芸術科学会論文誌 Vol. 16, No. 4, pp. 118 – 129 (2017)

ゼンタングルのモデル化とシミュレーション

堀友香 [1)](学生会員) 山田雅之 [1)](正会員) 中貴俊 [1)](正会員) 宮崎慎也 [1)](正会員)

1) 中京大学工学部

Modeling and Simulation of Zentangle

Yuka Hori[1)]**(Student Member)** **Masashi Yamada**[1)]**(Member)**
Takatoshi Naka[1)]**(Member)** **Shinya Miyazaki**[1)]**(Member)**

1) School of Engineering, Chukyo University

t314074 @ m.chukyo-u.ac.jp , {myamada, t-naka, miyazaki} @ sist.chukyo-u.ac.jp

概要

ゼンタングルとは対称性や自己相似性などの規則性を有するパターンアートの一種である．本論文では，ゼンタングルの自動生成を目的として，模様を描く手順をモデル化し，そのコンピュータシミュレーションを実現する．「規則性」「不均一性」及び「充填プロセス」の３つの特徴を有するゼンタングルを対象とし，これらの特徴を再現する手法を提案する．具体的には，３種類の基本図形を元に変形及び配置を繰り返すことにより，規則性と不均一性を有する様々なパターンを生成し，それらのパターンを充填対象領域に描画する操作を反復する．この手法によって自動生成されたゼンタングルの作品は，人の手描きによる作品と類似し，ゼンタングル固有の特徴を有することを実験により確認する．また，模様を対話的に構成できるように専用の制作ツールも作成した．このツールはパターンの意図的な選択や充填領域の指定を行うための機能，読み込んだモノクロの画像にパターンを追加する機能なども備えており，これにより，ユーザは自己の意図を反映した模様を制作できる．

Abstract

Zentangle is a pattern art composed of regular patterns, such as symmetric patterns and self-similarity patterns. We formalized a model of pattern generating processes and conducted computer simulation for generating zentangle patterns automatically. This paper describes three features of zentanlge covered in this research, regularity, ununiformity, and filling process, then presents a model of zentangle. This model premises that zentangle patterns are composed of three classes of patterns. Also, this model deforms the patterns and draws them on empty white regions repeatedly for simulating the features of ununiformity and filling processes. The patterns based on this model have similar features to hand drawn zentangle patterns. Our simulation tool has some functions for interactive pattern composition, as well as auto-generating functions. For example, users can select a sub-pattern to fill a certain target region to compose desired zentangle patterns, or can use gray scale images to define target regions filled with sub-patterns.

1 はじめに

ゼンタングルとは簡単なパターンの組み合わせで誰でも美しいアートを楽しむことのできるメソッドである．その名称は「Zen（禅）」と「Tangle（絡まる）」を合わせた造語で，ヨガや瞑想のように日常生活で取り入れるリラクゼーションの方法としてアメリカで考案された．近年では，ゼンタングル入門書 [1][2] も書店で売られるようになり，ゼンタングルを楽しむ人が増え，zentangle をキーワードとして Web 画像検索すれば，様々なゼンタングルの作品を見つけることができる．

対称性や自己相似性など，なんらかの規則性を有する模様を視覚的パターンといい，人はこれに美しさを感じることが多く，装飾にもよく用いられる．ゼンタングルは複数のパターンを組み合わせて得られる模様であり，単一のパターンに比べ複雑さを有し，それゆえ個性や面白さを有すると考えられる. 図 1 は筆者が描いたゼンタングルの例である．アステカ柄やパッチワークの模様に似ているが，ゼンタングルはそれらよりも図形要素や配置の自由度が高い．

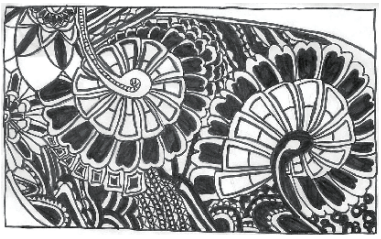

図 1　手描きのゼンタングルの模様

ゼンタングルの多くの作品をみると，想像のつかない構成や表現，作品群の多様性に魅了される．このように感じ，自らも描いてみたいと思う人が増えてきていると考えられる．したがって，多様なゼンタングルの模様を自動生成できれば，それらを鑑賞して楽しんだり，その中から美しいと思う作品を見つけて，塗り絵など別の用途に活用できる可能性がある．さらに，ユーザの意図を反映できれば，ユーザの好みや活用目的にあった模様を生成しやすくなると思われる．このような理由から，本研究は，ゼンタングルのような多様で複雑性のある美しい模様を自動生成，あるいは，少ないユーザ操作で半自動生成することを目的とする．

入門書 [1] には，紙面に線を描いて複数の領域に分割したあと，それぞれの領域にパターンを描くことにより，パターンで埋め尽くされた模様を制作する方法が示されている．図 2 は模様を描く過程の例であり，始めは大まかな図形を描いて領域に分割し，左上と右下の領域には描き込む図形の形や種類を変えながら徐々に変化をつけている．これは，図形を描いて得られる細部の領域に，さらに図形を描き込む充填的なプロセスと見ることもできる．このプロセスには，階層性や分岐があり，単純な図形を用いて多様な模様を生成する手順にもなっている．図 3 は階層数の異なる 4 つの手描きのゼンタングル作品例である．この入門書で示されている描き方の例の中で，このような充填的プロセスを確認できるものは，167 個中 68 個ある．また，インターネット上で見られるゼンタングルの作品において，充填的プロセスを部分的に，または全体に用いたと推測できるものは 5 割程度であった．ただし，充填的プロセスが認められるものの，紙面の大きさ，ペンの太さなどの制約から，細かい模様を描かず，結果的に階層が浅い作品もある．一方，充填的プロセスと異なる方法で描かれた例も多く見られる．あるモチーフ（例えば魚）を表現することを目的に描かれ，そのモチーフの表現に適したパターン（魚の場合，鱗模様）を描き込んだ作品や，濃淡を表現するためハッチングの代替としてパターンを描き込んだ作品もある．これらは，描画するパターンに意味や文脈があり，デザイン指向の高い作品と考えられる．これに対し，充填的プロセスは出来上がりの作品を想定せず発見的に模様を構成する作品によく見られる．

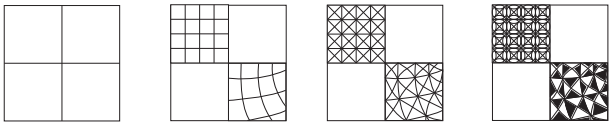

図 2　手描きの場合の描画手順の例

本論文は，先に述べたようにゼンタングルのような模様の自動生成・半自動生成を目的とするが，ゼンタングルのあらゆる作品を網羅するような機能の実現は現時点では困難なため，以下の 3 つの特徴を満たすゼンタングルの模様を対象とする．

規則性：　基本的な図形要素を繰り返し用いて視覚的パターンを構成する．

不均一性：　手描きが基本であることから同種の図形で

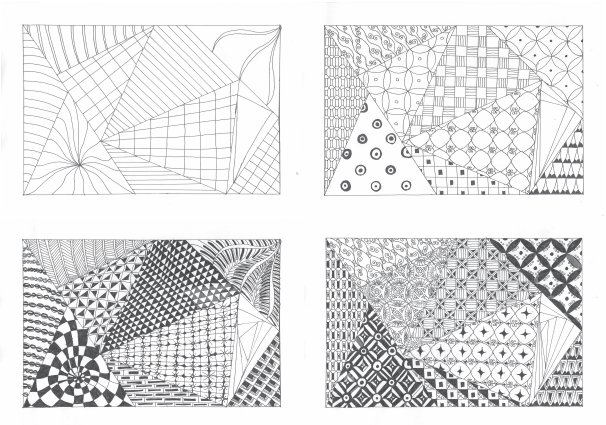

図 3　階層数の異なる手描きのゼンタングル作品例

も表現に「ゆれ」があり，全体的に不均一性がある．

充填プロセス：　白い領域にパターンを描くことを繰り返す. パターンを描くことにより，領域は分割され，面積の大きい領域がなくなるまで充填する．

本論文では，この特徴を再現するゼンタングルのモデルおよびシミュレーション方法を示し，実際のゼンタングルの作品に見られる美しさや複雑さを有する模様をどの程度再現できるかを確認する．また，そのプロセスに人が介在できる UI を作成し，少ないユーザ操作で簡単に満足できる模様を生成できるかを確認するとともに，美しいと感じるゼンタングルの模様の生成に必要な要素を考察する．

2　関連研究

パターンを人工的に表現・生成する試みは古代から現代に至るまで様々に行われ，芸術や装飾デザインなどに活用されてきた [3]．パターンに潜む規則性に関する性質には，フラクタルや対称性などがあり，その数学的な表現はコンピュータでパターンを生成する際にも利用されている [4][5]．植物の分岐構造や細胞の分裂プロセスを表現できる L-system，螺旋モデルなどは，コンピュータグラフィックスおいて自然物をモデリングする際によく利用されるパターンの表現手法である [6][7][8][9][10][11]．

本研究ではゼンタングルの模様の生成を目的とするが，模様生成に関する研究には，分岐構造を考慮しながら図形や画像パターンを平面に充填して装飾柄を生成するもの，逆に図形要素が配置された模様から配置規則を抽出し，その配置規則を他の模様の生成に利用するもの，組みひもパターンを組み合わせた装飾模様の生成を支援するシステムの開発などがある [12][13][14]．

近年では，エグザンプラーとして画像を与え，その濃淡特徴を反映したテクスチャを合成する手法も多く提案されており，ユーザが絵や模様を作成する際の塗りつぶしパターンやブラシパターンを合成することにも利用されている [15][16][17]．畳込みニューラルネットワークによりテクスチャ特徴を学習して，その特徴を任意の画像に付与する手法 [18][19] は汎用性が高く，ゼンタングルの模様をエグザンプラーとして与えれば，そのゼンタングルの模様の特徴を有する画像を合成できる．しかし，これらの手法ではエグザンプラーの濃淡特徴のみ考慮され，幾何的構成は考慮されない．そのため，本研究では，ラスタ画像ベースのテクスチャ合成のアプローチはとらず，先の例のように幾何学的図形を規則的に配置することによりゼンタングルの模様を生成する方法を検討する．

また，ユーザが描いたストロークパターンをエグザンプラーとし用いるベクターベースのテクスチャ合成手法も提案されている [20][21]．この中で Kazi らはユーザが入力したペンストロークで構成されるパッチを 1 次元的または 2 次元的に配置し，ユーザの意図を反映したパターンを合成する方法およびパターンのトーン（濃さ）や向きをユーザが制御できる UI を開発し，アーティストが高品質な作品を短時間で制作できることを示している．これは先に述べたデザイン指向の模様を制作するには有用である．しかしながら，階層的な組合せにより多様な模様を自動生成する本研究のアプローチとは異なる．

3　ゼンタングルの模様生成モデル

ここでは，上で述べたゼンタングルの特徴を考慮しながら，ゼンタングルの模様生成モデルを構成していく．まず，提案するモデルの概要を説明する．

ゼンタングルは基本的な図形要素から構成されるものとする．図 4 に基本図形とその名称（略名）を示す．これらの図形のうち，cr, sp, $c0, \cdots, c4$ は黒色の線のみで構成される図形であり，tr, sq, as, cl, st, ss, le は黒色の輪郭線と輪郭線で囲まれた閉じた領域をもつ図形である．後者の図形を描画する際，輪郭線で囲まれた領域は，白色あるいは黒色で塗りつぶすこととする．このような

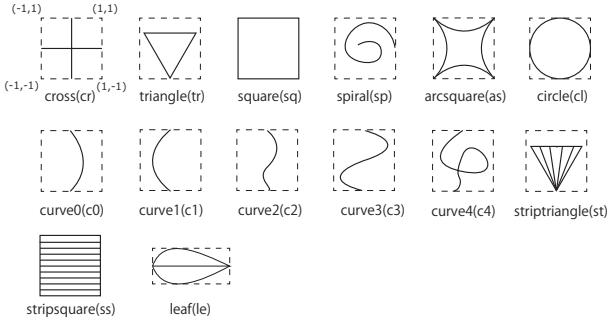

図 4　基本図形: 大きさや位置は，破線で表す原点中心，一辺 2 の正方形を基準とする

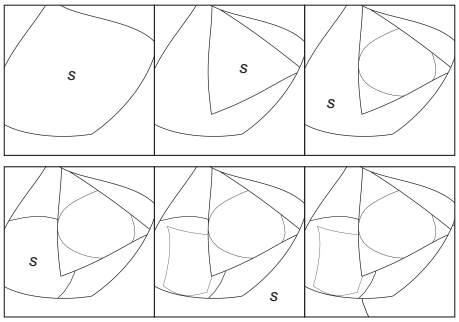

図 5　提案モデルに基づく描画例

色を領域の表示色と呼ぶ．図形は，自由形状変形 [22] とアフィン変換（回転，平行移動，スケーリング）を施してから描画するが，このとき乱数を用いて変形・変換のパラメータを調節し，ゼンタングルの特徴である不均一性を再現する．図形を充填する領域は，紙面上にある白色の領域の内部のみとする．ゼンタングルの充填プロセスを本モデルでは次のように定める．

1. 白い連結領域の集合を S とする. 最初は $S = \{s_0\}$ とし，s_0 は何も描かれていない紙面に対応する.
2. S から 1 つの連結領域 s を選択し，その内部に図形 p を描画する.
3. 図形 p を描画すると s は複数の領域に分割され，それらの集合を $\acute{S}$ とする．$S = S \cup \acute{S}$ を新たな連結領域の集合として 2 へ戻る.

図 5 は基本図形の三角形 tr，四角形 sq, 円 cl を上記に従って描画した例である．まず，四角形を変形したあとで紙面上に描画する．左上がその結果であり，次に s で示した領域に，変形した三角形を描画した結果が上段中である．同様にして，s で示した領域に，逐次，円，円，四角形，円を描画した結果が右下の模様である．このように，本論文における模様生成モデルとは，構成する図形とその描画プロセスからなる．

4　規則性の付与

ゼンタングルが美しく感じる要因として，その模様がパターン，すなわち図形の配置や大きさに関する規則性を有する模様である点があげられる．ここでは規則性をもつ模様を生成するために，2 種類の規則性である自己類似性および対称性を有するパターンを模様生成モデルに導入する．さらに，領域形状を考慮したパターンも導入する．以降では，これら 3 種類のパターンを，自己類似パターン，対称パターン，および領域依存パターンと呼ぶ.

4.1　自己類似パターン

フラクタル（自己相似）に準ずるものとして，厳密な自己相似性はないが，拡大しても縮小しても同じような模様に見えるという性質を，ここでは自己類似性と呼ぶことにする．以降では，模様生成モデルにこの自己類似性を有するパターンを付与するための方法について述べる.

基本図形には輪郭線で囲まれた領域をもつものがあることを先に述べた．描画する際は，表示色（白または黒）で塗りつぶすが，領域はこの表示色に加え，シードという属性をもつものとする．シードは，次のステップでその領域に埋め込むことができる図形の種類に対応するものとする．例えば四角形を描画するとき，その輪郭線内部の領域 s のシードを三角形とすれば，以降のステップで，領域 s に埋め込める図形は三角形のみとなる.

このようなシードに加え，図形を描画する際，領域の大きさに合わせて図形を拡大縮小することにより，自己類似性を有するパターンを生成する．図 6 に例を示す．上段は四角形の輪郭線内部のシードを四角形自身とした結果である．従って，四角形の中に繰り返し四角形が描かれたパターンとなる．なお，初期状態（白紙の状態）の領域についても，シードを四角形としている．左右は描画する図形の数を変えている．下段は，円の領域のシードを四角形，四角形の領域のシードを三角形，三角形の領域のシードを円とし，初期状態の領域のシードを { 三角形，四角形，円 } の 3 種類から選べるものとしたときの結果である．また，図形を描画する際の拡大縮小比率

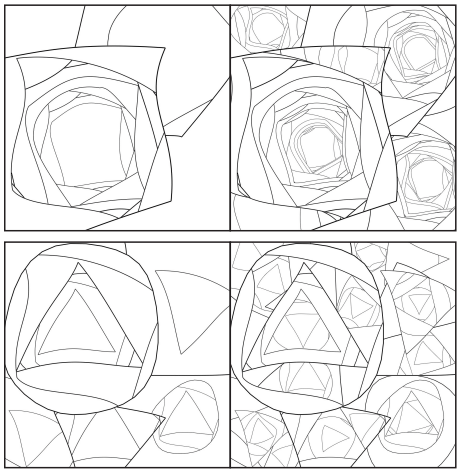

図 6　シードを指定した描画

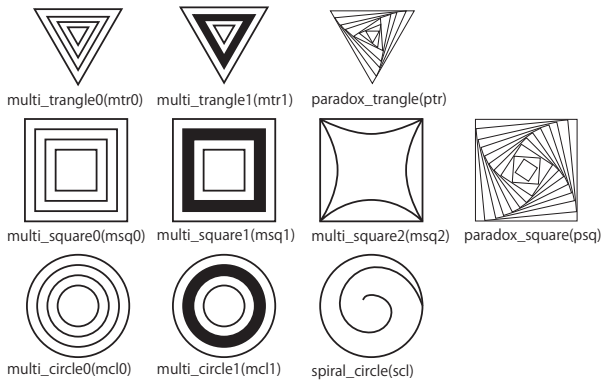

図 7　複合図形

は，領域の外接矩形の面積の平方根に比例するようにしている．

4.2　対称パターン

ここでは，並進対称性，回転対称性などの対称性をもつパターンを基本図形から構成する．まず，図 7 には基本図形の三角形，四角形，円の内部に基本図形を縮小して配置した図形を示す．これらは対称パターンの例であるが，以後，これらを複合図形とも呼ぶこととする．図 8 には，基本図形または複合図形を対称性を考慮して配置するための配置テンプレートを示す．アルファベットの記号 a, b, c, d は基本図形あるいは複合図形を表し，任意に選択できるものとする．ただし，MR は対称性の小さい，ランダムな配置としている．また，ML は植物の枝と葉をモチーフにしたパターンで，線分と基本図形 le で構成される．図 9 の左右の図は，それぞれ MC，MS により配置して得られるパターンである．

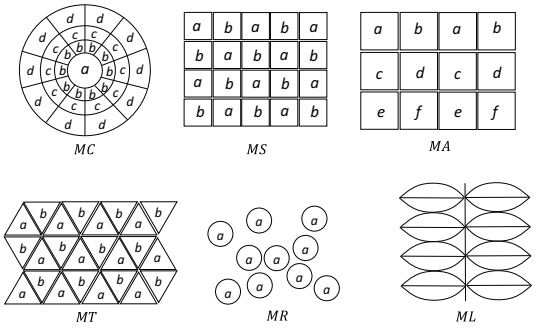

図 8　対称性を考慮した配置テンプレート

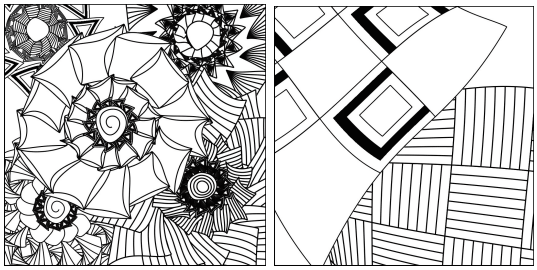

図 9　対称パターンの例: 左は MC，右は MS の配置を利用

4.3　領域依存パターン

上記の対称パターンは，図形を埋め込む領域の形状は考慮していない．一方で，ゼンタングルの作品を観察すると，領域の輪郭線形状を利用したパターンや，領域内部を同じ図形を使って埋め尽くすパターンが多くある．ここでは，図 10 に示すパターンをモデルに導入する．パターン $along_border$ は領域の一番外側の輪郭線[*1]に沿って，図形を配置したパターンである．パターン $contour_line$ は領域の輪郭線から距離に従って数段階の等高線を配置したパターンである．パターン $paradox$ は領域の一番外側の輪郭線を多角形で近似し，多角形の辺上の点を規則的に結んだ折れ線で構成されるパターンである．パターン $voronoi$, $delaunay_triangles$ はそれぞれ領域内部にランダムに点を配置して得られるボロノイパターン，ドロネー三角形分割である．パターン $fill$ は領域内を一種類の図形で隙間なく埋め尽くしたパターンである．

5　シミュレーションの方法

ここでは，ゼンタングルの模様生成のプロセスを上記のモデルに基づきシミュレーションする方法を述べる．

[*1] 領域はドーナツ状に穴が開いたものもある．

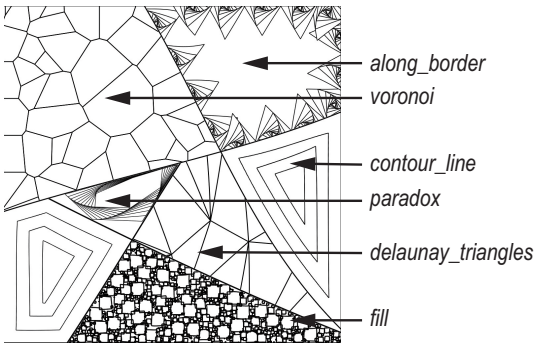

図 10　領域依存パターン

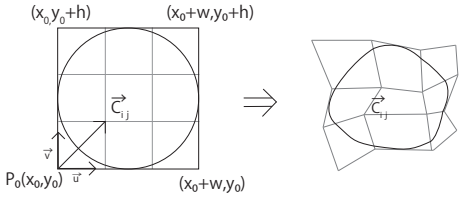

図 11　自由形状変形

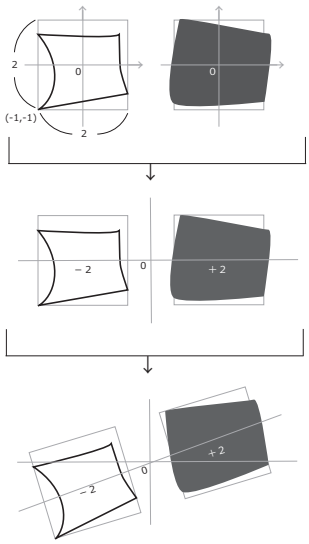

図 12　2 つの四角形に自由形状変形とアフィン変換を施す例

5.1　パターンの生成方法

4.2 で述べたように対称パターンは図 4 の基本図形から構成する．基本図形の配置や拡大縮小には平行移動 $T(x,y)$, 回転 $R(\theta)$，スケーリング $S(s)$ を合成したアフィン変換 m を用いる．また，ゼンタングルの特徴の 1 つである不均一性を再現するため，対称パターンを構成する基本図形あるいはパターン全体に自由形状変形を施す．自由形状変形には，図 11 に示す 4×4 の制御点格子を用いる．この格子を対象となる図形または対称パターン p の大きさと一致するように，拡大縮小したあとで，制御点 $\overrightarrow{C_{ij}}$ を $s\overrightarrow{u}+t\overrightarrow{v}$ だけ移動し，移動後の制御点に従って，p をスプライン変形する．ここで,$\overrightarrow{u}$,$\overrightarrow{v}$ は横方向, 縦方向の単位ベクトル，s, t は不等式 $|s| \leqq r_x/2, |t| \leqq r_y/2$ を満たす乱数値とし，r_x, r_y は横方向, 縦方向の格子点間の間隔を表す．なお，このような自由形状変形を $f = F^{r_x,r_y}_{x_0,y_0,w,h}$ で表す．ここで (x_0, y_0), w,h は制御点格子の左下座標，横幅，高さを表す．

次の式は，自由形状変形とアフィン変換を組み合わせて 2 つの四角形を変形・配置する例である．

$$[_{m_0}\ [_{m_1}\ [_{f_0}\ sq\ ]\ ],[_{m_2}\ [_{f_0}\ \hat{sq}\ ]\ ]\ ]$$
$$m_0 = R(\tfrac{\pi}{6}), m_1 = T(-2,0), m_2 = T(2,0), f_0 = F^{2/3,2/3}_{-1,-1,2,2}$$

この式では，アフィン変換 m や自由形状変形 f の適用範囲が明確になるように，図形やパターン p にそれらを適用することを $[_m\ p\ ]$，$[_f\ p\ ]$ のように表記する．また，白色で塗りつぶす四角形を sq，黒色で塗りつぶす四角形を $\hat{sq}$ と表す．図 12 はこの変形・配置の様子を示す．2 つの四角形をそれぞれ自由形状変形し（上段），次に中心が $(-2,0)$，$(2,0)$ となるよう平行移動し（中段），最後に両者を原点中心に $\frac{\pi}{6}$ 回転させる（下段）．

このような自由形状変形とアフィン変換の組み合わせにより，図 9 のようなパターンを基本図形から構成できる．

5.2　パターン生成・描画プロシージャ

基本図形，対称パターン，領域依存パターンを描画するための計 20 種類の描画プロシージャを作成し，そのプロシージャを繰り返し呼び出して，ゼンタングルの模様を描画していく．各プロシージャには ID を割り振る．図 13 には，各プロシージャの ID とそのプロシージャが描画する図形・パターンの例を示す．

これらのプロシージャのうち，基本図形や複合図形の組み合わせでパターンを構成し描画するものについては，構成要素とする基本図形・複合図形の集合を予め設定しておき，その中から組み合わせを生成する．また，プロシージャは線分・曲線のみを描画するものと，輪郭線およびその内部を塗りつぶし描画するものの 2 種類に分類できる．後者の塗りつぶし対象となる領域については，シードとしてプロシージャの ID を割り振り，以降のステップでその領域が描画対象となった場合は，その ID のプロシージャを呼び出す．

表 1 には各プロシージャの，(a) ID, (b) 種別（BD:

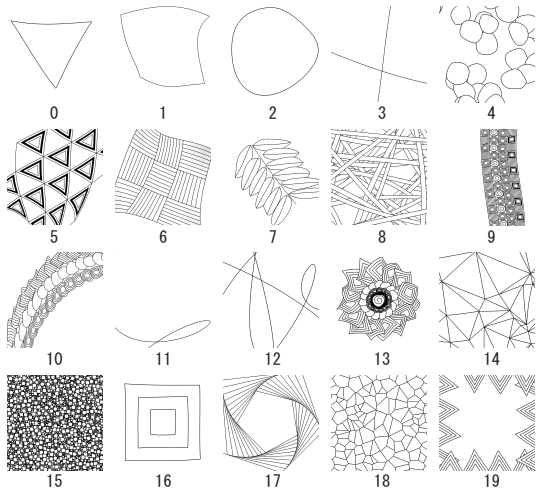

図 13　プロシージャの ID と描画例

基本図形，SP:対称パターン，RP:領域依存パターン)，(c) 対称パターンの場合の配置テンプレートの種類，(d) 描画に用いる基本図形，複合図形およびその他の構成要素，(e) 線のみの描画か，塗りつぶし描画を含むか，(f) 選択できるシードの集合を示す．

ID:0,1,2,3,11 は基本図形 (BD) の 1 つを，ID:4,5,6,7,8,9,10,12,13 は配置テンプレートを用いて対称パターン (SP) を，ID:14,15,16,17,18,19 は領域依存バターン (RP) を描画するプロシージャである．ここでは，これらの ID からなる G_1,$\cdots$, G_6 のグループを用意し，シードの設定の際に利用する．

$$G_1 = \{0, 1, \cdots, 19\},\ G_2 = \{14, 15, \cdots, 19\},$$
$$G_3 = \{2\},\ G_4 = \{0\},\ G_5 = \{1\},\ G_6 = \{16, 18\}$$

以降では，幾つかのプロシージャをとりあげ説明する．

ID0 のプロシージャは基本図形 tr に自由形状変形，アフィン変換を施したあと，輪郭線を黒色で，その内部領域 s を白色で塗りつぶして描画する．領域 s のシードは G_3 の要素である 2 とする．従って，以降の過程で領域 s に描画するときは，ID2 のプロシージャを呼び出す．

ID3 のプロシージャは基本図形 cr の線のみを黒色で描画する．

ID4 のプロシージャは表 1 の (d) に示す 15 種類の基本図形，複合図形 E_1 から 1 種類を選択し，それに自由形状変形を施した後，配置テンプレート MR に従って配置して描画する．各構成要素を描画する際，白色の塗りつぶし領域 s には，G_2 の中からランダムに選んだ ID をシードとして設定する．また，補足 1 に述べるように，1/3 の確率で領域を黒く塗りつぶしてから，パターンを描画する．

表 1　プロシージャの設定

(a)	(b)	(c)	(d)	(e)	(f)
0	BD		tr	線 + 塗	G_3
1	BD		sq	線 + 塗	G_4
2	BD		cl	線 + 塗	G_5
3	BD		cr	線	
4	SP	MR	E_1(補足 1)	線 + 塗	G_2
5	SP	MT	E_2	線 + 塗	G_2
6	SP	MS	E_3	線 + 塗	G_2
7	SP	ML	線分，le	線 + 塗	G_6
8	SP	MR	sq(補足 1,2)	線 + 塗	G_2
9	SP	MA	E_3	線 + 塗	G_2
10	SP	MA	E_1(補足 3)	線 + 塗	G_2
11	BD		E_4 の 1 つ	線	
12	SP	MR	E_4 の重複組合せ	線	
13	SP	MC	E_1	線 + 塗	G_2
14	RP		線分	線	
15	RP		sq, cl	線 + 塗/線	G_1
16	RP		閉曲線	線	
17	RP		線分	線	
18	RP		線分	線	
19	RP		E_1	線 + 塗	G_2

(表中の記号)
$E_1 = \{tr, sq, cl, st, ss,$
$mtr0, mtr1, ptr, msq0, msq1, msq2, psq, mcl0, mcl1, scl\}$
$E_2 = \{tr, st, mtr0, mtr1, mtr2\}$
$E_3 = \{sq, cl, ss, msq0, msq1, msq2, msq3, mcl0, mcl1, scl\}$
$E_4 = \{c0, c1, c2, c3, c4\}$
補足 1: 確率 1/3 で領域を黒く塗りつぶしてからパターンを描画する
補足 2: MR による配置の前に sq を横方向に拡大，自由形状変形，回転させる
補足 3: MA による配置の前に構成要素を自由形状変形，回転させる

ID8 のプロシージャは，ID4 と同様の配置テンプレート MR を用いるが，基本図形 sq を横方向に引き伸ばし，自由形状変形，回転してから配置する．

ID14,16,17,18 のプロシージャはそれぞれパターン $delaunay_triangles$，$contour_line$，$paradox$，$voronoi$ を線描画する．

ID15 はパターン $fill$ であり，基本図形 sq,cl のいずれかを用いて領域を埋め尽くす．このとき，基本図形の輪郭線のみを描画する方法と内部も白色で塗りつぶす方法のどちらかの描画方法を選択する．図 10，図 13(15) の $fill$ はそれぞれ前者，後者の描画方法を用いている．後

者については塗りつぶす領域 s に G_1 から選んだ ID をシードとして設定する．

次に，描画処理の詳細について，例を用いて説明する．ID4 のプロシージャにおいて，図 7 の $mtr1$ がパターンの構成要素として選択された場合，この $mtr1$ をランダムに配置して描画する．$mtr1$ の描画は図 4 の tr を使って次のように描画する．まず，塗りつぶし色を白にして，tr を描画し，次に tr を 0.8 倍に縮小し，塗りつぶし色を黒にして，上書き描画し，さらに，tr を 0.6 倍，0.4 倍縮小し，塗りつぶし色を白にして上書き描画していく．このとき，シードとして設定するプロシージャ ID を G_2 からランダムに選び，画素値を 2 チャンネル（表示色，シード) で表現し，フレームバッファにレンダリングすることで，シード属性を持つ領域の描画を実現している．上記の内部処理におけるフレームバッファおよび出力画像の解像度は 1024×1024 とし，本論文に示す結果の図にはこの解像度の画像を用いている．

5.3 充填対象領域の選択と領域内への描画

先に述べたゼンタングルの充填プロセスでは，充填対象領域を選択し，その領域内にパターンを描画する．この処理は以下のように実現する．なお，初期領域（白紙の領域）のシードはプロシージャの ID とは異なる値を設定，その値をシードとしてもつ領域にパターンを描画する際は G_1 の中から任意のプロシージャを選択できるものとした．

1. 模様全体に対し，白色領域のセグメンテーションを行う．
2. 個々の白色領域について距離変換[*2]を行い，距離値が最も大きい画素をもつ領域 s を選択し，s のシードとして設定されているプロシージャの ID を求め，それを i とする．
3. プロシージャ i を使って，領域 s 内に描画する[*3]．このとき，i の種別が BD,SP ならば，s の外接矩形の中心座標に一定の範囲で揺らぎを与えた c を求め，c を中心に基本図形，対称パターンを描画する．

[*2] 距離変換とは，各白画素について，最も近い黒画素まで距離をその画素値とする画像変換である．

[*3] 選択した領域内のみへの描画は，その領域のステンシルマスクを作成して実現している．

上記の処理を N 回繰り返したときの模様生成過程を図 14 に示す．左から順に，$N = 1, 2, 10, 50, 200$ のときの描画結果である．また，上中下段は乱数により異なるプロシージャが選択されたときの結果である．図 15 には他の自動生成結果を示す．

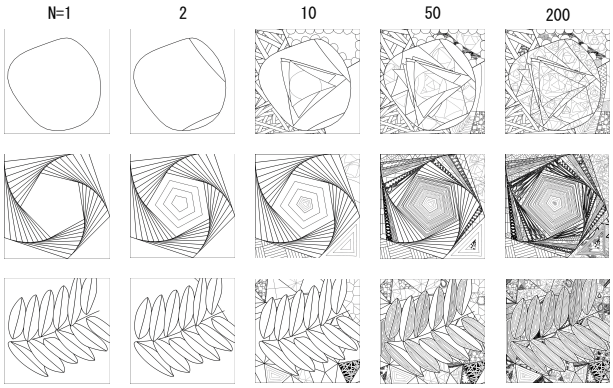

図 14　自動生成過程

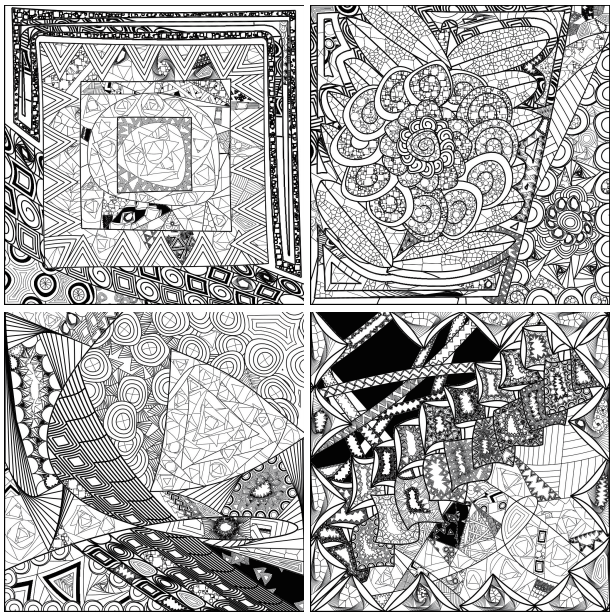

図 15　他の自動生成結果 ($N = 200$)

5.4 制作ツール

上記のシミュレーションは，図 16 左に示す専用のツールを作成して実施した．このツールでは，自動生成を行うための設定として，利用するプロシージャの選択，各プロシージャで利用する構成要素の選択，シミュレーションで用いる乱数生成器の初期値や繰り返し回数 N の設定などが行える．また，下記の機能も実現している．

- ユーザが領域を選択したり，プロシージャを選択して，手動で模様を生成する機能
- 別途用意したモノクロ画像を読み込み，それにパターンを描画していく機能
- ユーザが線を描き込む機能

図 16 にはこのツールを用いてユーザはプロシージャを選択しながら対話的に作成した模様を示す．左は象の輪郭線の描画に約 3 分，内部の模様の作成に約 3 分の時間を要した．右は初期状態として，別途用意した画像を読み込んだものであり，作成に約 2 分の時間を要した．

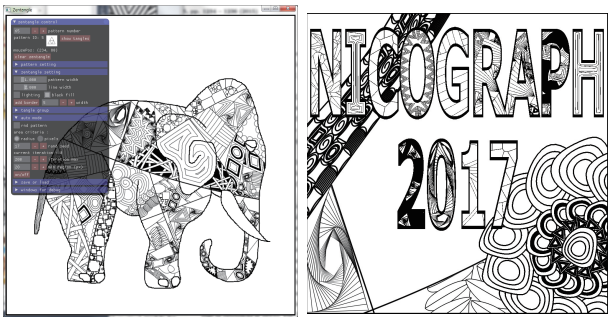

図 16　制作ツールと手動生成例

5.5　ユーザによる評価

自動生成される模様の美しさや制作ツールを用いて作成した模様の満足度について，アンケート調査を行った結果を示す．

アンケートでは，20〜30 代の男女 5 名の非アーティストを被験者 (a〜e) とし，ゼンタングルの 6 作品 (A〜F) を提示し，4 段階評価（1. 美しい，2. やや美しい，3. あまり美しくない，4. 美しくない)，および，どのような点を美しいと感じたかの記述を求めた．6 作品のうち A，B は図 15 上段左，右に示す自動生成結果，C，D は半自動で別途生成した作品，E[*4]，F[*5]はインターネット上に公開されている手書きの作品である[*6]．

4 段階評価結果および記述回答の要約を表 2 に示す．記述回答から美しさの評価基準にバラつきがあることが分かる．また，それが 4 段階評価の結果にも現れているが，全ての被験者が自動生成又は半自動生成の作品のどれか 1 つは美しい，又はやや美しいと評価している．被験者 d，e は細かく複雑なパターンを有するものを美しいと評価しており，被験者 b，c はパターンの複雑さではなく作品の構成やトーンの違いを評価している．

表 2　美しさに関するユーザ評価

	A	B	C	D	E	F	記述回答 （要約）
a	3	1	2	2	1	1	スッキリと見える
b	3	3	4	1	3	3	柄が少なく余白がある
c	2	2	2	1	1	1	模様の組み合わせ
d	2	1	3	3	1	2	細かな作り, 大きい花
e	1	1	2	3	1	2	細かいものが多い

また，2 名の被験者に制作ツールの UI を用いた作品制作を体験してもらい，その後で意見を求めた．「5 分程度で想像に近いゼンタングルが制作できる」という肯定的な意見と，マウスで複雑な形の閉曲線を描き込むとき，少し途切れてしまうと閉じた領域にならず，これについて「難しい」という意見や「Undo など基本的な機能がほしい」などの UI に対する不満の意見があった．

5.6　考察

上述したシミュレーションにより，ゼンタングルの特徴である規則性と不均一性をもつ模様が生成できた．また，限られた図形の組み合わせであるが，乱数を用いたシードの設定や描画回数 N の設定により，多様な模様を自動生成できること，制作ツールを利用してユーザの意図を反映した模様を生成できることを確認した．

一方で，シミュレーションで得られた模様を実際のゼンタングルの作品と比較したとき，美しさや自然さにおいて劣っている場合もあり，その原因について考察する．

5.6.1　シミュレーション方法とユーザ制御に関する課題

自動生成された模様には，パターンの階層的関係に不自然さのあるところも散在している．シミュレーションでは，各描画プロシージャにシードグループを設定し，次のステップで描き込めるパターンを制限しているが，この設定が不適切であることが 1 つの原因である．また，パターンの配置や向き，変形が乱数に依存していることも原因の 1 つであると考えられる．図 17 上段は三角の領域にパターンを充填する例である．本シミュレーションでは b のような充填を行い，c のような領域形状へのマッピングは行っていない．また，自動生成では，最大領域にパターンを描き込むことを繰り返すため，結果的に全体の線密度（トーン）が均一になり，大局的な濃淡コントラストが小さくなる傾向がある．半自動生成の場合には，ユーザの制御により上記のような不自然さのな

[*4] https://www.pinterest.jp/pin/341288477991566415/
[*5] http://bizzine.jp/article/detail/1610
[*6] C,D,E,F はコンテンツファイルを参照

い模様の制作やトーン調整ができることが望ましく，このためにはパターンの配置，向き，変形の制御やスケーリングによるトーンの制御ができる UI を制作ツールに追加する必要がある．

5.6.2 モデル化において考慮すべき課題

自動・半自動生成された模様は，機械的・人工的に生成された印象が残ってしまっている．この原因として，単純な幾何図形のみを用いたことやそれらの配置のバリエーションが不十分であることが考えられる．ゼンタングルの作品では，図 17 下段の d のような，有機的印象のある図形や立体感を表すための図形がよく用いられることから，ユーザがこのような図形要素を作成し，モデルに追加するような改善方法が考えられる．図形の配置については，図 17e の枝分かれするような配置は本モデルでは行っていない．また，kazi ら [21] はエネルギー最小化原理により図形要素間の距離を変え，配置のバラつき具合を制御しているが，本モデルでは乱数でバラつきを与えているのみである．従って，配置テンプレートの種類の追加や配置のバラつき制御について検討が必要である．また，手書き風の印象を与えるためには，個々の線を描画する際に，太さを変化させることや，図 17f のように，線（ストローク）単位で変形することも有効であると思われる．

提案モデルでは，紙面を複数のパターンで埋めることと，階層的な組合せにより多様な模様を生成することをねらいとして，それらを充填的プロセスという 1 つの手順で表現したが，上記のように不自然な模様が生成される場合がある．これらを個別のプロセスとして扱い，それぞれを何らかの評価基準に基づき適切に制御できるようなモデルも今後検討したい．

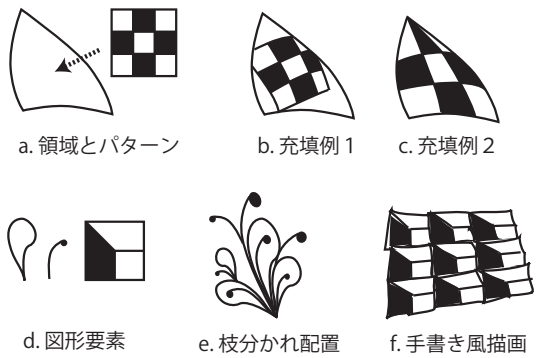

図 17　手書きゼンタングル作品にあるパターンの充填（上），図形の配置（下）

6　まとめ

本論文では，ゼンタングルのモデル化とシミュレーションの方法について述べた．得られた模様はゼンタングル固有の特徴を有することを実験により確認した．しかしながら，実際のゼンタングルの作品と比較すると，美しさや自然さにおいて劣っている点がある．そのため，今後は，上記「考察」で述べた課題を解決するための具体的な方法を検討し，より高品質な模様を自動生成・半自動生成することを試みる．

参考文献

[1] Marie Browning Suzanne McNeill, Sandy Steen Bartholomew. ゼンタングル. ブティック社, 2013.

[2] さとういずみ. はじめてのゼンタングル. 自由国民社, 2014.

[3] René Binet. *From Nature to Form*. Prestel, 2007.

[4] ベノワ・B・マンデルブロ. ラクタル幾何学. 日経サイエンス, 1984.

[5] Herman Weyl. シンメトリー. 紀伊国屋書店, 1970.

[6] Aristid Lindenmayer. Mathematical models for cellular interactions in development i. filaments with one-sided inputs. *Journal of Theoretical Biology*, Vol. 18, No. 3, pp. 280 – 299, 1968.

[7] Aristid Lindenmayer. Mathematical models for cellular interactions in development ii. simple and branching filaments with two-sided inputs. *Journal of Theoretical Biology*, Vol. 18, No. 3, pp. 300 – 315, 1968.

[8] Aristid Lindenmayer and Grzegorz Rozenberg. *Parallel generation of maps: Developmental systems for cell layers*, pp. 301–316. Springer Berlin Heidelberg, 1979.

[9] Gur Harary and Ayellet Tal. The Natural 3D Spiral. *Computer Graphics Forum*, 2011.

[10] Przemyslaw Prusinkiewicz. Applications of l-systems to computer imagery. In *Proceedings of the 3rd International Workshop on Graph-Grammars and Their Application to Computer Science*, pp. 534–548. Springer-Verlag, 1987.

[11] Adam Runions Przemyslaw Prusinkiewicz. Computational models of plant development and form. *New Phytologist*, Vol. 192, No. 3, pp. 549–569, 2012.

[12] Michael T. Wong, Douglas E. Zongker, and David H. Salesin. Computer-generated floral ornament. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pp. 423–434. ACM, 1998.

[13] O. St ava, B. Benes, R. Mech, D. G. Aliaga, and P. Kristof. Inverse Procedural Modeling by Automatic Generation of L-systems. *Computer Graphics Forum*, 2010.

[14] Matthew Kaplan and Elaine Cohen. Computer generated celtic design. In *Proceedings of the 14th Eurographics Workshop on Rendering*, EGRW '03, pp. 9–19, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[15] Li-Yi Wie, Sylvain Lefebvre, Vivek Kwatra, and Greg Turk. State of the Art in Example-based Texture Synthesis. In M. Pauly and G. Greiner, editors, *Eurographics 2009 - State of the Art Reports*. The Eurographics Association, 2009.

[16] Michal Lukáč, Jakub Fišer, Jean-Charles Bazin, Ondřej Jamriška, Alexander Sorkine-Hornung, and Daniel Sýkora. Painting by feature: Texture boundaries for example-based image creation. *ACM Trans. Graph.*, Vol. 32, No. 4, p. 116, 2013.

[17] Jingwan Lu, Connelly Barnes, Connie Wan, Paul Asente, Radomir Mech, and Adam Finkelstein. Decobrush: Drawing structured decorative patterns by example. *ACM Trans. Graph.*, Vol. 33, No. 4, pp. 90:1–90:9, 2014.

[18] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. arXiv:1508.06576, 2015.

[19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. arXiv:1603.08155, 2016.

[20] Pascal Barla, Simon Breslav, Joëlle Thollot, François X. Sillion, and Lee Markosian. Stroke Pattern Analysis and Synthesis. In *Proc. of Eurographics 2006 : Computer Graphics Forum*, Vol. 25 of *663-671*. ACM, 2006.

[21] Rubaiat Habib Kazi, Takeo Igarashi, Shengdong Zhao, and Richard Davis. Vignette: Interactive texture design and manipulation with freeform gestures for pen-and-ink illustration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 1727–1736, 2012.

[22] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, Vol. 20, No. 4, pp. 151–160, 1986.

堀 友香

2014 年中京大学工学部情報工学科入学，現在に至る．CG によるパターン表示の研究に従事．

山田 雅之

1994 年名古屋工業大学大学院工学研究科修了．同年同大学知能情報システム学科助手，98 年中京大学情報科学部助手，99 年同講師，07 年同大学情報理工学部准教授，12 年同教授，13 年同大学工学部教授，現在に至る．人工知能，メディア技術応用の研究に従事．博士（工学）．

中 貴俊

2008 年中京大学大学院情報科学研究科修了．同年中京大学情報科学部助手，2013 年同大学工学部助教，2014 年同大学同学部講師，現在に至る．CG による可視化やモバイル・メディアを活用した教育コンテンツなどに関する研究に従事．博士（情報科学）

宮崎 慎也

1994 年名古屋大学大学院工学研究科修了．93 年中京大学情報科学部助手，97 年同講師，99 年同助教授，07 年中京大学情報理工学部教授，13 年同大学工学部教授，現在に至る．CG，VR の特にソフトウェアによる仮想環境構築，空間操作に関する研究に従事．博士（工学）．