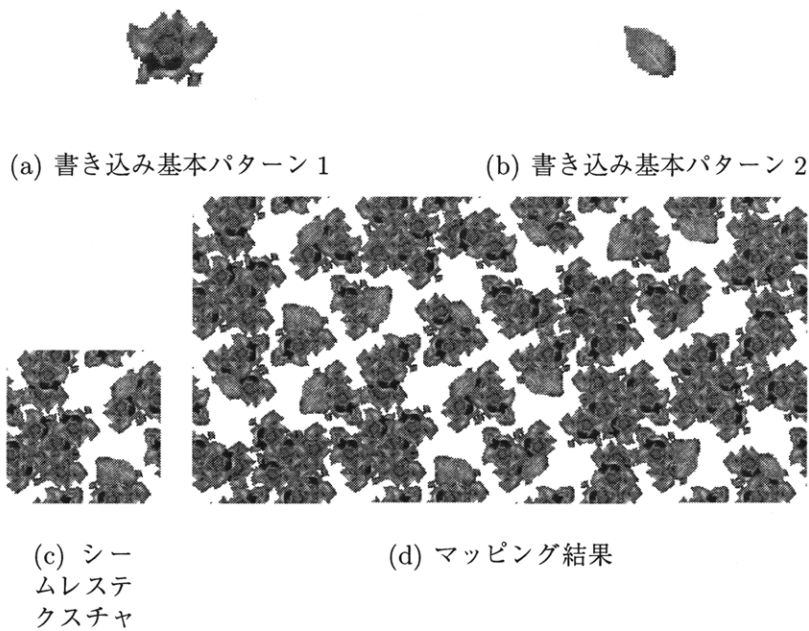


図 2.23: ランダム書き込み法によるシームレステクスチャ

### 2.3.4 フラクタル法によるシームレステクスチャの生成

フラクタル生成法を利用したシームレステクスチャの生成法を説明する。中点変位法によるフラクタル生成法では三角形の4分割法が一般的である。三角形の4分割法を図2.24に示す。

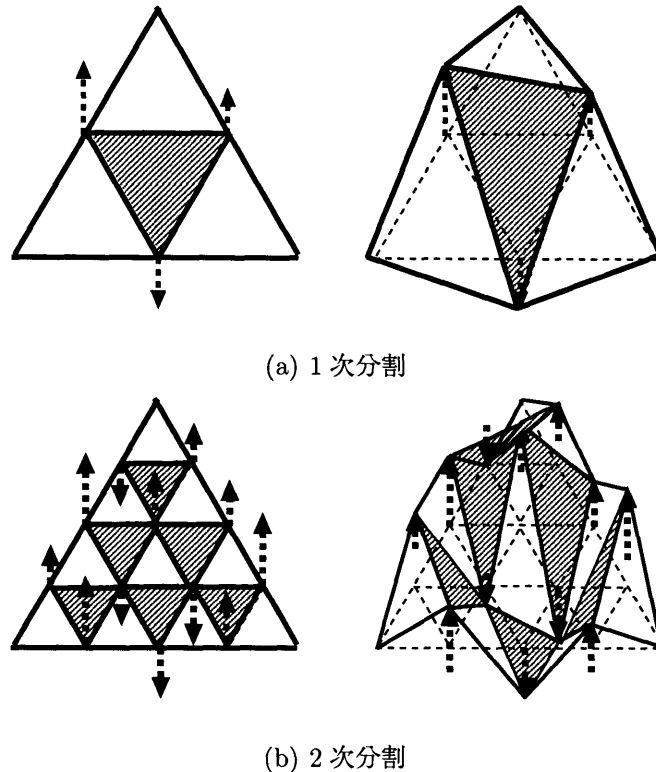
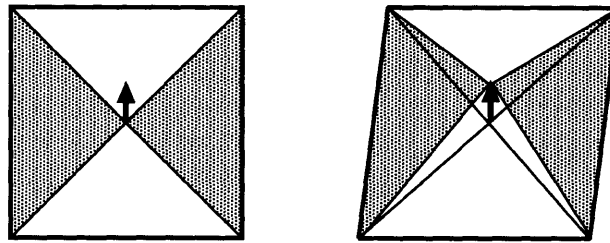


図 2.24: 三角形の4分割法によるフラクタル生成

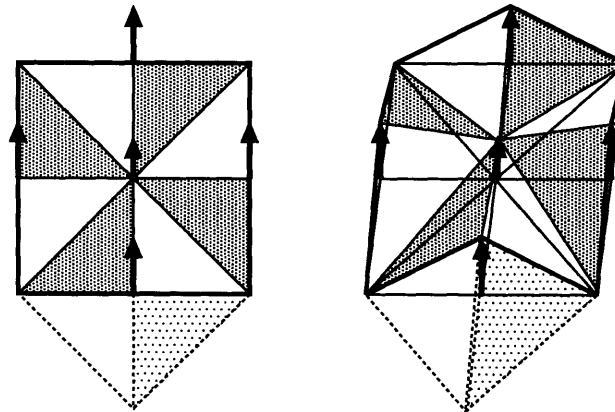
正三角形の4分割法は図2.24に示すように、各辺の中点の値に適当な減衰率パラメータ $r$ に乱数をかけた結果を加算した後、その中点同士を結び、生成された線分の中点に同様な操作を行い、これを繰り返すことによって、正三角形内部にテクスチャを生成していく方法である。図2.24においては2次分割までを例示している。減衰率パラメータ $r$ とは、中点の変位値を制御するパラメータで、分割次数毎に0~1.0の値を取る。減衰率は、1.0から減衰率パラメータの値をひいたものである。減衰率パラメータの値が大きい場合には、生成された平面の各画素間での濃度変位は大きく荒い質感を持つものとなり、反対に小さい場合には、生成された平面の各画素間での濃度変位は小さくなめらかな質感を持つものとなる。

この正三角形の4分割法を、図2.18(b)の条件に適用する容易さの観点から、フラ

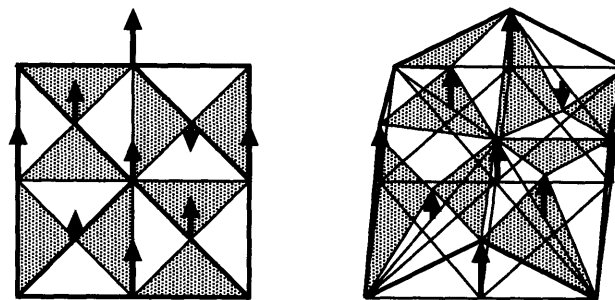
クタル生成法を図 2.25 に示す方法に変更して用いる。



(a) 1次分割



(b) 2次分割



(c) 3次分割

図 2.25: フラクタル生成

この方法では，図 2.25(a) に示すように正方形に対角線を引き，その交点に適当な減衰率パラメータ  $r$  に乱数をかけた結果を加算する．次にこの対角線の  $1/2$  を一辺とする菱形に対角線を引き，その対角線の一方を選択し，その選択した対角線の端

点の値の平均に適当な減衰率パラメータ  $r$  に乱数かけた結果を加算し対角線の交点の値とする。この時、図 2.25(b) のように、菱形の頂点の 1 つが生成を行う領域の外にある場合、つまり、算出点が境界上の時には、図 2.18(b) の条件から、その点に対応する領域内の点の値をその点の値として用いて値を算出する。図 2.25(b) の場合には、1 次分割で算出した点に対応点となる。また、同時に図 2.18(b) の条件から、同じ値を持つべき点にも算出された値を入れておく。3 次分割では、2 次分割での対角線の  $1/2$  を一辺とするそれぞれの正方形に対して 1 次分割の場合と同じ操作を行う。この操作を繰り返し行いフラクタルを生成する。

次にこの方法を用いてシームレステクスチャを生成する手順を示す。シームレステクスチャの大きさは  $L_x \times L_y$  ( $L_x = L_y = 2^n n$  : 自然数) 画素とし、画素位置を  $(x, y)$  ( $0 \leq x < L_x, 0 \leq y < L_y$ ) で表し、 $(x, y)$  の画素値を  $f(x, y)$  で表す。フラクタル法の計算の便宜上、 $x = L_x$ 、または、 $y = L_y$  とする画素が現れた場合には、 $x = L_x$  は、 $x = 0$  に、 $y = L_y$  は、 $y = 0$  に置き換える。

### <フラクタル法>

生成の際に用いる乱数の初期値および各分割回数における減衰率パラメータ  $r_m$  ( $m$  は分割回数) を与える。すなわち

$$\begin{aligned} 0.0 \leq r_m \leq 1.0 \\ m = 1, 2, \dots, 2n \end{aligned} \quad (2.10)$$

分割回数が偶数の場合 ( $m = 2k, 1 \leq k \leq n$ ) と奇数の場合 ( $m = 2k - 1, 1 \leq k \leq n$ ) に分け生成法の説明を行う。

#### a) 奇数の場合

分割回数が  $2k - 1$  のとき画素値がすでに算出されている画素中の 4 点により構成される 1 辺の長さが  $L_x/2^{k-1}$  画素で構成されるすべての正方形において、それぞれの正方形の対角線の交点位置の画素値を算出する。図 2.26 に 5 次分割の場合を示す。この図において●はすでに算出されている画素を示し、○は画素値の算出対象画素を示している。

図 2.26 内のある 1 つ正方形の対角線の交点位置  $(x_c, y_c)$  の画素値  $f(x_c, y_c)$  を算出する。この正方形は  $(x_c + L_x/2^k, y_c + L_x/2^k)$ ,  $(x_c + L_x/2^k, y_c - L_x/2^k)$ ,  $(x_c - L_x/2^k, y_c - L_x/2^k)$ ,  $(x_c - L_x/2^k, y_c + L_x/2^k)$  の 4 点で構成されている。まず、乱数により正方形の対角線のうちの 1 つを選択する。その選択された対角線の両端の値の平均値を算出しその値に減衰率パラメータ  $r_{2k-1}$  に乱数  $rand$  を乗じた値を加算した値を交点位置  $(x_c, y_c)$  の画素値  $f(x_c, y_c)$  とする。

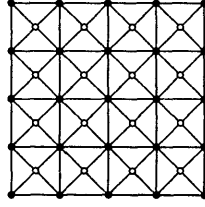


図 2.26: 5 次分割の場合の算出画素値位置

$$f(x_c, y_c) = \begin{cases} \frac{f(x_c+L_x/2^k, y_c+L_x/2^k) + f(x_c-L_x/2^k, y_c-L_x/2^k)}{2} + r_{2k-1} * rand \\ \text{or} \\ \frac{f(x_c+L_x/2^k, y_c-L_x/2^k) + f(x_c-L_x/2^k, y_c+L_x/2^k)}{2} + r_{2k-1} * rand \end{cases} \quad (2.11)$$

但し,  $k = 1$ , つまり 1 次分割の場合は, 式 (2.11) の計算を行う前に,  $(0, 0)$  に初期値を与える.

### b) 偶数の場合

分割次数が  $2k$  のとき画素値がすでに算出されている画素中の 4 点により構成される 1 辺の長さが  $\sqrt{2}(L_x/2^k)$  画素で構成されるすべての菱形において, それぞれの菱形の対角線の交点位置の画素値を算出する. 図 2.27 に 4 次分割の場合を示す. 図において, ● はすでに算出されている画素を示し, ○ は画素値のシームレステクスチャの領域内の算出対象画素を示し, + は領域外の点を示している.

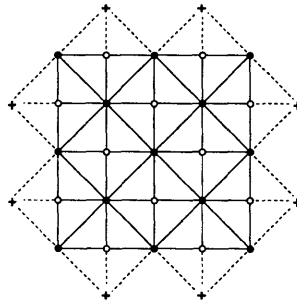


図 2.27: 4 次分割の場合の算出画素値位置

図 2.26 内のある 1 つの菱形の対角線の交点位置  $(x_c, y_c)$  を算出する。この菱形は  $(x_c + L_x/2^k, y_c)$ ,  $(x_c - L_x/2^k, y_c)$ ,  $(x_c, y_c + L_x/2^k)$ ,  $(x_c, y_c - L_x/2^k)$  の 4 点で構成されている。対角線の交点位置  $(x_c, y_c)$  により、算出方法を変える。

[case:1] 交点位置  $(x_c, y_c)$  が境界線上でない場合

乱数により菱形の対角線のうちの 1 つを選択する。その選択された対角線の両端の値の平均値を算出しその値に減衰率パラメータ  $r_{2k}$  に乱数  $rand$  を乗じた値を加算した値を交点位置  $(x_c, y_c)$  の画素値  $f(x_c, y_c)$  とする。

$$f(x_c, y_c) = \begin{cases} \frac{f(x_c + L_x/2^k, y_c) + f(x_c - L_x/2^k, y_c)}{2} + r_{2k} * rand \\ \text{or} \\ \frac{f(x_c, y_c + L_x/2^k) + f(x_c, y_c - L_x/2^k)}{2} + r_{2k} * rand \end{cases} \quad (2.12)$$

[case:2] 交点位置  $(x_c, y_c)$  が境界線上の場合

境界線上の点は図 2.18(b) の関係から正方形の各頂点から等距離にある 8 点は同じ点を共有する。従って、境界線上は算出対象となっている画素の算出は、その算出画素ともっとも近い正方形の頂点との距離と各頂点から同じ距離をもつ画素値を同時に算出することとなる。乱数により菱形の対角線のうちの 1 つを選択する。選択された対角線が境界線に沿っている場合には対角線の両端の値の平均値を算出しその値に減衰率パラメータ  $r_{2k}$  に乱数  $rand$  を乗じた値を加算した値を交点位置  $(x_c, y_c)$  の画素値  $f(x_c, y_c)$  とし、共有画素の値も同様に  $f(x_c, y_c)$  とする。

$$f(x_c, y_c) = \begin{cases} \frac{f(x_c + L_x/2^k, y_c) + f(x_c - L_x/2^k, y_c)}{2} + r_{2k} * rand \\ \text{or} \\ \frac{f(x_c, y_c + L_x/2^k) + f(x_c, y_c - L_x/2^k)}{2} + r_{2k} * rand \end{cases} \quad (2.13)$$

選択された対角線が境界線と直行している場合には、対象画素と共有点を交点位置とする対角線の端点のうち、シームレステクスチャの領域内の境界から距離  $L_x/2^k$  にある 8 点を抜き出し、この 8 点を関連する対象画素または共有点が、図 2.18(b) において点線上にある 4 点と実線上にある 4 点とに分ける。分けられたそれぞれの 4 点から 2 点づつを任意に選び、選ばれた 4 点の平均  $average_4$  に減衰率パラメータ  $r_{2k}$  に乱数  $rand$  を乗じた値を加算し中心の画素値とし、共有画素の値も同様に  $f(x_c, y_c)$  とする。

$$f(x_c, y_c) = average_4 + r_{2k} * rand \quad (2.14)$$

分割次数が偶数の場合と奇数の場合を交互に，シームレステクスチャの範囲画素内のすべての画素値の算出が終わるまで計算を繰り返す．すべての画素値の算出が終わったら，シームレステクスチャの範囲の画素値の最大  $max$  と最小  $min$  を算出し，式 (2.15) により正規化を行いテクスチャの生成を終了する．

$$f_{val}(x, y) = (f_{calc}(x, y) - min) \frac{Range}{(max - min)} \quad (2.15)$$

- $f_{val}(x, y)$  : 正規化後の値
- $f_{calc}(x, y)$  : 正規化前の値
- $Range$  : 正規化後の濃度階調数
- $max, min$  : シームレステクスチャの画素値の最大値，最小値

ここで分割数が偶数の場合の2の後者で4点の平均を用いた理由を説明する．

まず，対角線の端点のシームレステクスチャの領域内の1点と対応する残り3点ともう一方の端点に対応する4点を利用し平均値を求めた場合を考える．シームレス性の観点からは8点を使うのがもっとも良いといえる．しかしながら，境界線付近の画素の対角線の交点画素値算出に8点を用いるため，内部と比較して画素値が平滑化される．そのため，生成テクスチャの中央付近と境界付近において質感に差が現れることとなる．

次に，対角線の端点のシームレステクスチャの領域内の1点と対応する残り3点のうち1点ともう一方の端点に対応する4点のうち1点を利用し平均値を求めた場合を考える．この場合，8点を用いる場合のような生成テクスチャの中央付近と境界付近において質感に差は現れないが，選択された2点以外の画素値を無視することとなり，シームレス性の観点からは必ずしも良いとはいえない．

そこでその両者の中間を取り，4点を利用することとした．4点を利用する方法がもっとも良いと断言はできないが，4点を取った場合，算出画素値への元の画素値の影響は内部の画素と比較して1/2となり，算出に使われる確率は内部と同等になるという効果もある．

図2.28にフラクタルで生成したシームレステクスチャとマッピング例を示す．テクスチャのサイズは  $128 \times 128 (= 2^7)$ ，使用した乱数  $rand$  の値域は  $-2^{15} \leq rand < 2^{15}$ ，分割次数14で生成された結果を256階調に正規化を行っている．

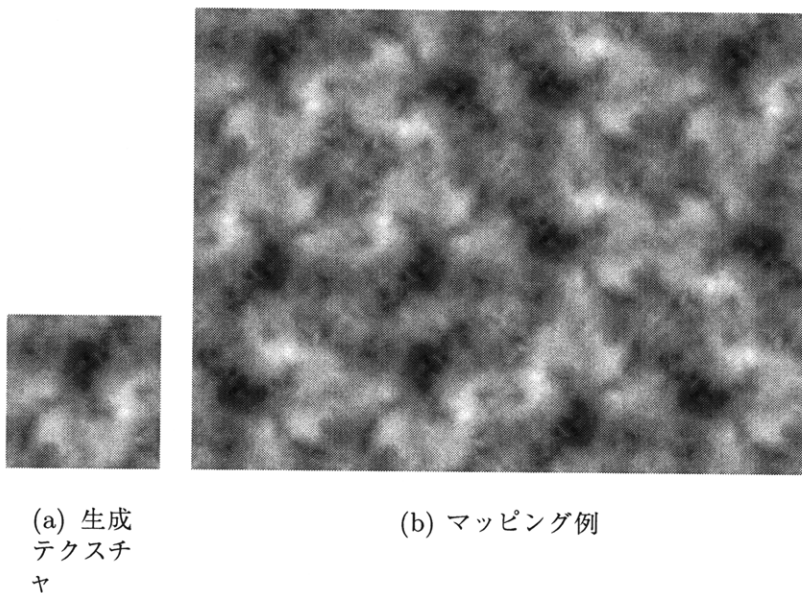


図 2.28: フラクタル法によるシームレステクスチャの例

### 2.3.5 ランダム書き込み法とフラクタル法の相違

前節において、シームレステクスチャの生成法として、ランダム書き込み法とフラクタル法を示した。ランダム書き込み法は、あらかじめ用意されている画像を利用して、シームレステクスチャの生成を行うので、ある程度生成結果の予想が可能である。それに対して、フラクタル法は、乱数を用いてシームレステクスチャの生成を行うので、生成結果の予測は不可能であるが、その代わりに、生成されるシームレステクスチャは変化に富むものとなる。

乱数の初期値と減衰率パラメータをかえて生成したフラクタル法によるシームレステクスチャの画像を、図 2.30, 図 2.31, 図 2.32, 図 2.33, 図 2.34, 図 2.35 に示す。なお、生成テクスチャの形状が分かりやすいように、256 階調の濃度値を図 2.29 を用いて色付けを行ってある。



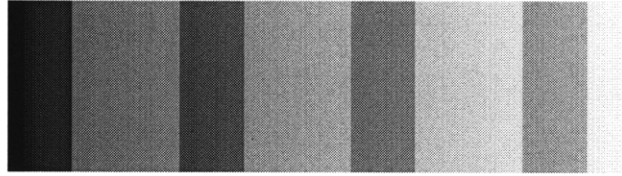


図 2.29: 色付け濃度マップ

図 2.30, 図 2.31, における減衰率パラメータを表 2.1 に示す.

表 2.1: 図 2.30, 図 2.31 の減衰率パラメータ

分割次数	減衰率パラメータ
1	1.000000
2	1.000000
3	0.424264
4	0.300000
5	0.127279
6	0.090000
7	0.038184
8	0.027000
9	0.011455
10	0.008100
11	0.003437
12	0.002430
13	0.001031
14	0.000729

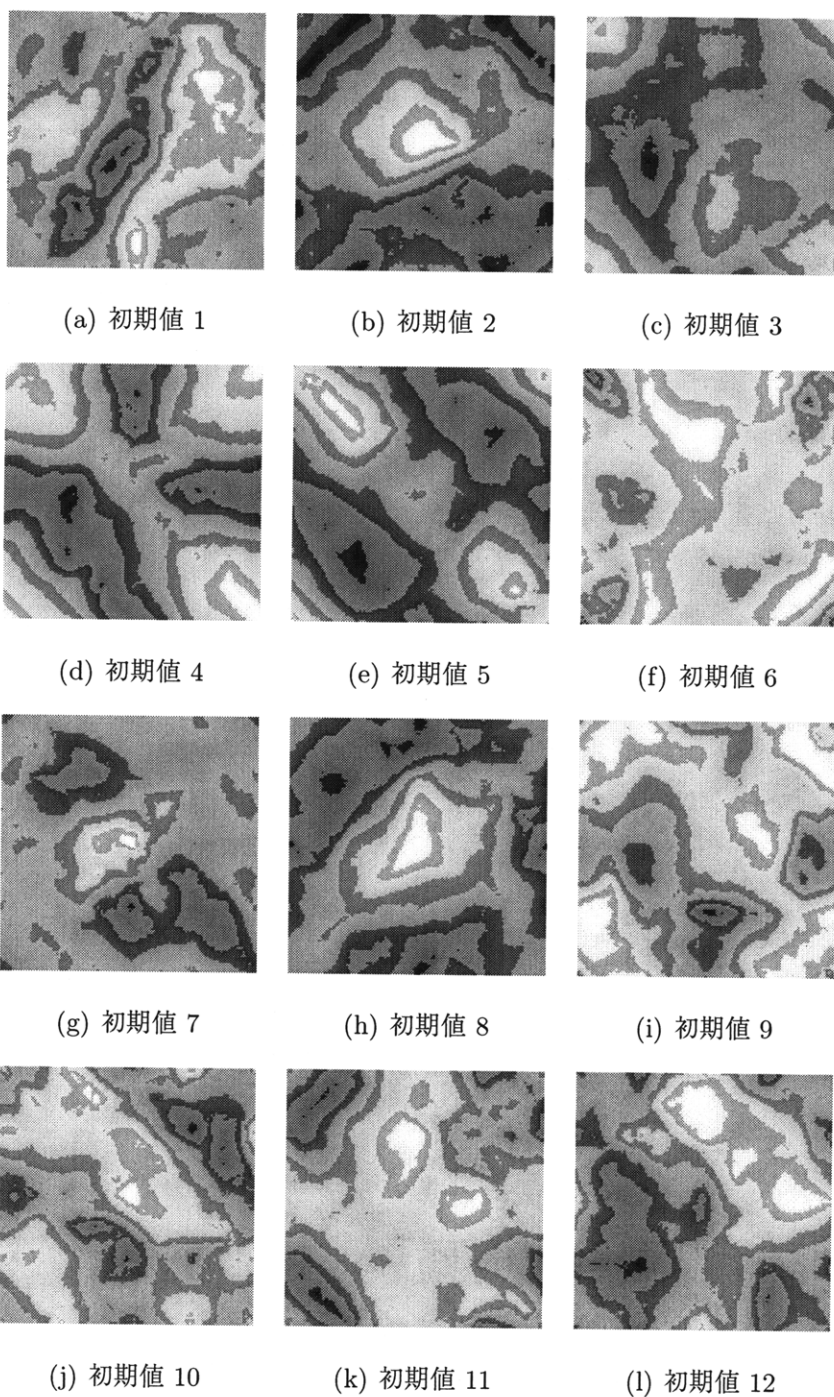


图 2.30: 乱数初期值 1-12

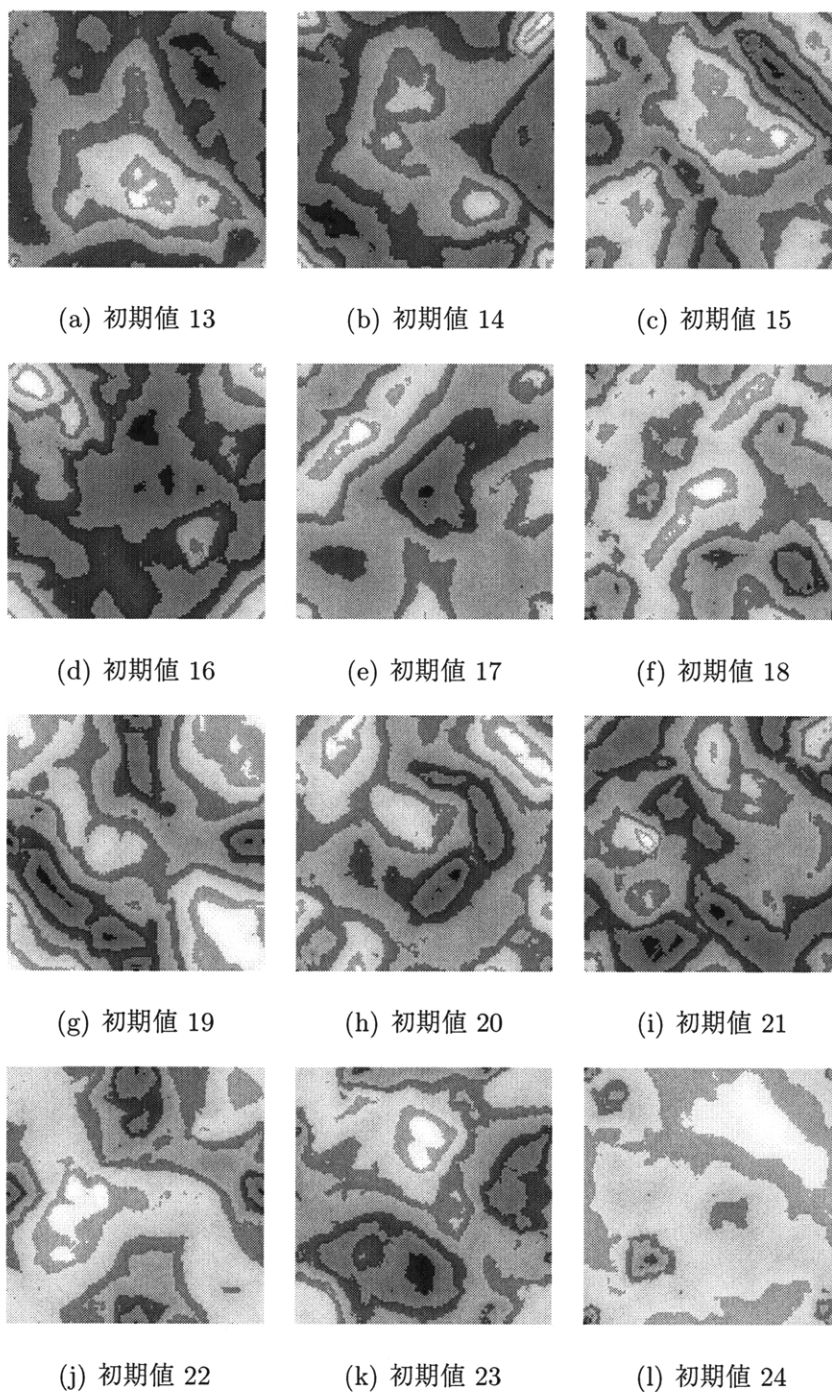


图 2.31: 乱数初期值 13-24

図 2.32, 図 2.33 における減衰率パラメータを表 2.2 に示す.

表 2.2: 図 2.32, 図 2.33 の減衰率パラメータ

分割次数	減衰率パラメータ
1	1.000000
2	1.000000
3	0.7507107
4	0.500000
5	0.353553
6	0.250000
7	0.176777
8	0.125000
9	0.088388
10	0.062500
11	0.044194
12	0.031250
13	0.022097
14	0.015625

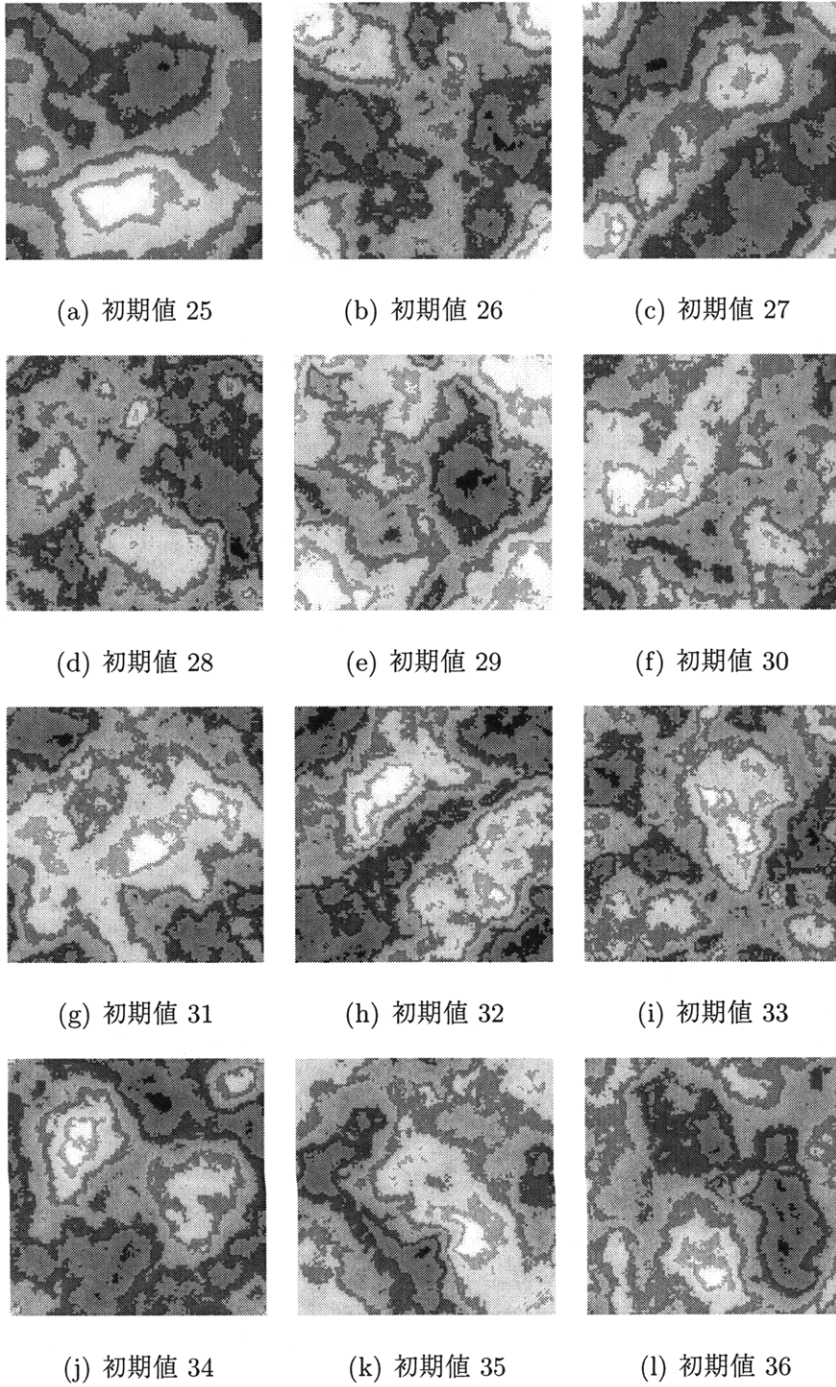


图 2.32: 乱数初期值 25-36

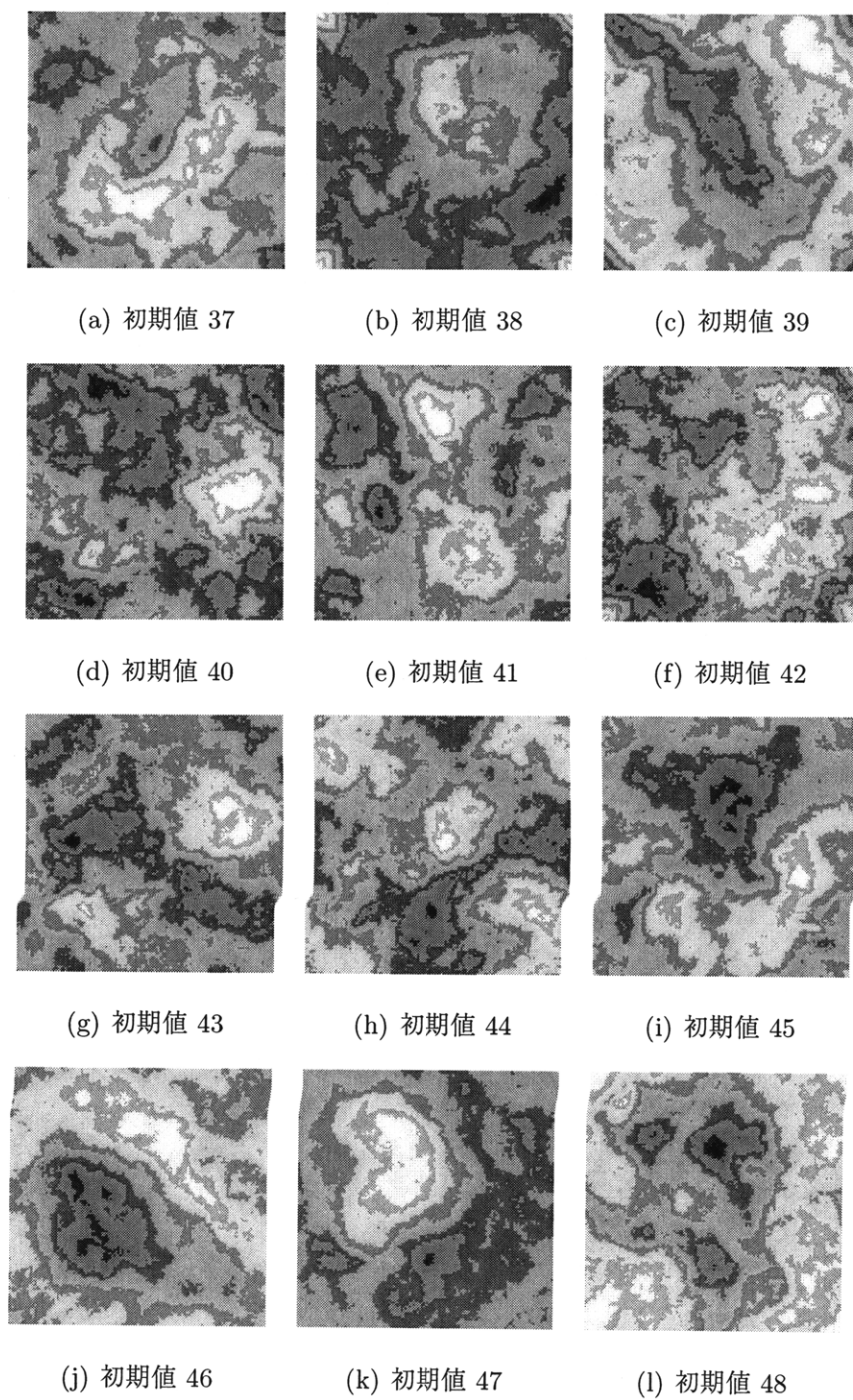


图 2.33: 乱数初期值 37-48

図 2.34, 図 2.35, における減衰率パラメータを表 2.3 に示す.

表 2.3: 図 2.34, 図 2.35 の減衰率パラメータ

分割次数	減衰率パラメータ
1	1.000000
2	1.000000
3	0.989949
4	0.700000
5	0.692965
6	0.490000
7	0.485075
8	0.343000
9	0.339553
10	0.240100
11	0.237687
12	0.168070
13	0.166381
14	0.117649

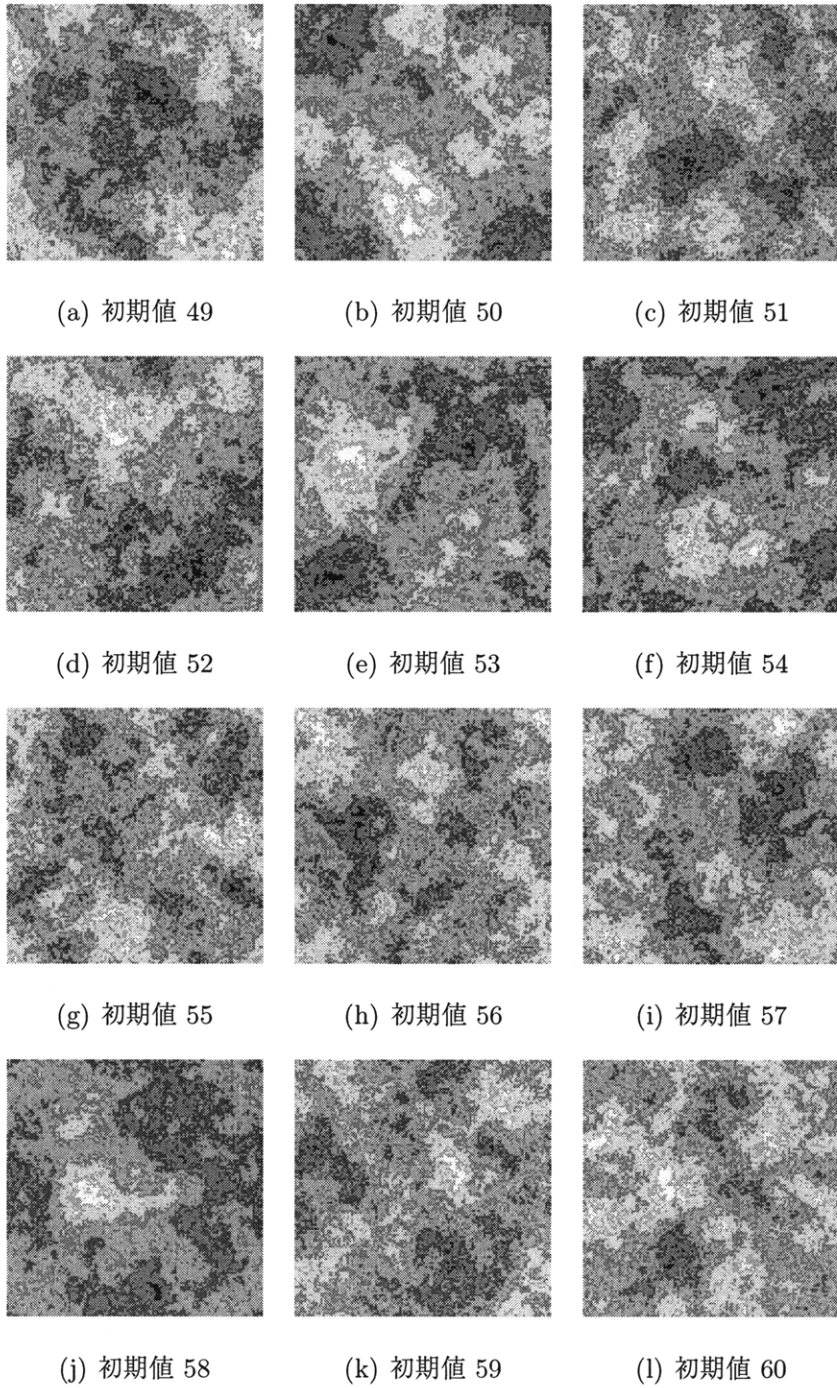


图 2.34: 乱数初期值 49-60



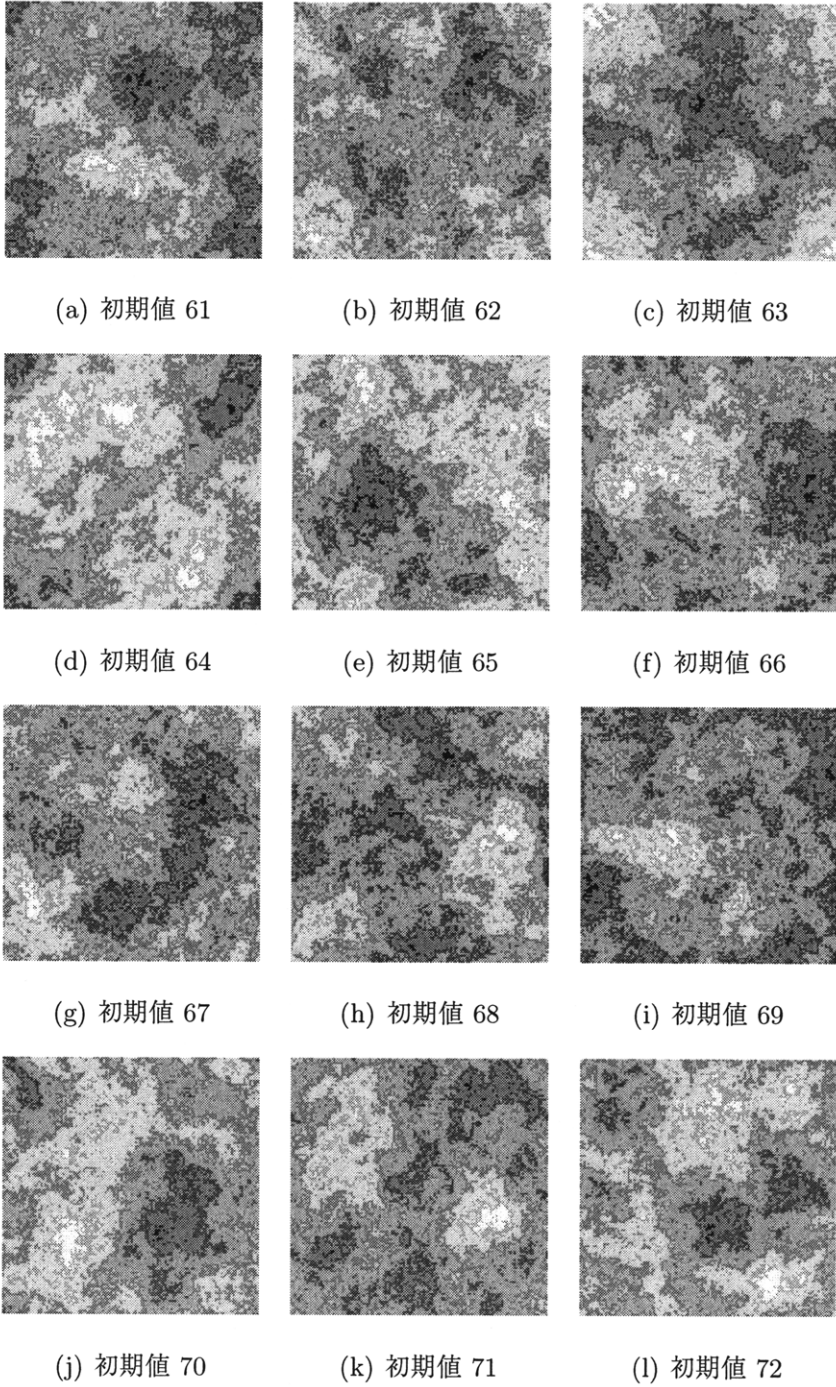
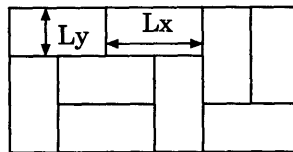


图 2.35: 乱数初期值 61-72

### 2.3.6 正方形から長方形への拡張

前節では簡単のため、基本テクスチャパターンの形状を正方形として説明を行ってきた。ここで、図 2.1(c) で示した並べ方が可能な長方形の基本テクスチャパターンについて考える。この並べ方が可能なテクスチャの境界の濃度条件は図 2.36 のようになる。又、図 2.36 は図 2.37 のように考えることができる。このことから、これまでのべてきたシームレステクスチャを利用して図 2.1(c) の並べ方を実現することができることがわかる。



(c)

図 2.1(c) テクスチャマッピングパターンの例 (再掲)

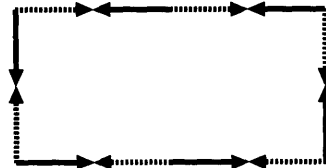


図 2.36: 図 2.1(c) が可能なテクスチャの濃度条件

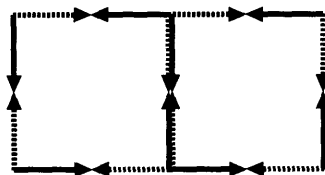


図 2.37: 図 2.36 の解釈

次に、図 2.37 のような解釈をせずに図 2.36 の平面にテクスチャを生成する場合を考える。この平面へのシームレステクスチャの生成法の拡張は図 2.37 から容易に類推可能である。正方形のシームレステクスチャの場合、4 辺が共通だったのが、図 2.36 の場合、6 辺が共通になっている。従って、境界をまたがる場合に行う操作を 4 回から 6 回に変更すれば、図 2.36 の平面に図 2.1(c) の並べ方が可能なテクスチャの生成ができることになる。

以下に、ランダム書き込み法、フラクタル法、それぞれの手順を示す。

### <ランダム書き込み法>

シームレステクスチャの大きさは  $L_x \times L_y$  ( $L_x = 2L_y$ ) 画素とする。

手順 1: 乱数により円の中心  $(x_i, y_i)$  および半径  $r_i$  を設定する。ただし、これらの値は以下の条件を満たす値とする。

$$\begin{aligned} 0 \leq x_i < L_x \\ 0 \leq y_i < L_y \\ r_i < L_y/2 \end{aligned} \tag{2.16}$$

手順 2: この円の中心および半径を用いて式 (2.1) で生成された円を書き込んでいく。その際に生成された円が境界を越えるかどうかを判定する。

$$\begin{aligned} 0 \leq x_i + r_i < L_x \\ 0 \leq x_i - r_i < L_x \\ 0 \leq y_i + r_i < L_y \\ 0 \leq y_i - r_i < L_y \end{aligned} \tag{2.7}$$

生成された円のすべての画素が式 (2.7) を満たす場合はそのまま書き込みを行い、一画素でも式 (2.7) を満たさない場合には生成された円を次の式によりそれぞれ 6 点に再割当てし、書き込みを行う。

$$(x_{after_1}, y_{after_1}) = \begin{cases} (x_{before}, y_{before}) & \text{if } 0 \leq x_{before} < L_x, 0 \leq y_{before} < L_y \\ (x_{before} + L_x, y_{before}) & \text{if } x_{before} < 0, 0 \leq y_{before} < L_y \\ (x_{before} - L_x, y_{before}) & \text{if } x_{before} \geq L_x, 0 \leq y_{before} < L_y \\ (x_{before}, y_{before} + L_y) & \text{if } 0 \leq x_{before} < L_x, y_{before} < 0 \\ (x_{before}, y_{before} - L_y) & \text{if } 0 \leq x_{before} < L_x, y_{before} \geq L_y \\ (x_{before} + L_x, y_{before} + L_y) & \text{if } x_{before} < 0, y_{before} < 0 \\ (x_{before} + L_x, y_{before} - L_y) & \text{if } x_{before} < 0, y_{before} \geq L_y \\ (x_{before} - L_x, y_{before} + L_y) & \text{if } x_{before} \geq L_x, y_{before} < 0 \\ (x_{before} - L_x, y_{before} - L_y) & \text{if } x_{before} \geq L_x, y_{before} \geq L_y \end{cases}$$

$$(x_{after_2}, y_{after_2}) = (L_x - x_{after_1}, L_y - y_{after_1})$$

$$(x_{after_3}, y_{after_3}) = \begin{cases} (x_{after_1} + L_y, y_{after_1}) & \text{if } x_{after_1} < L_y \\ (x_{after_1} - L_y, y_{after_1}) & \text{if } x_{after_1} \geq L_y \end{cases}$$

$$(x_{after_4}, y_{after_4}) = \begin{cases} (x_{after_2} + L_y, y_{after_2}) & \text{if } x_{after_2} < L_y \\ (x_{after_2} - L_y, y_{after_2}) & \text{if } x_{after_2} \geq L_y \end{cases}$$

$$(x_{after_5}, y_{after_5}) = \begin{cases} (y_{after_1}, L_y - x_{after_1}) & \text{if } x_{after_1} < L_y, y_{after_1} < L_y/2 \\ (y_{after_1}, x_{after_1} - L_y) & \text{if } x_{after_1} \geq L_y, y_{after_1} < L_y/2 \\ (L_y - y_{after_1}, L_y - x_{after_1}) & \text{if } x_{after_1} < L_y, y_{after_1} \geq L_y/2 \\ (L_y - y_{after_1}, x_{after_1} - L_y) & \text{if } x_{after_1} \geq L_y, y_{after_1} \geq L_y/2 \end{cases}$$

$$(x_{after_6}, y_{after_6}) = (L_x - x_{after_5}, L_y - y_{after_5})$$

$(x_{before}, y_{before})$  : 再割当て前の書き込み画素位置

$(x_{after_n}, y_{after_n})$  :  $n(n = 1 \sim 6)$  個目の再割当て後の書き込み画素位置

手順3: 手順1,2を必要回数繰り返す.

### <フラクタル法>

シームレステクスチャの大きさは  $L_x \times L_y$  ( $L_x = 2L_y = 2^{n+1}$   $n$ : 自然数) 画素とし, 画素位置を  $(x, y)$  ( $0 \leq x < L_x, 0 \leq y < L_y$ ) で表し,  $(x, y)$  の画素値を  $f(x, y)$  で

表すこととする。フラクタル法の計算の便宜上、 $x = L_x$ 、または、 $y = L_y$ とする画素が現れた場合には、 $x = L_x$ は、 $x = 0$ に、 $y = L_y$ は、 $y = 0$ に置き換えることとする。

### a) 奇数の場合

シームレステクスチャの領域を二つの正方形と考えそれぞれの正方形で正方形の場合と同様の処理を行う。

但し、1次分割の場合は、式(2.11)の計算を行う前に図2.38に示す共有する6点のうちの●で示されている2点に同一の値を初期値として与える操作を行う。

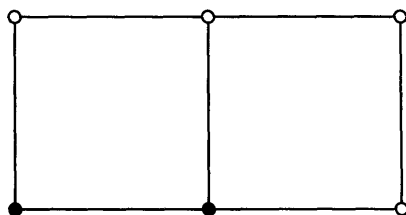


図 2.38: 初期値設定画素値位置

### b) 偶数の場合

[case:1] 交点位置  $(x_c, y_c)$  が境界線上でない場合

奇数の場合と同様に、それぞれの正方形で正方形の場合と同様の処理を行う。

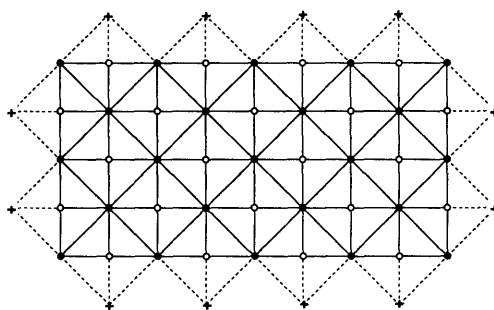


図 2.39: 4次分割の場合の算出画素値位置

[case:2] 交点位置  $(x_c, y_c)$  が境界線上の場合

境界線上の点は図2.36の関係から生成平面の図2.39の各点から等距離にある

12点は同じ値を共有する。従って、境界線上は算出対象となっている画素の算出は、その算出画素と生成平面の正方形の頂点のうちもっとも近いものとの距離と各頂点から同じ距離をもつ画素値を同時に算出することとなる。

乱数により菱形の対角線のうちの1つを選択する。選択された対角線が境界線に沿っている場合には正方形の場合と同様に処理を行う。

選択された対角線が境界線と直行している場合には、対象画素と同じ値を共有する点から生成平面内に境界に対して垂直で対角線の  $L_y/2^k$  の距離にある12点を図2.36において対角線が点線から出ている6点と実線から出ている6点とに分ける。分けられたそれぞれの6点から3点づつを任意に選び、選ばれた6点の平均  $average_6$  に減衰率パラメータ  $r_{2k}$  に乱数  $rand$  を乗じた値を加算し中心の画素値とし、共有画素の値も同様に  $f(x_c, y_c)$  とする。

$$f(x_c, y_c) = average_6 + r_{2k} * rand \quad (2.17)$$

分割次数が偶数の場合と奇数の場合を交互に、生成平面内のすべての画素値の算出が終わるまで計算を繰り返す。

## 2.4 むすび

シームレステクスチャの生成する方法を示した。シームレステクスチャは任意の向きで4角形領域を並べても境界が必ず連続すると言う特徴があり、シームレステクスチャを予め生成し、レンダリング時には向きをランダムに変えながらマッピングすることにより、以下のような効果がある。

- [効果1] 少量のマッピングデータを用いて、テクスチャの境界を感じさせない大面積のマッピングが可能である。
- [効果2] テクスチャパターンの単純な繰り返しが現れず、一枚の大面積のテクスチャデータをマッピングしたと同様の効果を持つ。
- [効果3] テクスチャ生成とレンダリングが別プロセスであるので、レンダリング時のオーバーヘッドがほとんどない。