

# 正方形粒子の最密充填手法を用いた石畳テクスチャの生成

宮田 一 乗<sup>†</sup> 伊藤 貴 之<sup>††</sup> 嶋田 憲 司<sup>†††</sup>

本論文では、正方形粒子の最密充填手法を応用した石畳テクスチャの生成手法を提案する。引力や斥力を持つ四角形粒子を、力学シミュレーションによって与えられた領域内部に最密充填して、敷石の敷き詰めパターンを生成する。その後、各四角形粒子を3次元の敷石形状に置き換えて、石畳テクスチャを生成する。続いて、敷石形状をサブディビジョンサーフェイス法で自動生成し、その表面にフラクタルノイズを用いて微細形状を付加する。本手法では、道路形状の境界形状と敷石の平均サイズなどを幾何データとして指定し、敷石の表面の粗さや色、光学属性などの属性データもあわせて指定する。提案手法では、従来の手法と比較して短い計算時間で、現実感のある石畳テクスチャを自動生成することが可能になった。

## Generating Pavement Textures via Close Packing of Squares

KAZUNORI MIYATA,<sup>†</sup> TAKAYUKI ITOH<sup>††</sup> and KENJI SHIMADA<sup>†††</sup>

This paper presents a method of generating pavement textures via close packing of squares. A pattern of packed square cells for a given area is generated by performing particle simulation with proximity-based forces. The pavement texture is then obtained by generating a stone shape for each cell with the subdivision surface method and then applying fractal noise to create a detailed surface geometry. In this method, the boundary shape of the pavement and the average size of the pavement stones are specified as input geometric data, along with attribute data such as the roughness, color, and optical attributes of the stones. The proposed method automatically generates a realistic looking pavement texture for an arbitrarily shaped pavement with much lower computational cost than previous methods.

### 1. はじめに

物体を特徴付ける要素には、大きく分けて、物体の形状と、物体表面の色情報や光学属性などの表面属性がある。一般的に、物体の表面属性の表現には、バンブマッピングなどのマッピング手法がとられている。

本論文で表現対象としている石畳を表現する場合、CGデザイナーが路面の敷石を1つ1つモデリングして表現することが理想ではあるが、大変な労力を要するため現実的ではない。一方、石畳の写真素材を路面にマッピングして表現することも可能であるが、以下のような問題が生ずる。

- 表面の起伏が激しいため、光源の位置やカメラアングルの相違による不整合が発生しやすい。

- 素材の画像と道路形状のアスペクト比の違いにより、石畳の敷石模様には継ぎ目が発生する。
- 素材の画像と道路形状の不一致により、道路の境界における隙間や敷石の切断が発生する。

本論文では、上記のような問題点を解決する1つの手法として、スクウェアパッキングと呼ばれる閉領域の充填パターン生成法を用いて、指定された閉領域を覆う石畳テクスチャを生成する手法を提案する。

本手法により、敷石をモデリングするような手間も要せず、表現したい道路形状の領域と数個のパラメータを設定することで、デザイナーが希望する路面を被覆する石畳テクスチャの自動生成が可能になった。

### 2. 研究の背景

#### 2.1 石畳パターンの特徴

図1に示すような実際の石畳を観察して、石畳パターンに対して以下のような傾向を仮定できる。

- 道路の進行方向に沿って、敷石が格子状に整列するように石畳が作成されていることが多い。
- おおむね等密度に敷石が詰まっている。

<sup>†</sup> 東京工芸大学芸術学部メディアアート表現学科  
Media Art, Tokyo Institute of Polytechnics

<sup>††</sup> 日本アイ・ビー・エム株式会社東京基礎研究所  
IBM Research, Tokyo Research Laboratory

<sup>†††</sup> カーネギーメロン大学工学部機械工学科  
Mechanical Engineering, Carnegie Mellon University



図1 実際の石畳の例

Fig.1 A collection of pavements.

これらの性質を考慮すると、

- 複雑なカーブを描く道路形状に対しても、自動的に等密度で粒子（パーティクル）を充填する、
- 指定された道路形状の領域に、うまく沿って整理するように、格子状に粒子を充填する、

という条件を満たす粒子モデルが石畳の表現に利用できると思われる。

筆者らは、すでに報告したスクウェアパッキングの手法<sup>1)</sup>が以上の条件を満たす手法であることから、敷石パターンの生成過程において適用することにした。一方、与えられた閉領域の分割法としては、たとえばポロノイ分割などが考えられるが、この分割法では一般に等方的な分割パターンの生成となり、以上で述べた目的を満たす手法ではない。等方的に敷石を敷き詰めた石畳テクスチャを生成したい場合は、以降述べるパッキングパターンの生成手法の部分、ポロノイ分割手法に置き換えればよい。すなわち、本論文では、敷石の“流れ”が感じられるような、敷石の充填に異方性を持った石畳テクスチャの生成を目的とする。

## 2.2 過去の関連研究

本論文の関連研究には、与えられた領域を分割して各領域に対してテクスチャを生成するタイリングテクスチャの生成法<sup>2)~4)</sup>、岩石をCGで表現する手法<sup>5),6)</sup>、パーティクルシステムを用いた物体表面の形状モデリング法（以下、Cellular Texture 法と呼ぶ<sup>7)</sup>）などがあげられる。以上の研究例のうち、Cellular Texture 法は本論文での提案手法に関連が深いので、以下ではこの手法について述べる。

Fleischer らは、鱗や棘などの物体表面の形状モデリング手法として Cellular Texture 法（以下 CT 法と略す）を提案した<sup>7)</sup>。CT 法では、セルプログラム

により性格付けされた粒子が、位置や方向性、周囲の粒子の影響などを考慮しながら、テクスチャを貼り付ける物体表面に配置される。その後、各粒子は棘や半球などの幾何形状に置換され、最終的なテクスチャが生成される。

本手法と CT 法とは、アルゴリズムの大筋では非常に類似している。特に、

- 各粒子がエネルギーポテンシャルを持つ、
- 各ポテンシャルは、隣接粒子との距離や方向性から算出される、
- ポテンシャルの総計が小さくなるように粒子の配置を求める、

という3点では、両手法の概念はほぼ一致している。

CT 法では、ポテンシャルの総計が最小化された状態を求めているが、一般的にエネルギー最小化問題は計算時間が大きくなることが多い。文献<sup>7)</sup>では、CT 法では計算時間が数時間かかる例があると述べている。

それに対して本手法では、粒子間の引力や斥力から運動方程式を求め、その反復計算の過程で粒子の移動量が小さくなった時点で計算を終了する。この解法は、エネルギー最小解ではなく極小解を求めることに相当するが、本論文の目的である「石畳の表現」に対しては、配置結果はつねに最小解である必要はない。むしろ CT 法よりも計算時間を短縮できる長所のほうが重要であると考え、筆者らは本手法に運動方程式を用いることにした。なお、本手法による粒子配置結果は、初期位置にある程度依存するが、視覚的な印象には大差ない。適切な初期位置の算出方法については、文献<sup>8)</sup>ですでに議論している。

また CT 法では、粒子を整列させるモデルは提案しているが、粒子を格子状に配列させるモデルを提案していない。この点においても、本手法に新規性があると考えられる。

## 3. 提案手法の構成

### 3.1 提案手法の概略

提案手法は、図2に示す3つの処理から構成される。

- (1) 充填パターン生成処理 スクウェアパッキング法を用いて、指定された道路形状に沿った敷石の充填パターンを生成する。
- (2) 敷石形状の生成処理 生成された充填パターンに従い、サブディビジョンサーフェイス法やフラクタルノイズ関数を用いて、敷石の形状および敷石表面のテクスチャを生成する。
- (3) レンダリング処理 生成された敷石のデータを、指定された光源データのもとにレンダリングし、

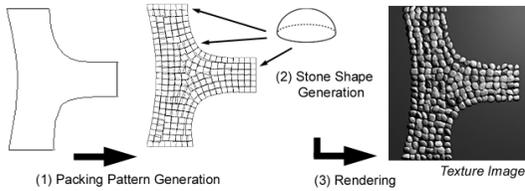


図2 提案手法の概略  
Fig. 2 Process overview.

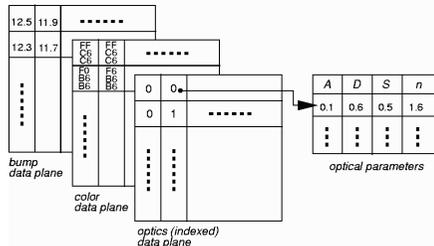


図3 テクスチャのデータ構造  
Fig. 3 Texture data structure.

最終的なテクスチャ画像を生成する。

### 3.2 テクスチャのデータ構造

本手法では、図3に示すように、テクスチャデータをバンプ属性、色属性、光学属性の3つの属性データプレーンに分類して、2次元マトリックスの各データセルに保持する。データプレーンの解像度は、生成されるテクスチャ画像の解像度と等しく設定する。

バンプ属性プレーンは、物体表面の凹凸状態を表し、底面からの変位量（高さ）をデータとして保持する。

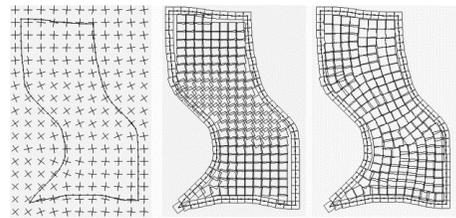
色属性プレーンは、物体表面の色を表す。色の表現モデルにはRGBモデルを用いている。

光学属性プレーンは、物体表面の光学属性を表す。光学属性は、物体を表現する際に用いるシェーディングモデルに依存している。本手法ではPhongのモデルを用いており、構成データとしては環境光の反射率（A）、拡散反射率（D）、鏡面反射率（S）、およびハイライト特性（n）の4つからなる。本手法では、1つの敷石に1つの光学属性を設定している。この場合、光学属性のデータをデータセルごとに保持するのは非効率的なので、光学属性データテーブルを参照するインデックス形式で保持する。

敷石を生成するごとに、各データをそれぞれの属性プレーンにスキャンコンバージョンし書き込む。

### 3.3 テクスチャデータのレンダリング法

テクスチャデータのレンダリングは、データセルごとに格納された情報を用いて行う。バンプ属性プレーンのデータは、シェーディング計算時の面の法線ベク



(a) Input data. (b) Initial location of particles. (c) Result of the packing process.

図4 スクウェアパッキングの実行例  
Fig. 4 An example of the output of the square packing method.

トルを算出する際に用いる。色属性および光学属性プレーンのデータは、レンダリング時の属性データとして用いる。

## 4. スクウェアパッキングについて

スクウェアパッキング<sup>1)</sup>とは、引力や斥力を持つ四角形粒子を、力学シミュレーションによって入力形状内部に充填する手法である。本手法には、

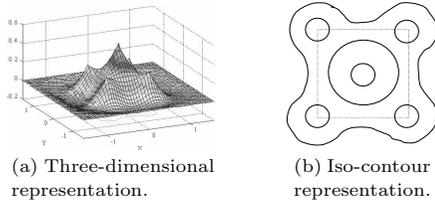
- (1) 複雑な形状であっても自動的に、隙間や重なりが少ない適度な密度で粒子を充填できる、
- (2) ユーザの意図する任意のベクトル場に沿うように整列させて粒子を充填できる、

などの特徴がある。

本手法ではまず、図4(a)に示すように、領域形状および、粒子の整列方向を表すベクトル場を入力する。続いて、図4(b)に示すように、領域境界線上および領域内部に適当な密度で粒子を発生させる。本手法では、領域境界線上の粒子は不動であり、領域内部の粒子は可動であるとする。

続いて、各々の粒子が近接粒子間から受ける引力や斥力の総和を算出し、運動方程式を用いて領域内部の粒子の移動量を算出する。この粒子移動処理を反復することで、粒子が整列方向ベクトル場に沿いながら最密充填した状態を得ることができる。図4(c)に、粒子を充填した結果を示す。

本手法では、粒子から近隣粒子への斥力を、粒子の中心点を原点としたときの近隣粒子の位置の関数で算出する。図5に、ある粒子から近隣粒子への斥力の分布図を示す。この関数は、四角形粒子の中心点および4頂点に極大点を持つ5個の3次関数を重ね合わせたものである（文献1）参照。この関数がもたらす斥力の等値曲線は、四角形に近い形状となる。この斥力が、四角形粒子の内部まで近隣粒子が接近することを防ぎ、近隣粒子間の一定の距離を保っている。



(a) Three-dimensional representation.

(b) Iso-contour representation.

図5 粒子の引力や斥力の算出に用いる関数

Fig. 5 A function of attractive and repulsive forces of a particle.

四角形粒子の方向は、入力データであるベクトル場を参照して決定される。ベクトルの大きさがゼロになる特異点（渦中心点など）では、四角形粒子の4頂点の座標値を算出できないので、四角形粒子の中心点を極大点とする斥力の関数だけを用いて、近隣粒子との斥力を算出する。現実的には、ベクトル場の特異点と厳密に同じ位置に四角形粒子が到達する可能性はきわめて低いので、この実装によって結果に影響が出ることはほとんどない。

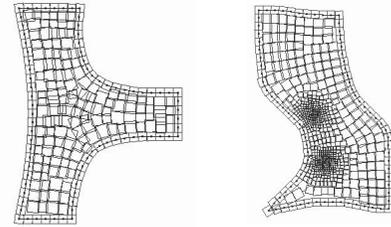
本手法で粒子を整列させるために入力するベクトル場は、ユーザが明示的に与えることもできるし自動生成することもできる。本論文では、形状領域の境界に沿った整列方向を自動算出することでベクトル場を生成している。形状領域の境界線が  $n$  本の線分列  $s_i (i = 1, \dots, n)$  で構成されているとき、本論文では形状領域内部の任意の点  $p$  におけるベクトル値  $v$  を、式(1)で算出する。ここで  $|s_i|$  は線分  $s_i$  の長さ、 $d_i$  は  $p$  と  $s_i$  との距離を表す。

$$v = \frac{1}{n} \sum_{i=1}^n \frac{|s_i|}{d_i^2} \quad (1)$$

本手法では、粒子の整列方向だけでなく、大きさも制御することができる。図6(a)は、粒子の大きさに乱数を用いて充填した結果である。図6(b)は、位置の関数で表現された粒子の大きさ分布を参照しながら、粒子を充填した結果である。図6(a)のように乱数を用いることで、石畳を構成する石材の不規則な大きさ変化を表現することができる。

スクウェアパッキングの計算時間は、近隣粒子間の力の算出回数に比例する。ここで、粒子の総数を  $n$ 、1個の粒子に近隣する粒子の数の平均を  $m$  とすると、計算時間は  $O(mn)$  となる。筆者らの実装では、空間分割法を用いて近隣粒子を検索しながら斥力計算をするので、1個の粒子に近隣する粒子の数  $m$  をおおむね定数と見なすことができる。よって本手法の計算時間は、おおむね  $O(n)$  である。

以上の説明は平面形状を仮定しているが、以下の2



(a) Sizes of particles are randomized.

(b) Sizes of particles are calculated according to a function of position.

図6 大きさの異なる四角形粒子を充填した結果

Fig. 6 A result of packing process of non-uniform square particles.

通りの手段を用いることで、平面だけでなく空間曲面に対して本手法を適用することも可能である。

1. ポリゴンの集合として空間曲面が定義されているときには、各々のポリゴン上で粒子を生成する。入力形状が非常に細かいポリゴンの集合で定義されているときには、入力形状を単純化した形状の上で粒子を生成し、続いて生成された粒子を入力形状上に投影する。

2. パラメトリック曲面の集合として空間曲面が定義されているときには、パラメトリック曲面上で粒子を生成する。

本論文での実装には1.を用いている。2.も、文献9)に示した「パラメトリック曲面上での楕円形粒子生成」の手法を拡張することで実現できるが、本論文での実装には用いていない。以上により、本手法は平らな地面だけでなく、曲面として坂やうねりを表現した地面に対しても石畳を表現することができる。

## 5. 石畳テクスチャの生成について

石畳テクスチャは、生成したパッキングパターンの各閉領域（セル）に対し、以下の3つの処理により敷石テクスチャを生成して得る。

- (1) 敷石形状を定義する初期格子の生成と変形
- (2) 基本格子のスムージング（丸め）処理
- (3) 敷石表面への微細凹凸の付加

### 5.1 初期格子の生成と変形

はじめに、得られた2次元のパッキングパターンの各セルを必要に応じてスケールする。この処理は、生成したい路面のタイプに応じて、敷石の間隔を空けたり、逆に敷石をより緊密に敷き詰めたりしたい場合に行う。続いて、得られた各セルを用いて、3次元の初期格子を生成する。初期格子は、パッキングパターンの各セルを底面とし、図7(a)に示すように、与えられた底面を指定された敷石の高さ分だけ垂直方向に

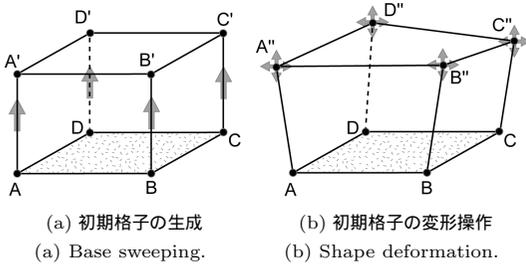


図7 初期格子の生成と変形  
Fig.7 Initial stone mesh.

スイープした角柱として与えられる。

生成された初期格子は、角柱の上面に位置する四隅の頂点  $A'$ ,  $B'$ ,  $C'$ ,  $D'$  を、図7(b)に示すように与えられた変形量の範囲でランダムに移動させて変形する。指定された変形率  $Dr$  に対する変形量の範囲  $Range$  は、敷石の大きさが  $Ssize$  で与えられた場合、式(2)で表される。

$$Range = Ssize \times (1.0 + Dr \times Random) \quad (2)$$

ここで、 $Random$  は、正規分布乱数である。

設定する変形量が大きいと、自然のままに採取されたような敷石が生成され、変形量が少ないと、人工的に形の整えられたような敷石が生成される。

### 5.2 初期格子のスムージング

5.1節で述べた変形後の初期格子を制御格子と想定して、サブディビジョンサーフェイスの生成による角柱のスムージング(丸め)処理を行い、敷石のおおまかな形状を生成する。

#### 5.2.1 サブディビジョンサーフェイスの生成法

サブディビジョンサーフェイスの生成手法には、Doorらの手法<sup>10)</sup>や、Catmullらの手法<sup>11)</sup>、Loopの手法<sup>12)</sup>などが報告されている。本手法では、三角形メッシュが生成されることなどの理由から、筆者らの開発環境になじみやすいLoopの手法を用いている。以下に、Loopのアルゴリズムを簡単に述べる。

Loopのサブディビジョンサーフェイス生成のアルゴリズムでは、図8(a)のように与えられた三角メッシュの各エッジに対し、図8(b)に示すように新しい頂点(図中の黒丸)を計算し、各エッジを2つに分割する。さらに、新しい頂点どうしを結んで、1枚の三角形が4枚の三角形に新たに分割され、分割前の三角形を置き換えていく。新しい頂点の位置は、周囲の頂点の位置とエッジの接続状況などから計算される。

#### 5.2.2 敷石形状の種類

本手法では、初期格子のメッシング法を変えることで、丸石と角石の2種類の敷石形状を定義する。

(1) 丸石の場合 変形後の基本格子に対し、図9に

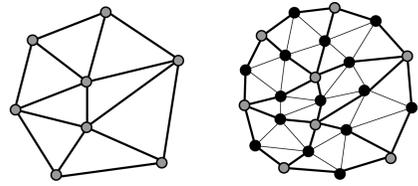


図8 Loopの手法による三角メッシュの分割  
Fig.8 Loop's subdivision method.

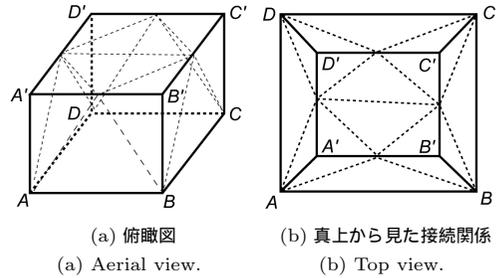


図9 丸石に対する基本格子の三角メッシュ  
Fig.9 Triangular mesh for cobblestone.

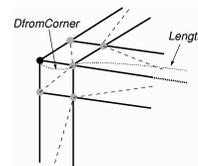


図10 角石の三角メッシュ時の追加頂点と三角メッシュ(一部)  
Fig.10 Triangular mesh for flagstone (partial).

示すような三角メッシュを生成し、上記の分割を繰り返すことで、基本格子をスムージング処理して敷石形状を生成する。

(2) 角石の場合 角石を生成する場合は、図10に示すように、基本格子のコーナーの部分に頂点を追加して、分割を行う。ここで、図10において、灰色の点が追加された頂点であり、コーナーからの各距離  $DfromCorner$  は、エッジの長さ  $Length$  との比率として与えられたパラメータ  $Br$  から、式(3)を用いて求める。本手法では、 $Br$  をベベル率と呼び、敷石の角の丸め度合いを制御するパラメータとする。

$$DfromCorner = \frac{Length}{2} \times Br \quad (3)$$

### 5.3 微細凹凸の付加

5.2節で生成された敷石形状の表面に、フラクタルノイズを用いて微細な凹凸形状を付加して、最終的な敷石形状とする。フラクタルノイズは、中点変位法<sup>13)</sup>にて生成されるものとし、作業領域として用意された

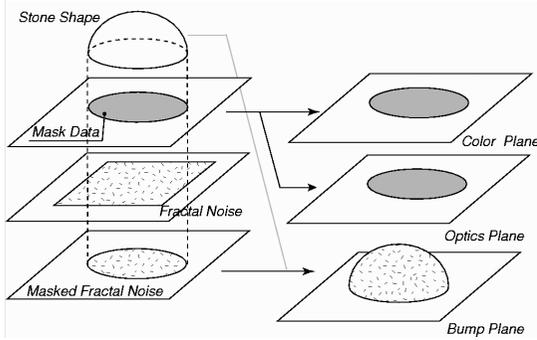


図 11 微細凹凸の付加と敷石の属性データ  
Fig. 11 Surface displacement and attributes.

2次元格子上に発生させる。

微細凹凸を付加するには、まず、図 11 に示すように、生成された敷石形状を垂直方向に投影してできる領域、すなわち、敷石形状のデータが存在する領域を、マスクデータとして作成する。続いて、生成されたフラクタルノイズを、このマスクデータでマスキングし、敷石形状をバンプ属性のデータプレーンに書き込むときに付加する。また、このマスクデータは、後述する属性データの出力の際にも利用する。

5.4 敷石の属性

敷石の色や光学属性を指定するそれぞれの属性データプレーンを、与えられた初期値で初期化する。各敷石の属性データを、敷石の高さ情報をバンプデータプレーンに書き込む際に、図 11 に示すようにそれぞれの属性データプレーンに書き込んでいく。本手法では、あらかじめ属性データのセットを与えておき、コンピュータが乱数を振って、それらの属性データのセットからランダムに選択して、各敷石に色や光学属性を割り振るようにしてある。そして、マスクデータを用いて、敷石の存在する領域に対して、決定された属性値を設定する。

6. 実験結果

本章では、以上で述べた手法で生成された石畳テクスチャ(大きさは、512 × 512)を数例あげる。

パラメータ

生成時に指定したパラメータリストを表 1 にあげる。以下に、各パラメータ記号の意味を列挙する。

- Cs: セル(敷石の底面)の平均の大きさ
- Vs: セル(敷石の底面)の大きさのパラツキ
- Hs: 敷石の高さ
- Sv: セル(敷石の底面)のスケーリング値
- Dr: 初期格子の変形率(0.0 < Dr < 1.0)
- Fd: 敷石の表面の粗さ(フラクタル次元, 1.0 <

表 1 パラメータリスト  
Table 1 List of parameters.

Fig. No	Cs	Vs	Hs	Sv	Dr	Fd	Af	Br	Ts	Tt
12(1b)	23	0.0	18	1.0	0.3	1.5	2.0	-	2.1	8.4
12(2b)	18	0.1	18	1.0	0.3	1.5	2.0	-	2.1	8.8
12(3b)	23	0.3	18	1.0	0.3	1.5	2.0	-	2.1	8.7
13(b)	20	0.0	20	1.0	0.3	1.5	2.0	-	11.9	10.5
13(c)	20	0.0	12	1.0	0.2	1.4	1.4	0.2	11.9	11.3
14(a)	23	0.0	20	1.1	0.3	1.5	2.0	-	3.6	7.8
14(b)	23	0.0	20	1.1	0.0	1.5	2.0	-	3.6	7.7
14(c)	23	0.0	20	1.1	0.6	1.5	2.0	-	3.6	7.9
15(a)	21	0.0	18	1.0	0.3	1.5	2.0	-	9.1	9.3
15(b)	25	0.0	20	1.0	0.5	1.5	2.0	-	7.4	13.6
15(c)	15	0.0	8	1.0	0.2	1.3	1.0	0.2	2.1	8.8
15(d)	23	0.0	14	1.1	0.3	1.5	2.0	0.2	4.0	11.8
15(e)	20	0.0	12	1.0	0.2	1.3	1.2	0.2	4.8	12.7
15(f)	20	0.0	18	0.8	0.3	1.5	2.0	-	3.6	8.2

$Fd < 2.0$ )

Af: フラクタルノイズの振幅

Br: 敷石のベベル率(0.0 < Br < 1.0)(角石のみ)

Ts: スクエアパッキングに要した計算時間(秒)

Tt: テクスチャ生成に要した計算時間(秒)

計算時間

平均の処理時間は、ペンティアム II の 300 MHz の PC を用いて生成するテクスチャサイズが 512 × 512 の場合、スクウェアパッキングの処理に約 5 秒、石畳テクスチャを生成するのに約 10 秒であった。

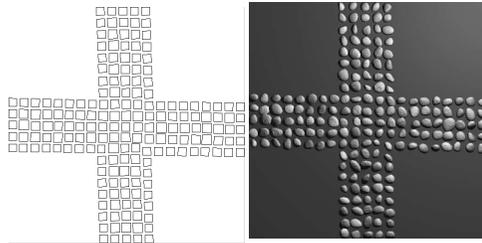
図 12~15 に示した実行例では、スクエアパッキング法の 1 回の力学計算は 0.1 秒以内であり、30~200 程度程度の反復計算で安定状態に達するため、2~12 秒程度の計算時間で収束に至っている。なお、反復回数は形状や整列方向に強く依存している。全体的な傾向として、四角形に近い形状領域、直線的な整列方向を与えたときに収束が早く、カーブを描くような形状領域や整列方向を与えたときに収束が遅い傾向にある。

スクウェアパッキングのパラメータによる違い

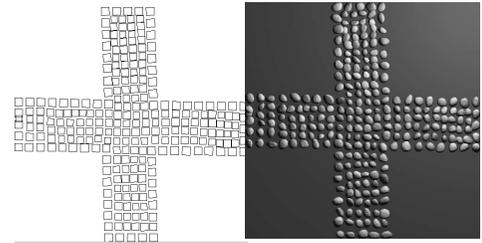
図 12 に、スクウェアパッキングのパラメータを変化させた場合の違いをあげる。図 12(1a)~(3a) が得られたパッキングパターン、図 12(1b)~(3b) が生成された石畳テクスチャの例である。図 12(1b) を基準とした場合、図 12(2b) はパッキング粒子の大きさを小さく(パラメータ Cs を小さく)した例、図 12(3b) はパッキング粒子の大きさのばらつきを大きく(パラメータ Vs を大きく)した例である。図 12(3b) では、敷石の大きさが大きくばらついていることが分かる。

丸石と角石

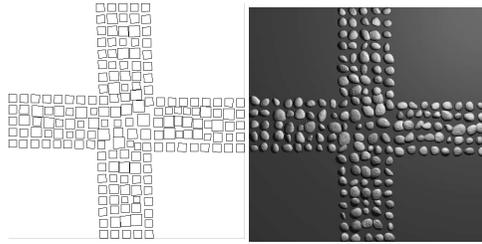
本手法で定義されている 2 種類の敷石形状による違いを図 13 に示す。図 13(a) に示すパッキングパターンから、丸石(図 13(b))と、角石の石畳テクスチャ



(1a)  $C_s = 23, V_s = 0.0$  (1b)

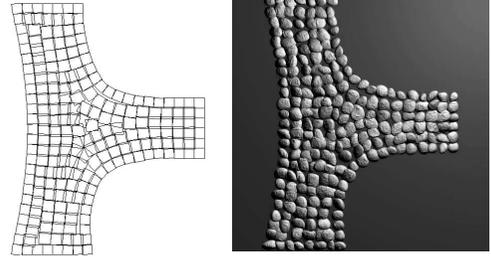


(2a)  $C_s = 18, V_s = 0.1$  (2b)



(3a)  $C_s = 23, V_s = 0.3$  (3b)

図 12 パッキングパラメータによる違い  
Fig. 12 Change in packing parameters.



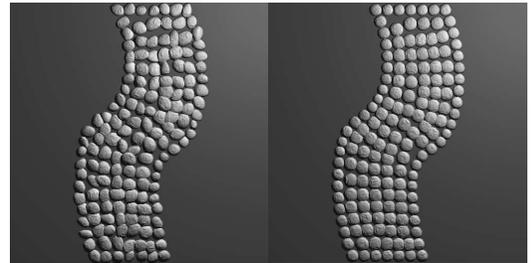
(a) Packing pattern. (b) Cobblestone pavement.



(c) Flagstone pavement.

図 13 敷石の形状の違い

Fig. 13 Comparison of cobblestone and flagstone.



(a) 変形率 :  $Dr = 0.3$  (b) 変形率 :  $Dr = 0.0$



(c) 変形率 :  $Dr = 0.6$

図 14 初期格子の変形率による違い

Fig. 14 Change in deformation ratio.

( 図 13 (c) ) をそれぞれ生成した例である .

初期格子の変形率の違い

初期格子の変形率の違いを 図 14 に示す . 図 14 (a) を基準とした場合 , 図 14 (b) は基本格子の変形率 ( パラメータ  $Dr$  ) を 0 にした例 , 図 14 (c) は基本格子の変形率を大きくした例である . 図 14 (b) では , ほとんどの敷石が同じような形状になっていることが分かる .

様々な石畳テクスチャ

図 15 に , 様々な石畳テクスチャの例を示す . 道路に穴が開いていたり , 道路が分岐したりしていても , 敷石が “流れ” を保ちながら敷き詰められていることが分かる .

7. おわりに

道路形状と数個のパラメータを入力して , 自動的に石畳テクスチャを生成する手法について述べた . 本手法を用いることにより , 1 つ 1 つの敷石をモデリングするような手間も要せず , デザイナーが希望する路面を被覆する , 石畳テクスチャの自動生成が可能になった .

ユーザが , よりデザインしやすいシステムにするには , 初期入力した道路形状を編集することで , 石畳テクスチャを編集できるようなシステムへの変更が必要である . これは , ユーザインタフェースを組み替えることで可能であると考えている .

今後の課題として , 以下の 3 つがあげられる .

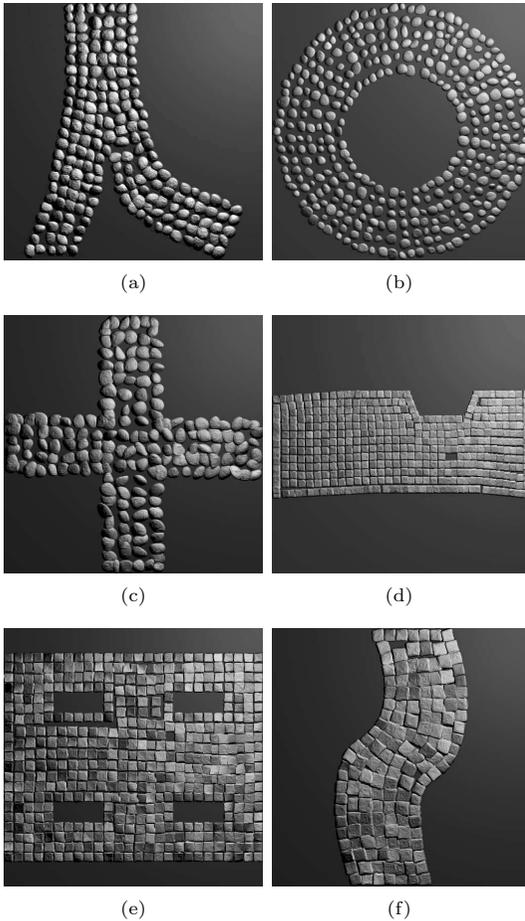


図 15 様々な石畳テクスチャの例  
Fig. 15 Various pavement textures.

### 経年変化

より現実感を増すためには、何らかの経年変化を付加する必要がある。経年変化に関しては、いくつかの研究例が報告されている<sup>5),14)</sup>。これらの自然な経年変化以外に、石畳の場合には、車の走行や人の歩行に起因する人為的な摩滅や破壊なども考慮する必要がある。

### 様々な敷石

本手法では、敷石形状をプロシージャルな手法で生成した。今後の課題としては、実サンプルからの敷石形状のパラメータ抽出法や、ぬれた敷石や苔むした敷石、敷石の汚れの付加などの表現があげられる。

### 有機的なテクスチャへの応用

本論文では正方形セルの充填モデルを用いたが、ほかにも円形セル<sup>8)</sup>や、楕円セル<sup>9)</sup>など、様々な形状のセルに対する充填法を報告している。これらの手法を組み合わせ、爬虫類の鱗や亀甲などの有機的なテクスチャの生成法に応用していきたい。

### 参考文献

- 1) Shimada, K., Liao, J. and Itoh, T.: Quadrilateral Meshing with Directionality Control through the Packing of Square Cells, *7th International Meshing Roundtable*, pp.61–76 (1998).
- 2) Miyata, K.: A method of generating stone wall patterns, *Proc. SIGGRAPH '90*, pp.387–394 (1990).
- 3) 石井ほか：表面の微細構造に注目した皮膚の質感表現の一手法，情報処理学会論文誌，Vol.32, No.5, pp.645–654 (1992).
- 4) 東海ほか：爬虫類皮革の質感表現の一手法，情報処理学会論文誌，Vol.35, No.5, pp.801–809 (1994).
- 5) Dorsey, J., et al.: Modeling and Rendering of Weathered Stone, *Proc. SIGGRAPH '99*, pp.225–234 (1999).
- 6) Perlin, K.: An Image Synthesizer, *Proc. SIGGRAPH '85*, pp.287–296 (1985).
- 7) Fleischer, K.W., et al.: Cellular texture generation, *Proc. SIGGRAPH '95*, pp.239–248 (1995).
- 8) Shimada, K.: Physically-based Mesh Generation: Automated Triangulation of Surfaces and Volumes via bubble Packing, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, U.S.A. (1993).
- 9) Shimada, K., Yamada, A. and Itoh, T.: Anisotropic Triangular Meshing of Parametric Surfaces via Close Packing of Ellipsoidal Bubbles, *6th International Meshing Roundtable*, pp.375–390 (1997).
- 10) Doo, D. and Sabin, M.: Analysis of the behavior of recursive division surfaces near extraordinary points, *Computer Aided Design*, Vol.10, No.6, pp.356–360 (1978).
- 11) Catmull, E. and Clark, J.: Recursively generated B-spline surfaces on arbitrary topological meshes, *Computer Aided Design*, Vol.10, No.6, pp.350–355 (1978).
- 12) Loop, C.: Smooth subdivision surfaces based on triangles, Master's Thesis, Department of Mathematics, University of Utah (1987).
- 13) Fournier, A., Fussell, D. and Carpenter, L.: Computer rendering of stochastic models, *Comm. ACM*, Vol.25, No.6, pp.371–384 (1982).
- 14) Dorsey, J. and Hanrahan, P.: Modeling and rendering of metallic patinas, *Proc. SIGGRAPH '96*, pp.387–396 (1996).

(平成 13 年 2 月 27 日受付)

(平成 13 年 9 月 12 日採録)



宮田 一乗 (正会員)

1962年生。1986年東京工業大学大学院総合理工学研究科物理情報工学専攻修士課程修了。同年日本アイ・ピー・エム(株)入社,東京基礎研究所勤務。1998年より東京工芸大学芸術学部映像学科助教授,2001年より同学部メディアアート表現学科助教授。博士(工学)。フラクタル,自然形状のモデリングおよびレンダリング,テクスチャ生成,画像合成等の研究に従事。電子情報通信学会,画像電子学会,芸術科学会,ACM各会員。



伊藤 貴之 (正会員)

1968年生。1992年早稲田大学大学院理工学研究科電気工学専攻修士課程修了。同年日本アイ・ピー・エム(株)入社。現在同社東京基礎研究所に勤務。博士(工学)。写実的な画像生成,メッシュ生成,曲面生成,情報視覚化等の研究に従事。IEEE,ACM,芸術科学会各会員。



嶋田 憲司 (正会員)

1983年東京大学工学部精密機械工学科卒業。1985年同大学院精密機械工学専攻修士課程修了。同年日本アイ・ピー・エム(株)入社,東京基礎研究所にて形状処理,ロボティクス,コンピュータグラフィックス,計算力学等の研究に従事。1989年より1993年までマサチューセッツ工科大学博士課程。Ph.D. 1996年よりカーネギーメロン大学機械工学科とロボット研究所の助教授と準教授を歴任。1994年本会山下記念研究賞,同年ニコグラフ論文コンテスト優秀論文賞,2000年米国立科学財団(NSF) CAREER AWARD。ACM, ASME, IEEE, SIAM各会員。