グラフィクスとCAD 100－9
（2000．9．8）

CGによるマーブリングテクスチャの生成

鈴木敏和* 茅暁陽** 今宮淳美**
*山梨大学大学院工学研究科電子情報工学専攻
**山梨大学工学部コンピュータ・メディア工学科

印象的で，綺麗な模様を作り出す伝統的技法であるマーブリングは，芸術的な作品だけでなく，我々が日常よく目にするような包装紙などの工業デザインとしても重用されている．現実のマーブリングにおける溶液の流動は，本来的に不可逆操作であり，一度できてしまった不本意な模様を元に戻すようなことはできない．そこで，所望の模様を得るには，試行錯誤が繰り返しおこなえるコンピュータ上での処理が有用である．本稿では，マーブリング製作に必要な様々な要素を考慮して，マーブリングという特有な問題についてナビエ・ストークス方程式を数値的に解くことによりＣＧマーブリングテクスチャを生成する手法を提案する．

Generating marbling texture with CG

Toshikazu Suzuki* Xiaoyang Mao** Atsumi Imamiya**
*Ymanashi University Graduate School of Eng.
**Ymanashi University Department of Computer and Media Engineering

For centuries, marbling has been loved by peoples all over the world both as an art and a useful texture generation technique. Today, we can see marbled designs on everything from jewelry and scarves to greeting cards and tissue boxes. This paper proposes a new technique for generating marble textures with computer graphics. Marble textures are generated as the results of moving colored particles along the 2D vector fields obtained by numerically solving the Navier-Stokes equation of the marbling process. A simple user interface is also provided for allowing users to interactively design their combs and apply marbling operations.

1. はじめに

印象的で綺麗な模様を作り出すマーブリング(marbling)は，古来より世界各地で広く知られている伝統的技法の一つである．マーブリングによって生成される模様は，芸術的な作品としてだけでなく，我々が日常よく目にするような包装紙やポストカード，本あるいは衣類やスカーフの柄などの工業デザインとしても重用されている．現実のマーブリングにおける溶液の流動は，本来的に不可逆操作であり，一度できてできてしまった不本意な模様を元に戻すようなことはできない．効果的な模様を生成するためには，溶液の粘性，顔料の性質，櫛の形状や攪拌時の軌跡・速度，写し取る紙の材質等，多くの事項を調節しなければならず，手間のかかる反復作業が強いられる．

本稿の目的は，CG マーブリング生成システムの開発である．このシステムにより，所望のテクスチャを得るまで，試行錯誤の過程を効率

化することができ，また，CAD/CAMのシステムに組み込むことにより，デザインしたテクスチャを各種の製品に直接的に利用できるという利点がある．

また，本システムは物理法則に基づくシミュレーションをおこなっているため，既存の，フラクタルを用いるシステムや，ある模様専用の関数を用意するシステムでは不可能であった，実際のマーブル模様に近い多種多様な模様の生成が可能である．

2. マーブリングの概略

実際のマーブリングの生成行程は，以下のような3つの基本ステップから構成される [1].

2.1. 小石模様の生成

粘性のあるゴム溶液に，絵の具などの数色の顔料を適当に落として表面を埋め，無数の丸い小石が飛び散ったような「小石模様」を作る．

2.2. 溶液の攪拌

等間隔に釘を打ちつけた櫛や細い棒などを用いて液面上をゆっくりかき混ぜることにより、様々な模様を液面上に作り上げる．

2.3. 転写

模様を壊さないように液面を固定させた後，模様を紙に写し取り，紙面を水洗いして乾かす．

3. CGマーブリング作成手順

実際のマーブリングでは，櫛や棒を動かした時に溶液中に流れ場が生じる．この流れ場に沿って溶液中の顔料が移動することにより小石模様が変形を受け，マーブリング特有の模様が生じる．顔料は溶液表面に薄く漂っていることから，マーブリングは2次元の問題として近似する事ができると仮定する．この仮定から，溶液表面には顔料の色を持つ粒子が広がっていると し，櫛や棒の動作によって生じる流れ場を記述できれば，その流れ場に沿った色付きの粒子の移流現象として模様の攪拌をシミュレートできると考えられる [2].

CGマーブリングの作成手順を以下に示す．

1. 小石模様のテクスチャの生成
2. 櫛による溶液の攪拌をシミュレートし，ベクトル場を生成
3. ベクトル場に基づく小石模様の変形

このうち手順1では，既にできあがった小石模様の画像をスキャナで取り込むこととする．手順2では，流体の運動を支配している非圧縮性ナビエ・ストークス方程式（以下，ナビエ・ストークス方程式）を用いて，ベクトル場を生成する．手順3は，手順2で求めたベクトル場にしたがって入力画像のピクセルを移動することにより，小石模様を変形する．

3.2. ナビエ・ストークス方程式

流体は運動量およびエネルギー保存則にしたがって流動する．この保存則を厳密に定式化すれば，それは現象を支配する方程式となり，これを数値的に解くことで，現象を理論的に予測できる．粘性流体の流れを支配する流体の運動方程式をナビエ・ストークス方程式という．

以下にナビエ・ストークス方程式 [3,4,5] を示す．

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \tag{1}$$

$$\frac{\partial u}{\partial t}+\frac{\partial u^2}{\partial x}+\frac{\partial uv}{\partial y}=g_x-\frac{1}{\rho}\frac{\partial p}{\partial x}+\upsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) \tag{2}$$

$$\frac{\partial v}{\partial t}+\frac{\partial uv}{\partial x}+\frac{\partial v^2}{\partial y}=g_y-\frac{1}{\rho}\frac{\partial p}{\partial y}+\upsilon\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right) \tag{3}$$

ここで u, v はそれぞれ x, y 方向の流速，p は密度，ν は動粘性率，g_x, g_y はそれぞれの方向に作用する外力である．式（1）は質量保存の法則を表わし，連続の式と呼ばれる．これは任意の点における流入する速さの和

と，流出する速度の和が等しいことを示す．また，式(2)と式(3)は運動量保存の法則より導かれ，左辺は局所的な流体加速度と対流に由来する速度変化を表わす．右辺は重力，もしくは他の様々な力による加速度と，圧力変化による加速度，そして，動粘性による抵抗を表わしている．

このナビエ・ストークス方程式に適当な境界と初期条件を設定することによって，流体の現象をシミュレートする．

4. ナビエ・ストークス方程式の数値的解法

時間と空間を離散化し，数値積分によって u, v を求める．

4.1. 時間と空間に対する離散化

ここでは，オイラー法を用いてナビエ・ストークス方程式を時間的に離散化する．式(2)と式(3)の右辺をそれぞれ関数 f, h とおくと時間の離散化により以下のようになる．

$$\tilde{u} = u + \delta t \cdot f(u, v, p, g, t) \qquad (4)$$

$$\tilde{v} = v + \delta t \cdot h(u, v, p, g, t) \qquad (5)$$

また，空間的に離散化するには，トレーの空間を n×m の2次元格子に分割し、それぞれのセルに，図1に示すように速度・圧力と定義する．

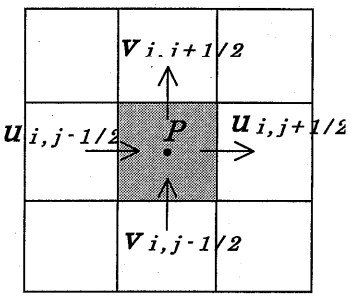

図1：圧力と速度の定義

オイラー法による時間的離散化，格子状に区切ることによる空間的離散化をおこなうことにより，式（2）は以下のように離散近似される．

$$\begin{aligned}\tilde{u}_{i+1/2,j} = u_{i+1/2,j} &+ \delta t\{(1/\delta x)[(u_{i,j})^2 - (u_{i+1,j})^2] \\ &+ (1/\delta y)[(uv)_{i+1/2,j-1/2} - (uv)_{i+1/2,j+1/2}] \\ &+ g_x + (1/\delta x)(p_{i,j} - p_{i+1,j}) \\ &+ (\upsilon/\delta x^2)(u_{i+3/2,j} - 2u_{i+1/2,j} + u_{i-1/2,j}) \\ &+ (\upsilon/\delta y^2)(u_{i+1/2,j+1} - 2u_{i+1/2,j} + u_{i+1/2,j-1})\end{aligned} \qquad (6)$$

式（3）も同様に離散近似される．

4.2. 条件の設定

シミュレーションをおこなう上での各条件は次のように設定する．

初期状態では溶液中の顔料は静止しているため全てのセルにおける速度は0とし，圧力は0でない一定の値とする．

顔料を移動させる外力はユーザがおこなう櫛の攪拌によると考える．この時，櫛の太さは無視し，櫛のある点に外力が加わると仮定する．

境界条件は以下の2つが考えられる．

1. 境界上では境界と垂直をなす方向の速度は0
2. 2つの向かい合う境界上のそれぞれの境界と垂直をなす方向の速度は等しい

境界条件1では境界上ではその境界を越えて溶液が飛び出すことはなく，顔料の回り込みが生じる．すなわちこれは実際のマーブリングの境界を表わしている．一方，境界条件2では，境界を越えた流れは向かい合う境界に流れ込む．よって，この条件を用いれば，シームレスタイリング可能な周期的テクスチャを生成できる．

4.3. 速度修正

以上のように離散化し、初期条件、外力および境界条件の下で式（2），（3）を数値的に解くことにより u, v が求まるが，さらにこの u, v は式（1）を満たす必要がある．

式（1）を用いて速度を修正するには，まず，u 方向の流体の流入，流出の差と，v 方向の流入，流出の差の和を表わす D を以下の式（7）にしたがって求める．

$$D_{i,j} = -((1/\delta\ x)(u_{i+1/2,j} - u_{i-1/2,j}) + (1/\delta\ y)(v_{i,j+1/2} - v_{i,j-1/2})) \quad (7)$$

D が正ならば，流入が流出よりも大きいことを示すので，流体がセルの中へ流れ込み圧力は増加する．逆に，D が負ならば，流出が流入よりも大きいことを示すので，流体がセルの外へ流れ出て，圧力は減少する．

よって，圧力に対して，D に比例した値を加える必要がある．圧力変化は以下の式（8，9）によって与えられる．

$$\delta\ p = \beta\ D \quad (8)$$

$$\tilde{p} = p + \frac{1}{\rho}\delta\ p \quad (9)$$

ここで β は比例定数である．

さらに，圧力が増加すれば，流出速度が増加し，低くなれば流入速度が増加するので，圧力変化に比例した量を式（10，11）のように各速度に加える．

5. 画像変形処理

ナビエ・ストークス方程式を解くことで求めた速度ベクトル場 u により各セルのピクセルの移動先セルが決まる．よってこのベクトル場の流線に沿ってピクセルを移動させていくのだが，この方法では，異なる地点から出発したピクセルが同一のセルに移動するという現象が生じてしまうことがある．このとき，どこからもピクセルが移動してこないセルが存在してしまう．

この問題を回避するために，各セルの逆方向ベクトルの流線に沿って，ピクセルの移動元を辿っていく．これにより，出力テクスチャの各セルの色が，入力テクスチャのどのセルの色であるかを求めることができる．

6. 実装

前節までに述べた方法を実装し，Visual C++6.0 を用いてマーブル模様テクスチャを生成する Windows 用アプリケーションを開発した．

6.1. 初期設定

ユーザは実際に溶液攪拌の作業を行う前に攪拌時間の設定と，動粘性率の設定をおこなう．

攪拌時間を増加させることにより顔料の移動量も増加し，実行時間も増加する．動粘性率を増加させることにより外力の影響を受ける範囲が広がり，攪拌時間同様実行時間も増加する．

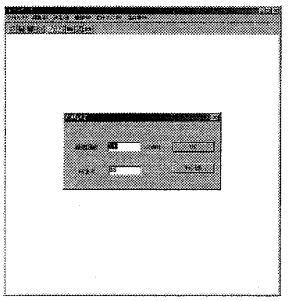

図 2：初期値の設定

6.2. 外力の入力

ユーザによる外力の設定方法は，主に櫛目模様（7.1 節参照）を作る平行外力と，渦巻き模様（7.2 節参照）などを生成する自由外力の 2 つの方法が選択できる．

1. 平行外力は櫛の本数，もしくは櫛と櫛の間隔によって外力を設定する．そして，外力の方向を上下左右の中から選択する（図 3）．
2. 自由外力は，画像領域内でマウスをドラッグすることにより，その軌跡を外力の位置と方向として設定する（図 4）．

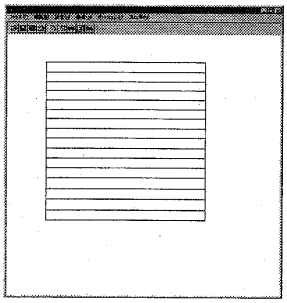

図３：平行外力

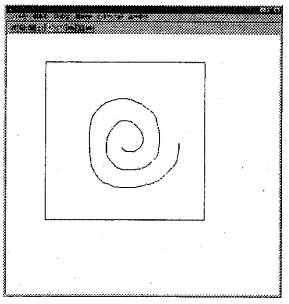

図４：自由外力

6.3. 作業の取り消し

外力を設定し，得られた結果がイメージしていたものと異なっていた場合，その作業を取り消す（undo）ことができる．

子の機能を活用することで，所望の結果を得るまで何度でも試行錯誤が可能となる．

7. 実行結果

今回製作したアプリケーションを実行し，生成されたマーブル模様を示す．

7.1. 櫛目模様

図６は櫛目模様を生成した結果である．櫛の方向とその本数は，以下のとおりである．

実行順序	櫛の方向	櫛の本数（本）
1	右	8
2	左	7
3	上	２０

表１：櫛の方向と本数

入力となる小石模様は図５を用い，境界条件１を用いて実行した．

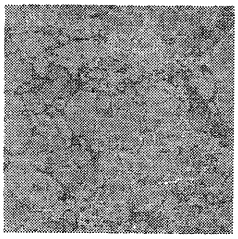

図５：初期画像

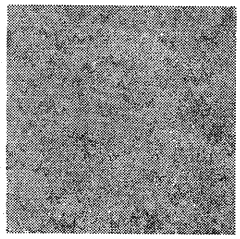

図６：櫛目模様

7.2. 渦巻き模様

図７は渦巻き模様を生成した結果である．初期画像は図５，境界条件は１である．

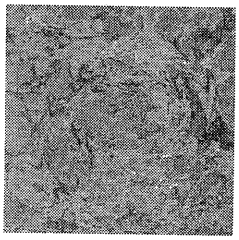

図７：渦巻き模様

7.3. 周期的境界条件を用いた結果

境界条件を周期的にすることによって，シームレスタイリングが可能となるマーブル模様を生成することができる．

図９は周期的境界条件を用いて生成した実行結果である．ただし，タイリング可能にするには初期画像も周期的である必要があるので，図８のような擬似的に生成した周期的な小石模様を用いる．

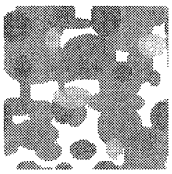

図８：周期的初期画像

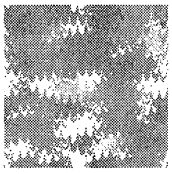

図９：周期的境界条件下での実行結果

実際に，図１０は図９の画像を上下左右２つずつ並べたものである．図１０から生成された画像は周期性が保たれていることがわかる．

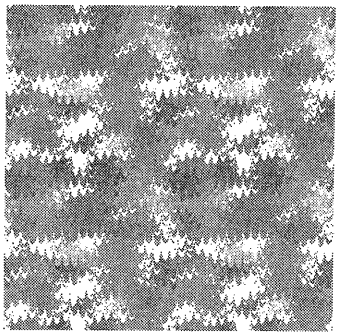

図１０；タイリングした実行結果

8. まとめ

本稿では流体運動を支配するナビエ・ストークス方程式を数値的にとくことにより，マーブリングテクスチャを生成する方法を提案した．また，この方法によって作成された速度ベクトル場に初期模様であるスキャナで取り込んだ小石模様を適用することによってマーブリングテクスチャを生成するアルゴリズムを設計し，実装した．

本手法では，物理シミュレーションをおこなっているため，動粘性や攪拌の仕方などを考慮した実際の模様に近い多様なマーブリングテクスチャの生成が可能である．

本稿の画像変形処理では，ピクセルを流線に沿って移動させるという方法を用いているが，顔料は流れ場に沿って移動すると同時に，面積を広げるように周りに拡散するという性質を持つ．よって，顔料の粘性や，周辺粒子との色の濃度の差を考慮した拡散方程式を用いて画像変形処理を行えば，より実際のマーブル模様に近い結果が得られると考えられる．

また，画像変形処理の改良のほかに，実行時間の短縮，小石模様のテクスチャ生成の実現も目指す予定である．

実行時間の短縮をおこなうにはStamの提案する手法が有用であると考えられる [6]．本稿では連立した方程式を単独で解いていたため２段階のステップを踏んでいた．一方，Stamの手法では連立方程式を速度についての単一の方程式とすることで，圧力$\boldsymbol{p}$を消去し，１ステップでナビエ・ストークス方程式を解くため，実行時間が大幅に短縮できる．

参考文献

[1]. 貴田　庄：母と子のマーブリング，美術出版社，1992

[2]. 藤田　昇：CGマーブリングの基礎的検討，山梨大学修士論文，1999

[3]. Nick Foster and Dimitri Metaxas：Realistic Animation of Liquids, Graphical Models and Image Processing, 58(5), pp471-483, 1996

[4]. 加藤　宏：ポイントを学ぶ　流れの力学，丸善株式会社，pp.37-48，1989

[5]. 河村　哲也：流体解析Ⅰ，朝倉書店，pp.49-65，1996

[6]. Jos Stam：Stable Fluid, Computer Graphic Proceeding, Annual Conference Series,1999, pp121-128, 1999