

タイトル	タイルパターンデザインシステムの開発
著者	菊地, 慶仁
引用	北海学園大学工学部研究報告, 36: 113-120
発行日	2009-02-20

タイルパターンデザインシステムの開発

菊地 慶仁*

Development of Tile Pattern Design System

Yoshihito KIKUCHI*

Abstract

Tile is “A thin, flat or convex slab of hard material such as baked clay or plastic, laid in rows to cover walls, floors, and roofs”¹⁾. In general, tile has a simple diagram pattern or mono color. However, allocated at four directions and its variation can make a colorful design pattern. In this paper, we report about a development of tile pattern design system.

1. はじめに

タイルは、焼結させた粘土もしくはプラスチックのような硬質の素材で作られた薄型で平坦もしくは凸多面体で、壁、床もしくは屋根を覆うように列を作って配列される¹⁾。タイルは、一般的に単純な図形パターンもしくは単色で作られる。本論文では、このタイル1枚のデザインを支援するシステムの開発について報告する。

タイル画は、日本では公衆浴場の壁画に代表されるように複数の単色タイルを組み合わせたモザイク画が一般的であった(図1)²⁾。またポルトガルには彩色されたタイルを組み合わせた: azulejoと呼ばれるタイル画(図2)がある³⁾。

タイル画では、単純な図案のタイルでも設置する方向を組み合わせることによりタイルの様子が連続することで幾何学的なパターンを作りだすことが可能である(図3)。本研究では、このようなパターンを作りだすためのピットマップ画像の編集を支援するシステム開発について報告する。

* 北海学園大学工学部電子情報工学科

* Department of Information and Electronics Engineering, Faculty of Engineering, Hokkai-Gakuen University



図1 日本におけるタイル画



図2 ジョルジェ・コラス作のタイルによるパネル『A Ala dos Namorados』³⁾

第2章では、タイルパターンをデザインを対象とした時に既存のソフトウェアで対応可能かどうかを検討し、必要と考えられる課題をまとめる。第3章では、システムの具体的な開発について報告し、4章、5章で考察及び結論とする。

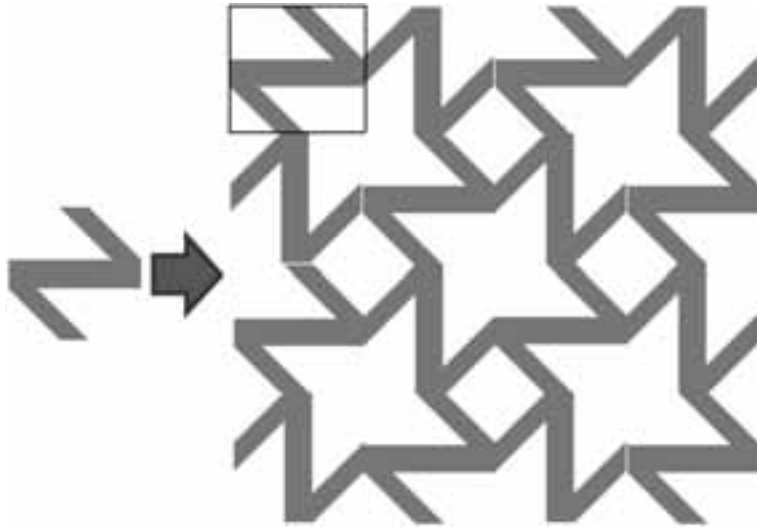


図3 シンプルなタイルを組み合わせて作られたパターンの例

2. タイルパターンデザインシステムの概要

2.1 既存のアプリケーションシステムを用いる場合の問題点

図形描画用のアプリケーションシステムは商用／非商用を問わず多数存在している．本節では既存のシステムでの開発を考慮した場合の課題について考察する．図形描画用アプリケーションは一般的には次の2種類に大別することができる．

1) 複数の階層でドット単位でイメージを編集するペイント系ソフトウェア：

Microsoft社製のPC用オペレーティングシステムWindowsに組み込まれている「ペイント」、写真イメージに対する処理に特化したAdobe社の「Photoshop」などが一般的に知られている．基本的な描画の対象は、階層として管理される複数のビットマッププレーンであり、マスクパターンを設けることはできるがプレーン上に特定のオブジェクトを配置し操作することはできない．

2) 図形を線画として描画し内部を塗りつぶすドロー系ソフトウェア：

Corel社のDrawが良く知られている．また、プレゼンテーション用のMicrosoft社のPowerPoint、図面作図用のAutodesk社製AutoCadなども同様の目的で使用することができる．描画オブジェクトは拡大縮小が可能なベクターデータ（輪郭点の集合）として扱われることが多く、この場合は拡大縮小操作を行っても図形の斜め線（ジャギー）が目立つことが無い．

タイルとして配置した全体的な画像を確認するためには、編集したタイルのパターンを格子状に配置する必要がある、その向きは90度単位に個別に指定できる必要がある．

タイルのパターンをデザインするために対象を1ドット単位で修正することが必要であるが、ペイント系のソフトウェアはタイルに相当する小部分を個別に配置し回転させるためには、タイル1枚1枚を別のレイヤーに分ける必要がある。従って本研究では、ビットマップとしての個別タイルのパターンデザインと配置及び確認とを両立させる必要がある。

2.2 開発の方針

第1章で述べたように、タイル画にはいくつかの方式があるが、ビットマップ系もしくはドロー系のどちらかの方式で全ての必要性を満たすデザインシステムは不可能と考えられる。そこでタイルパターンのデザインシステムを開発するに当たって、本研究ではデザイン作業に幾つかの方針を設けてシステムの開発を行った。

- 1) タイルの形状は正方形で全て同じサイズとする。
- 2) タイルの配置は規則的な格子状とする。
- 3) タイルは単一のパターンとするが、同じ背景色を持った無地のタイルも使用可能とする。
- 4) マウスのクリック操作でクリックした位置のタイルの向きを変更できるようにする。
- 5) タイルパターンの編集機能はWindows内蔵の「ペイント」で用い、デザインシステムは特定の配置上での向きの指定やタイル画全体の確認の作業を主に担当する。
- 6) タイルパターンを修正してファイルとしてセーブした後に、配置された全てのパターンを一括して更新する機能を設ける。
- 7) 配置された各タイルの向きの情報、及び完成した図をセーブする機能を設ける。

3. タイルデザイン支援システム

3.1 開発環境

本研究で開発したシステムについて報告する。Microsoft社が開発したオブジェクト指向に基づくコンピュータ言語である⁴⁾C#を用いた。C#は、構文的にはC++によく似た構造を持っており、実行処理ではSun Microsystems社のjavaと良く似たメカニズムで、コンパイラによって共通言語基盤(Common Language Infrastructure, CLI)と呼ばれる解釈実行系で実行可能な中間言語に翻訳されて実行される。

本来はXML WebサービスやASP.NETの記述を目的に開発され、Net実行環境をインストールしていなければ実行できない、図形处理的分野ではMicrosoft社の3次元グラフィックス環境DirectXのラッパーライブラリのManaged DirectXが用意されている。DirectXは比較的低水準の機能を提供しているために、高度な機能を実現可能であるが、逆にプログラミングで開発するコードサイズと作業量が増加する弊害がある。ラッパーライブラリでは高水準なインターフェースを持たせるために提供されている。Managed DirectXはCLI上で実行可能な全ての言語に簡便なプログラミングを提供してくれるために、従来はC++言語以外では難しかったDirectXを

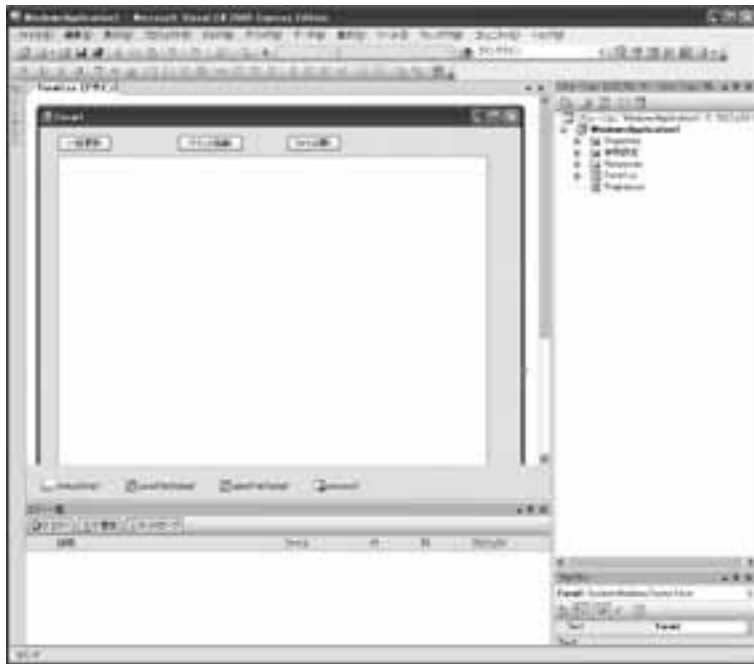


図4 Microsoft Visual C# 2005 EXPRESSによるフォーム（ウインドウ）デザイン

用いたグラフィックプログラムが開発しやすくなっている。

開発にはフリーで提供されている開発環境のMicrosoft Visual C# 2005 EXPRESSを用いた。Visual EXPRESS環境は、フォームエディタを用いることで基本となるウインドウにGUI部品を配置し、該当するイベントに対応するソースプログラムを記述することで開発が比較的容易に行える。図4に開発中のウインドウ画面を示す。図は、フォームエディタを起動してメインのウインドウ、そのウインドウ上に3つの操作用のボタン、ひとつの描画用の画面を配置している。ボタンは特定の用途を直接起動するために用意しており、このシステムでは（1）windows付属の「ペイント」の起動、（2）画像ファイルを開いての読み込み、（3）読み込んだ画像をあらかじめ決めておいた向きでの再配置、に用意している。また一般のWindows用アプリケーションに用意されているプルダウンメニューも用意している。

3.2 デザインシステムの実行

図5に実行中の画面を示す。この図は、「ペイント」で作成したタイルパターンと、それが開発されたデザイン支援システムでどのように配置されているかを示している。プログラムは以下の手順で実行されている。

- 1) 起動した時点では、タイルパターンのイメージは空のまま、配置された時の各タイルの向きに関する情報はすべて初期化されている。この状態で画面をクリックしても回転情報は変更されるが画像は読み込んでいないので効果を画面上に示すことはできない。

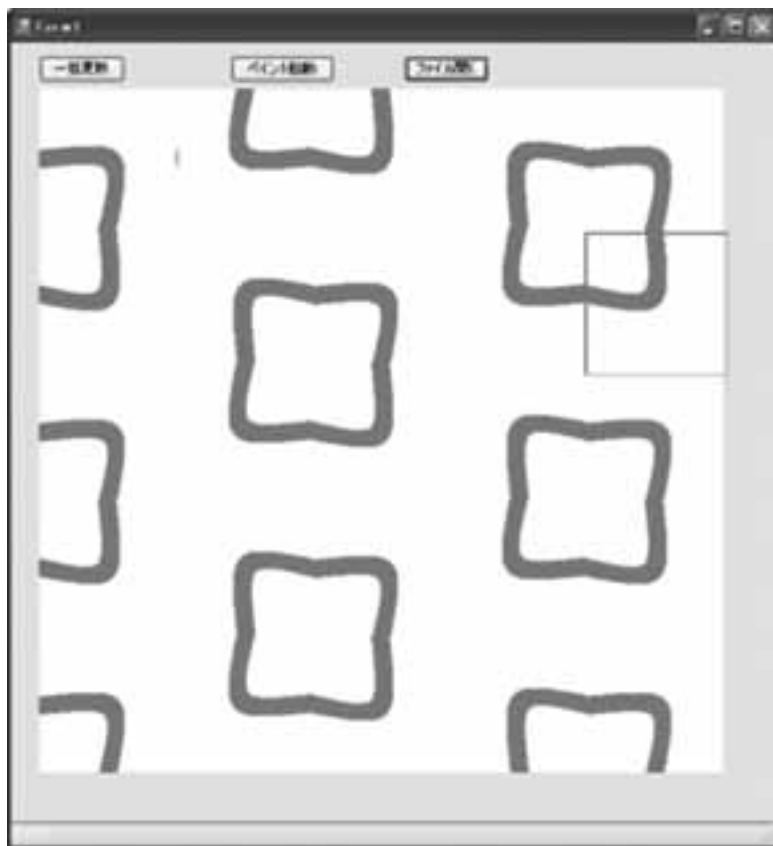
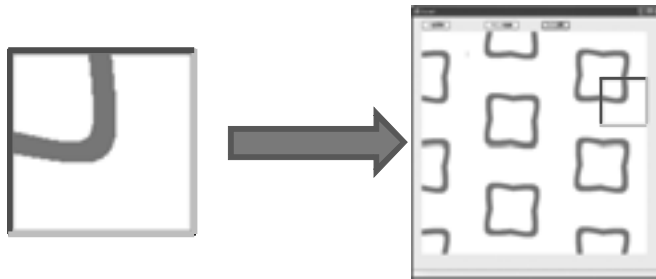


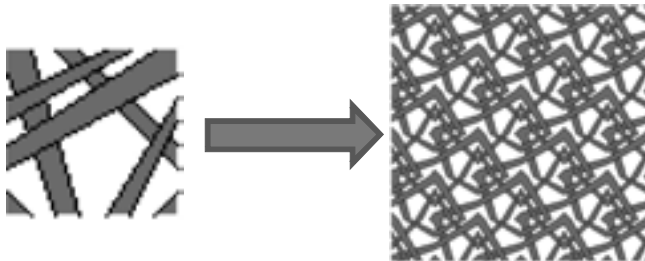
図5 実行中のウィンドウ

- 2) ウィンドウ上に配置したボタン「ファイル読み込み」をクリックして、ファイルダイアログを呼び出し、読み込む画像ファイルを指定する。
- 3) ウィンドウの描画エリアをクリックすると、回転角度が90度ずつ増加され、タイルが左に回転するように描画される。
- 4) 全体としての効果を確認しながら、「ペイント」でタイルパターンを微調整しながら、ウィンドウ上のボタン「一括更新」を押すと、同じファイル名の最新のタイルパターン画像を読み込み、ウィンドウ上で再描画する。
- 5) 必要に応じて角度情報をテキストデータとしてセーブする。タイルの向きの配置を再利用することが可能であるし、手動で向きのパターンを用意しておくことも可能となる。

描画領域のクリックと再描画は、クリックした座標を元に、該当するタイルの番号を同定し、その向きを90度左に回転させるとともに、当該領域だけを再描画することで行っている。



(1) 2辺を同じパターンにするデザイン方式



(2) 4辺が同じパターンのデザイン方式

図6 タイルパターンのデザイン的方式

4. 考察

タイルのパターンは、「ペイント」での描画とデザイン支援システム上での全体配置を比較しながら決めて行った。実際には図6に示したように、2辺～4辺の境界を同一のドットパターンに予め固定して内側の部分をドットで描画する方法を取った。

しかしながら、全体の再描画は一括で行えるもののドット単位の描画に時間がかかり、無作為に与えたパターンが連続するように画像を調整する機能や、より大きな画像から半自動的にタイルパターンを切り出すような機能が必要と考えられた。

また、実写画像の一部をタイルパターンとする際に、画像のトリム範囲を変更しながら同時にタイルパターンの変化を見る機能なども必要と考えられる。

5. 結論

本研究では、タイルパターンのデザインを目的として以下の項目を報告した。

- 1) 既存のアプリケーションシステムの機能を検討して、タイルパターンのデザイン支援に必要な機能を検討した。
- 2) 外部で作成したドットイメージを読み込んで指定した部分を順次回転させて全体のパターンを確認するタイルパターンデザイン支援システムを開発した。

今後必要な機能として以下の項目が挙げられる。

- 1) ドットイメージの編集を一般的な外部プログラムに任せるだけでなく、連続画像となる制約を持たせた画像編集機能の開発
- 2) 実画像などからのタイルパターンの切り出しとタイルとして配置したシミュレーション機能

参考文献

- 1) The American Heritage Dictionary of the English Language, Third Edition (アメリカン・ヘリテージ英英辞典 第3版) copyright © 1992 by Houghton Mifflin Company. Electronic version licensed from INSO Corporation. All rights reserved.
- 2) <http://www.jin.ne.jp/ichifuji/jp/history.html>
- 3) <http://ja.wikipedia.org/wiki/%E3%82%A2%E3%82%BA%E3%83%AC%E3%83%BC%E3%82%B8%E3%83%A7>
- 4) 植田政美, “ひと目でわかるMicrosoft Visual C# 2008アプリケーション開発入門”, 日経BPソフトプレス