

絵柄の不規則配置による連続文様の生成手法

佐々木 尚孝*

遠藤 善道**

*岐阜県工業技術センター

**岐阜県繊維試験場

〒501-61 岐阜県羽島郡笠松町北及

模様の制作にコンピュータグラフィックシステムが利用されるようになってきたが、模様を構成する絵柄の配置は、デザイナーの手作業と感性に頼っている現状にある。そこでコンピュータにより、手作業を必要としない効果的配置手法の実現を目標として研究を実施している。

本報では、絵柄を配置する際に必要な様々な規則を疑似乱数等によって与えることにより、多様でかつ連続性のある模様の生成手法と、得られた模様の効果について報告する。

The Method of Random Mapping for Textile Design

Hisanori SASAKI*

Yoshimichi ENDOU**

*Gifu Prefectural Industrial Research Technical Center

**Gifu Prefectural Textile Research Institute

Kitaoyobi, Kasamatsu-cho, Hasima-gun, Gifuken 501-61, Japan

We have been using Computer Design System for making textile design. However when we try to place the element patterns of textile design, we depend on designer's manual labor and his sense of beauty.

Then we are studying the method of placing the element patterns with computer. In this study, We report the method of making some rules of placing with pseudo-number, and it's effect.

1. はじめに

服地やカーテン地などのテキスタイル分野、包装紙などの印刷分野、そして壁紙やタイルといった建築内装材分野においては、それらに施される模様が最も重要である。これら分野の製品を製造する業界では、模様の作成にコンピュータグラフィックス（以下CG）システムを利用することが一般的になって来た。

CGシステムにより模様作成が非常に効率的に行えるようになったが、まだ連続模様の作成において欠点を持っている。それは、規則的な繰り返しによる制作は複写機能などにより容易であるが、単純な繰り返しでないように配置する方法が準備されていない点である。現状ではデザイナーがコンピュータ上にもかかわらず手作業で配置検討をしており、複数の絵柄を組み合わせたり、縮小や重ね合わせ効果を検討しようとしても、その試みには限度がある。

そこで、コンピュータにより自動的に絵柄配置を行って様々な模様制作の可能性を広げることと、デザイナーでなくても時代に適合した感性で絵柄作成ができることを目的として、その手法検討と模様制作ソフトウェアを試作している。本報では、模様を構成する個々の絵柄の制作ではなく、あらかじめ準備された絵柄による模様生成手法について、基本的な考え方とそれら手法の適用結果を報告する。

2. 連続模様生成の考え方

模様の制作は、様々な絵柄（伝統文様では紋と呼ばれる）を、一定の矩形領域内に組み合わせたり配置したりして行われる。手作業でなく自動的に配置しようとする場合には、以下の項目の値設定が問題となる。

- ①絵柄の変換規則の設定
- ②絵柄の配置規則の設定
- ③絵柄の重ね規則の設定

これらについて、次に説明する。

2. 1 絵柄の変換規則

模様のバリエーションを得るには、絵柄その

ものを様々に変形することが考えられる。拡大・縮小・変形に加えて、色の変更も考慮するとひとつの絵柄から多種類の模様生成が可能となる。色変更では、明度・彩度変更により同一色相での多様化、色相変更により色彩の多様化ができる。

2. 2 絵柄の配置規則

絵柄の配置を行う前提として、得ようとする模様の一定領域のサイズや形状それにその領域をどのように連続化するかが重要となる。この一定領域を単位として、製品に必要な面積となるよう組み上げられる。この領域の形状は、通常矩形であるが、面を埋め尽くす他の形状で行うことも考えられる。どの形状にしろ組み合わせられた模様を連続的に見せるためには、単位の領域の組み合わせ方から、領域のどの位置が相互につながるのかによって絵柄の配置処理を行う必要がある。

さて、絵柄配置のはじめの問題として、絵柄位置の規則性がある。規則的に見えにくくなる段階として、次の3段階が考えられる。

- ①規則的な配置の中での不規則性が見える：絵柄の位置関係は規則的に配列されているが、絵柄の向きや大きさなどが不規則な場合。
- ②不規則な配置の中での規則性が見える：絵柄の向きや大きさそして個々の位置関係が不規則的だが、全体の分布に法則性がある場合。
- ③不規則な配置としか見えない：絵柄の位置関係も向きや大きさにも規則性がない場合。

これらは、絵柄の変換規則とも関係し、配置方法により不規則性の相乗効果が得られると期待できる。

次の配置問題として、絵柄のどの部分を基準としてどの程度の間隔で配置するかがある。配置基準点としては、図1のように、絵柄の重心位置とサイズの中心位置とが考えられる。配置間隔は、絵柄の配置数量や全体数に関係し、絵柄が重ならない条件や重なる場合は重ね規則とも関係してくる。それに、複数絵柄を用いる場合には、その親近性という意味合いも持ち、あ

る程度デザイナーの意図を入れた多様な模様作成の鍵となると思われる。

2. 3 絵柄の重ね規則

絵柄の重なりを許した場合、個々の絵柄の描画や全体の重なり規則など様々な条件を決める必要がある。個々の絵柄では、図と地の関係を処理したり、複数の絵柄が重なった場合に色の変更を受けない領域（マスク）を設定する必要がある。図2のように、雪の結晶の外側領域が地で内部は図であるが、雪の骨格部分にマスク処理がされた場合、重なって描かれた他の図部分は変化するというものである。

相互の絵柄の重ね量は配置間隔と、重ね方は変換規則と関係する。重ねた場合の色処理とマスク処理の条件設定もある。色変更では色の混合法により、様々な効果が期待できる。

3. 模様生成手法の適用への考え方

模様生成と利用のためには、図3のような操作の流れが必要となる。前述の手法の考え方を適用するためには、それぞれの課題点を明確にする必要がある。ここでは、適用ために必要な項目について述べる。

3. 1 絵柄の前処理

これは、処理するデータ形式を統一したり、配置に必要な情報を付加するものである。

コンピュータで制作された絵柄のデータ形式は、大きく次の2通りに分かれる。

①ラスタデータ：スキャナーなどで読み込みレタッチソフトで修正保存された画像や、ペイント系ソフトなどで描かれた画像で、点の集まりのデータ。

②ベクトルデータ：CADやドロー系ソフトにより描かれたもので、図形の輪郭や塗り情報を点と線分それに数式などで持ったデータ。

これらのデータ形式は様々であり、配置に必要な情報を付加することを考慮すると、扱うデータを専用データに変更するか、付加情報のみ別ファイルで持つ必要がある。

配置に必要な情報とは、様々の規則も含むが

絵柄の配置関係や重なりを判定するための境界をつくる処理が前処理として必要となる。

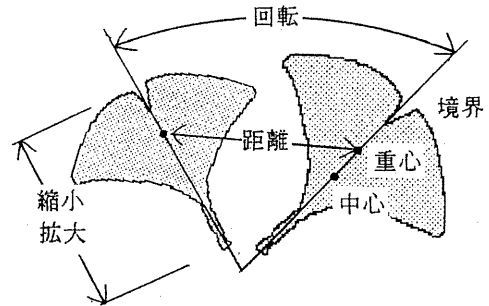


図1 絵柄作成を制御する項目

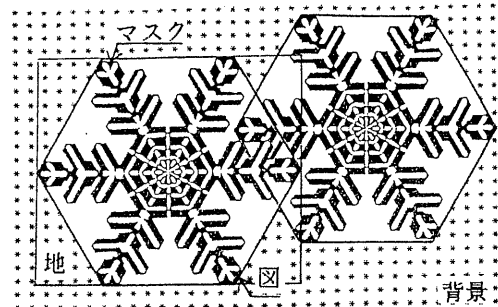


図2 図と地およびマスクによる重ね処理

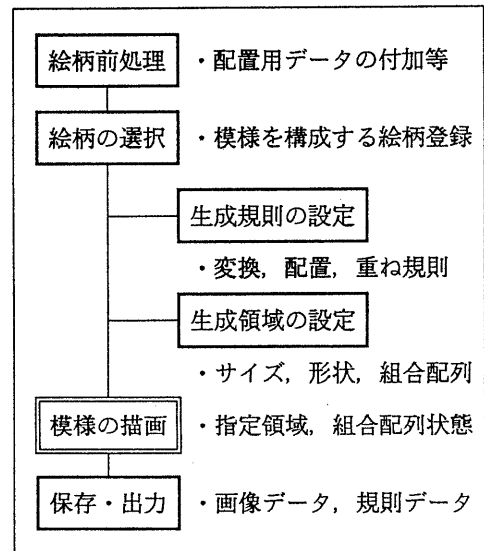


図3 連続模様の描画までの操作流れ図

3. 2 生成規則の設定

選ばれた絵柄は、変換や配置そして重ねなど各種の規則を、コンピュータでも人でも任意に与えることができる必要がある。また、デザイナーが設定する場合には、数値ではなく“密になった”－“ばらばらの”など間隔的な言葉で表現された尺度での入力方法が望ましい。

3. 3 生成領域の設定

この場合の領域は、ディスプレイ上の領域を意味するのではなく、実際に必要となる単位領域を意味する。ただし、描画はディスプレイ上のため、実寸法との関係付けが必要となる。

また、連続模様を描画であるため、その領域が繰り返し使われる組合せ配列の仕方によって描画される画像が異なる。方法としては、基準となる領域において隣接する領域の位置を定義して同じ描画を行うやり方と、接続する辺同士の間隔を記述して連続する領域として扱う方法が考えられる。

3. 4 表示・保存・出力

表示は、指定領域とその領域を配列規則に従って展開した状態の表示が必要となる。また、制作画像が実際に必要な解像度であればそのまま保存してもよいが、メモリーの関係でできない場合はファイル等への出力により指定解像度を得る工夫が必要である。また、コンピュータが自動的に設定した情報を含めて、模様制作に必要なデータを全て一括ファイル保存することにより、気に入った連続模様をいつでも望みの解像度で得られる利点がある。

4. 試作ソフトと模様生成上の課題

手法の適用を検討するため、基本的なソフトウェアを試作して効果を確かめた。使用した機器は、Macintosh (Apple Computer) を画像データを中心にして用い、ベクトルデータの図形は Indigo2 (Silicon Graphics) を利用した。

4. 1 画像データの配置による模様生成

Macintoshの ColorQuickDraw 機能を利用

してソフトウェアを試作し、図4と図5に示すように配置効果を確かめた。

配置される絵柄は、事前にregionを持つ画像として変換し、選択し登録される。登録された絵柄はランダムに選ばれ、回転角度と位置も疑似乱数を用いて配置される。重ね描きを禁止すれば、regionを判定して図5のように重ならない状態で配置される。重ね描きには、重なり部分の色変更規則を8種類準備した。

4. 2 ベクトルデータによる模様生成

模様表示には、ハードウェア専用のグラフィックライブラリーを用いた。絵柄データは、点と線分による図形表現形式のみを扱い、2次元図形も3次元として処理した。従って、変形効果は、3次元空間のXYZ軸における拡大縮小と回転操作で得られるようにした。なお、投影方法は、連続性が必要なため正射影である。また、重ね描き禁止処理のため、図形原点は奥行きのZ値を同じ位置としている。

図6と図7に作成例を示す。

4. 3 乱数による描画結果と課題

変換規則や配置規則そして重ね規則を様々に設定することにより、手作業では困難なバリエーションが実にたやすく生成できた。特に、配置する絵柄の数が多い場合は、コンピュータの支援は必ず必要であると感じた。図4は、重なりを許可して80の絵柄をオーバーライトで描いたものである。連続化した表示でも、規則性が見えにくかった。重なる絵柄の種類や数を多くすると、模様の複雑性がさらに増加した。図5は重なりを禁止し、密な状態での表現を得たものである。表示数が多い場合は規則性が見えにくい、少ないと何らかの規則性が見えた。

規則的な配置でかつ変換規則も適用せず重ね規則だけを行うと、伝統的模様のように整った模様が得られる。この状態に、重ね量が大きくなるように変換規則を適用すると、規則配置にもかかわらず規則的には見えにくくなった。図6はその例で、10行10列に規則的に配置された立方体を乱数を用いてアフィン変換と回転を行

ったものである。一方図7は、30個の立方体を全て乱数で配置決定したものであるが、方向性が見えている。表示数が少ない場合には、連続させると何らかの規則性が見えてしまう傾向にあった。表示数が少なく規則性がわからないためには、繰り返し単位の領域が視野角度より大きくないと困難であることがわかった。そのため、実際と同じ状態での確認が必要となるが現実サイズでの見え方をディスプレイ上で縮小表示された模様から想像することは不可能に近い。正確な確認には、より実サイズに近いスクリーンへ投影するか、ヘッドマウントディスプレイなどにより仮想空間で見る必要がある。プリプレスによる確認では、時間とコストの問題があり案の絞り込みが必要となる。

大きな課題としては、模様は多量に生成されるがデザイナーの意図や時代の感性に適合しているかどうかという問題である。“規則的—不

規則的”“複雑—シンプル”など感性を表現する言葉と模様生成に必要な各種規則の値との関連を付け、さらに操作を通じて好まれる状態の値を記憶することにより、コンピュータが絞り込みをして候補となる模様を生成できることが求められるだろう。

5. まとめ

手作業に頼らない効果的模様生成手法を確立するため、まず絵柄の変換規則・配置規則・重ね規則など制御すべき項目を明確にした。そして、乱数によってこれらに数値を与えることにより様々な模様の生成を行った。その結果、構成する絵柄の種類や数が多い場合は、手作業では困難な不規則配置が簡単に行え、かつ各種規則を様々に制御することにより多種多様な模様が生成できた。乱数を用いても必ずしも不規則的に見える模様は得られないが、各種規則を制

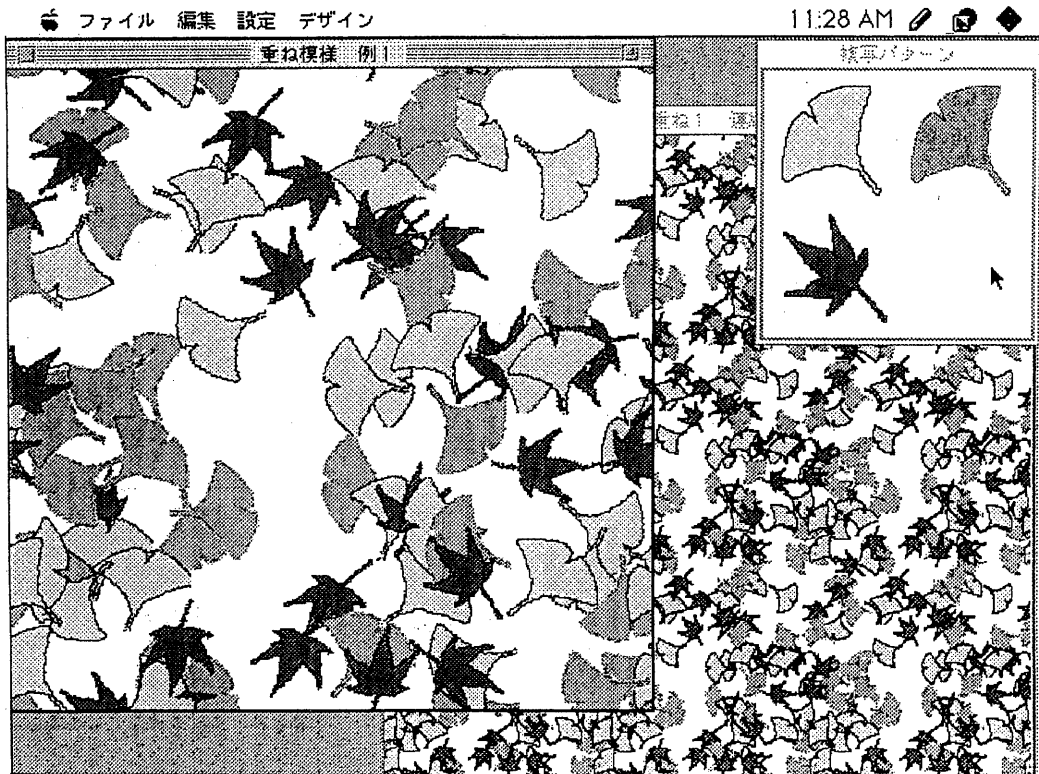
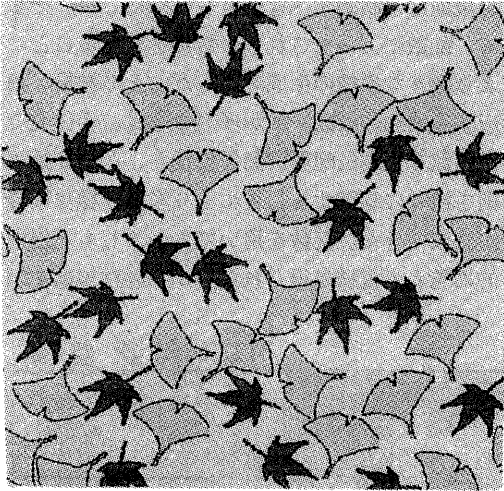


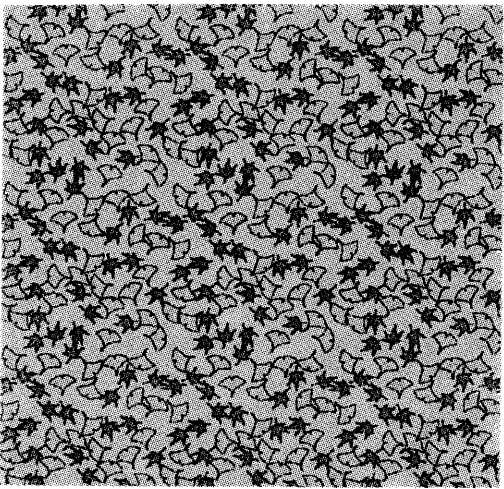
図4 画像データの配置による模様生成例：重ね許可（色変更は重ね描き）

御することにより、規則的から不規則的に至る中間の状態の模様を多量に生成可能であるとわかった。

今後の課題として、操作者の意図をよりの確に伝えることのできるように、規則の数値入力や制御範囲の設定手法を検討する点とともにコンピュータ側が乱数の制御範囲を学習することにより、あまり美的感覚から離れた模様を生成しない仕組みを検討する点があげられる。

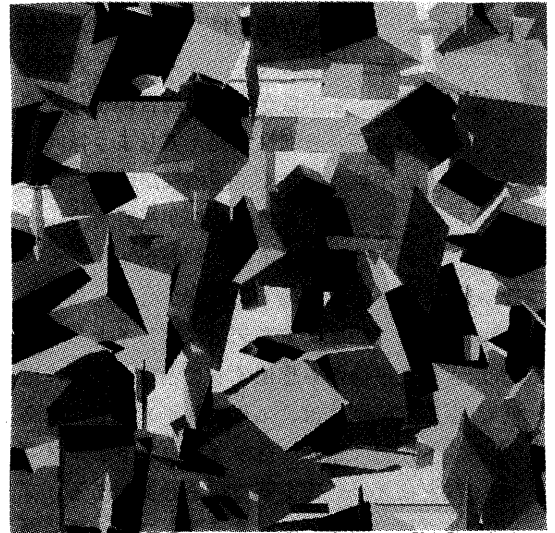


(a) 単位領域の描画

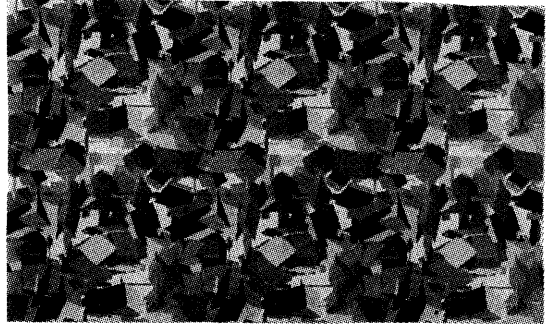


(b) 隣接領域との連続化表示

図5 重ね禁止の描画例



(a) 単位領域の表示



(b) 連続化した状態の表示

図6 規則配置に変換規則に乱数適用例

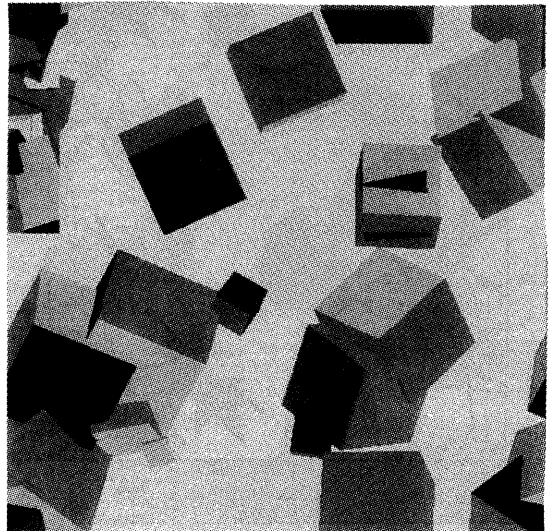


図7 サイズと回転と位置への乱数適用例