グラフィクスとＣＡＤ 35－6
（１９８８. 9. ２９）

テクスチャを素材にした アニメーションの制作

大島　登志一*　　板橋　秀一**

筑波大学　*工学研究科　**電子・情報工学系

写実的なＣＧ映像を制作する際、様々なテクスチャとマッピング技術を利用することによって、比較的低いコストで効果的に物体のディテイルを描写することができる。筆者らはテクスチャそのものが持つ表現力に興味を持ち、テクスチャを素材として、これを加工することにより、幾つかの対象の表現とそのアニメーションの制作を試みた。今のところ、そのためのデザイン作業およびコーディングは，全てデザイナーによる手作業である。そこで、この試行錯誤を頻繁に伴う作業の効率化を図るため、テクスチャに対する基本加工操作を分類し、その組み合わせをダイアグラムとして記述することを考えた。本論文では、その概要を報告する。

Trial Making of Texture Animation

Toshikazu OHSHIMA* and Shuichi ITAHASHI**

* Doctor's Program in Technology, Graduate School, University of Tsukuba
** Institute of Information Sciences and Electronics, University of Tsukuba

1-1-1 Tennodai, Tsukuba, Ibaraki, 305 Japan

In creating realistic imagery by means of computer graphics, it is possible to depict detail of objects effectively at low cost, by making use of various texture and mapping techniques . The authors have been interested in the potential of texture for representation, and tried to make animations of several objects making use of texture as material . In order to make designer's manual work including design and coding efficent, the authors systematized basic operations for material texture and devised a kind of diagram . This paper describes a trial version of the method .

1. はじめに

デザイナーの手をあまり煩わせずに写実的なＣＧ映像を制作するための手法・アルゴリズムについて、数多くの研究がなされている。筆者らは、主流であるそのような「３次元シミュレーション指向」から少し離れて、物理的な写実性にはこだわらない「２次元アート指向」のＣＧアニメーション・デザイン・ツールを開発中である。本論文では、その一環として行った試行について、その端緒を報告する。

コンピュータ・グラフィックスによって写実的な映像を制作する場合、比較的低いコストで物体のディテイルを表現することのできる効果的な手法として、テクスチャ・マッピング（ texture mapping; texturing ）［1-1 ］が頻繁に使用される。これは、反射・色・凹凸・透過等あらゆる応用が考えられ、基本的な技法として普及している［1-2 ］～［1-7 ］。様々な画像がテクスチャとして使用されるほか、アルゴリズムによるテクスチャの生成についても数多くの研究がなされている。

筆者らは、テクスチャそのものが持つ表現力に興味を持っている。しかし、従来のマッピングでは、テクスチャの扱いに関して以下のような傾向が一般的に強い。

① テクスチャだけでは対象そのものを表現しない。
② テクスチャを完成品と見做してそのまま使う。
③ 各テクスチャをそれぞれ単独で用いる。
④ テクスチャは静的なデータである。

筆者らは、これらに対して以下のような考え方に重点を置き、テクスチャを素材として、これを加工することにより、いくつかの対象の表現とそのアニメーションの制作を試みた。

① テクスチャが対象そのものを表現する。
② テクスチャを素材と見做して加工する。
③ 複数のテクスチャを合成して用いる。
④ テクスチャを動的なデータにする。

このようなテクスチャの加工を本論文では「テクスチャのアレンジ」と呼ぶことにする。テクスチャのアレンジによる対象の表現は、文献［1-8 ］に示唆されてはいるものの、ほとんど研究がなされていない。現時点では、テクスチャのアレンジに必要なデザイン作業およびコーディングは全てデザイナーによる手作業であるが、この試行錯誤を頻繁に伴う作業の効率化を図るため、テクスチャに対する基本加工操作を分類し、その組み合わせをダイアグラムとして記述することを試みた。以下、その概要を述べる。

2. 基本操作関数の分類

テクスチャの加工は、既存の手法によって得られるテクスチャをベースとして、関数を幾つか組み合わせてテクスチャの再合成をプログラムする形で行う。まず基本操作関数の分類について述べる。

ここで目的とする加工後のテクスチャの最終的な形式は、式(2.1) に示すような、２次元の単位空間における正規座標｛ｘ，ｙ｜０≦ｘ≦１，０≦ｙ≦１｝および任意の時間スケールでの時刻ｔを入力すると濃度値｛ｉ｜０≦ｉ≦１｝を返す時間関数である。

$$i = f(x,y,t) \quad 0 \leq x,y,i \leq 1 \quad \ldots(2.1)$$

このような関数を、以下に示すような入出力関係によって分類する４種類の型の基本操作関数を組み合わせて構成する。

① テクスチャ関数：濃度値の供給
　入力＝座標［，時刻］　／　出力＝濃度値
② 座標系変換関数：座標系の変換
　入力＝座標［，時刻］　／　出力＝座標
③ 濃度値混合関数：複数の濃度値の混合
　入力＝濃度値の組［，時刻］　／　出力＝濃度値
④ 濃度値変換関数：濃度値の変換
　入力＝濃度値［，時刻］　／　出力＝濃度値

ごく大まかに言えば、座標系変換関数によって流れや揺らぎ・動きを表現し、テクスチャ関数によって供給される濃度値を、濃度値変換関数・濃度値混合関数によって希望に沿うように加工する。

関数によっては、上に示した入力の他に、その作用を調整するための局所的な「制御変数」が与えられる。時刻は、座標や濃度値のような変換されるべき入力データとしてよりは、制御変数としての役割を果たすが、アニメーションにおいて重要な意味を持つ特殊な大局的パラメータであるため、別扱いとした。なお、時刻を入力に持つ関数には、「時間スケール係数」を制御変数として与える。時間スケール係数は時間経過による作用の度合いを制御するもので、これを０に設定することによって、時間の影響を無視することができる。以下、各型の基本操作関数について述べる。

2.1 テクスチャ関数

テクスチャ関数は濃度値の供給源であり、値域は［０，１］である。これは、純粋に数学的に定義したものでもテーブルを使用したものでも構わない。テーブルを参照する場合、パターンを繰り返し表示できるようにインデックスを循環させたり、指定した座標値に対応するイン

デックスが整数にならないときにはテーブルの値を補間するなどの工夫が必要である。

なお、主な用途によりテクスチャを大きく次の２種類に分類した。

ｄ（detail）型：　主に表現すべき対象のディテイルを表すための素材とする。

ｓ（shape ）型：　主に表現すべき対象の大まかな形状を表すために用いる。

２．２　座標系変換関数

形状の変形や流れ・揺らぎ・動きなどは全て座標系の変換によって表現する。座標系を変換すると、その効果は視覚的には逆変換として認識される。例えば、直交座標系で座標値を右方向に移動させると、表示されるテクスチャは左へと移動する。拡大縮小・回転なども同様である。

２．３　濃度値混合関数

ここでいう濃度値の「混合」とは単なる加算的なものでなく非常に広い意味を持つ。例えば、二つの入力濃度値があるとき、一つの濃度値は他方を修飾するためのパラメータとなる場合が多い。典型的には、前者はｓ型テクスチャであり、後者はｄ型テクスチャである。例えば、あるｄ型テクスチャに対して座標値に依存した閾値処理を行いたい場合、その閾値としてｓ型テクスチャを用意し、それらの値を「混合」する。

濃度値混合関数の入出力は濃度値であるので、この定義域・値域はともに［０，１］である。

２．４　濃度値変換関数

濃度値変換関数は、濃度値の非線形補正を行う。定義域［０，１］・値域［０，１］のスカラー関数であれば、希望するような入出力関係を持ったどんな関数でも濃度値変換関数として使うことができる。

濃度値混合関数と濃度値変換関数との間には、ある程度の互換性がある。つまり、濃度値混合関数の入力の一つを制御変数として扱うことによって、これを濃度値変換関数として使用できる。また、濃度値変換関数のある制御変数が［０，１］の範囲で有効なものであれば、これをもう一つの濃度値入力として扱うことによって、濃度値混合関数として使用することもできる。

３．　ダイアグラムとコーディング

デザイン作業には試行錯誤を繰り返しながらのコーディングを伴うので、これを効率的に行うために、全体の処理を把握する助けとなるダイアグラムがあると非常に便利である。今回採用したダイアグラムは、基本的には、関数を表すノードをデータの流れを表す矢印で接続した有向グラフである。座標・濃度値・時刻の３種類のデータの流れを表す矢印を図3.1 に示す。また、各型の関数を表すノードを図3.2 に示す。ダイアグラム中、ノードには関数名を、矢印には変数の識別名を添えることによって、プログラムとの対応をとる。なお、時刻の識別名は常にｔとした。図3.3 に記述例を示す。これは、２つの濃度値入力を持つ識別名BLEND の濃度値混合関数を表す。濃度値混合関数は複数の濃度値入力を持つので、それらを識別するために、その端子としてノードを囲む同心円を１つずつ対応させることにした。図3.3 に対応するプログラムは、例えば次のようになる。

```
i3 = BLEND( i1, i2, t ) ;
```

ダイアグラムの各部分では、その入出力関係によって、

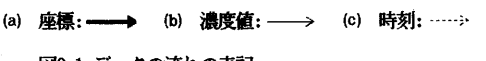


図3.1 データの流れの表記

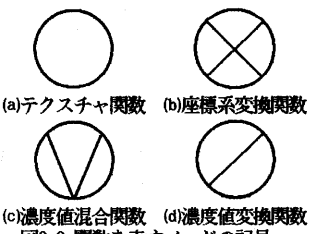


図3.2 関数を表すノードの記号

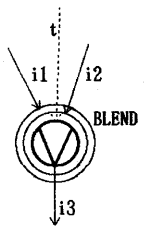


図3.3 記述例

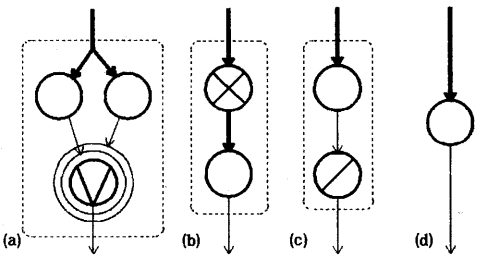


図3.4 ノードの置き換え①

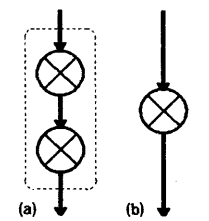


図3.5 ノードの置き換え②

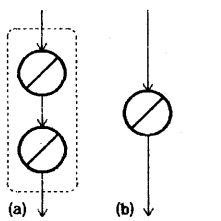


図3.6 ノードの置き換え③

複数のノードを単一のノードにまとめて置き換えることができる。図3.4 (a)～(c)はテクスチャ関数のノード(d)と置き換えることができる。また、図3.5 (a)は座標系変換関数のノード(b)と、図3.6 (a)は濃度値変換関数のノード(b)とそれぞれ置き換えることができる。このようなダイアグラムの構造化によって、基本的な関数から複雑な関数のライブラリを容易に構築することができ、デザイン作業をより系統的に進めることができると期待される。

4. 中点変位法による基本テクスチャの生成

実験に使用した主なテクスチャは、一般に「中点変位法」として知られるアルゴリズム（recursive subdivision algorithm）［4-1］によって生成した。中点変位法は、フラクタル曲線・曲面を生成する効率的なアルゴリズムである。フラクタル［4-2］は、幅広い範囲のスケールでの統計的自己相似を特徴とする、複雑な形状のクラスである。多くの自然現象あるいはこれに起因する天然の存在（乱流、稲妻、海岸線、山岳、樹木、雲、河川網など）がフラクタルとして扱えることが確認されている［4-2］～［4-4］。そのため、この特徴を持つテクスチャは、多くの対象の表現に適用できると期待され、今回の表現の素材」としての用途に適する。

本インプリメントとオリジナル［4-1］との主な相違は次の２点である。① 計算の際、配列の添字を、循環しているものとして扱った。② 中点変位の実行を開始するレベルを指定するようにした。

4.1 配列の循環

テクスチャは２次元配列に格納する。オリジナルでは、最大分割レベルをｎとすると配列のサイズは（2^n+1）×（2^n+1）である。そして、そのアプリケーション（地形のモデリング）の都合から、図4.1 のように、配列の最上下行・最左右列上の各要素は、内部の要素とは独立にその列・行上の要素のみを参照して求めている。しかし、これでは全体の特性が均一にはならない。また、今回の用途の必要上、この同一のテクスチャを単位とした繰り返しを表示すると、隣接する境界で特徴の不連続が生じる。これらを解決するため、計算の際に配列の行と列の各添字を循環させて扱った。

まず、例外処理を行っている最上下行・最左右列上の各要素を内部の要素と共に、正規の処理により求めるためには、図4.2 のように、実際には存在しない要素が必要になる。そこで、このような実在しない要素を、添字の最大値2^nを最小値０にオーバーラップさせ、添字を循環させて扱うことによって、同一の配列から参照する。これは、直観的には、図4.3 のように、円環の表面を覆おう端のない配列上でテクスチャを生成することを意味する。最上行と最下行、最左列と最右列は、それぞれ重複するので、これらのうち添字０の最上行と最左列のみを求めれば良い。したがって、最下行、最右列を除いた$2^n \times 2^n$の要素を求めるために参照される添字の範囲は［-2^{n-1}，2^n］となるが、計算過程で配列の実際の添字範囲［０，2^n-1］を逸脱して参照される添字を式(4.1) によって置き換える。

$$i' = \begin{cases} i, & 0 \leqq i \leqq 2^n - 1 \\ 0, & i = 2^n \\ i + 2^n, & -2^{n-1} \leqq i \leqq -1 \end{cases} \quad \ldots(4.1)$$

n： 最大分割レベル
i'： 新しい添字
i： 元の添字

図4.4 にこのようすを示す。これにより、繰り返し表示

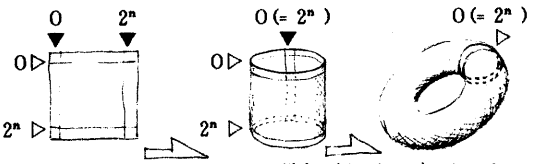
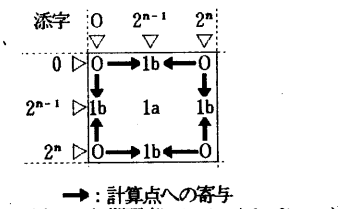
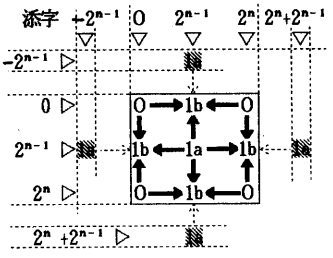


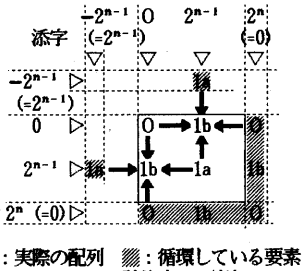
図4.3 配列の行・列の循環



→：計算点への寄与
計算順序:0(初期要素),1a,1b (,2a,2b,...)
図4.1 オリジナルにおける例外処理



□：実際の配列　▒：存在しない要素
→：計算点への寄与　⇢：満たされない寄与
図4.2 存在しない要素の参照



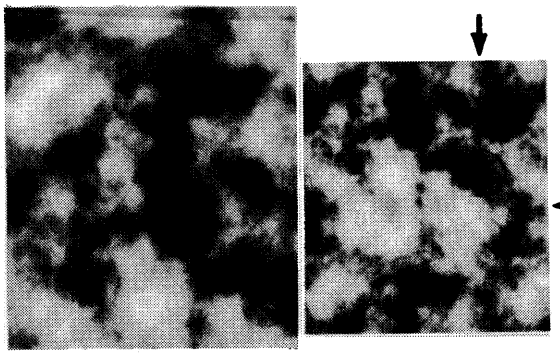
□：実際の配列　▒：循環している要素
→：計算点への寄与
図4.4 配列の循環

の連続性が実現でき、全ての点が統一的な処理により計算できるほか、必要な配列のサイズが最大分割レベルｎで$2^n \times 2^n$となり扱いやすい。なお、中点変位法による各点の計算を完了した後、後処理として、その値を［０，１］に正規化する。テクスチャの例（部分）と、これを４つ並べて表示した接合部分付近を図4.5 に示す。

4.2　中点変位開始レベル

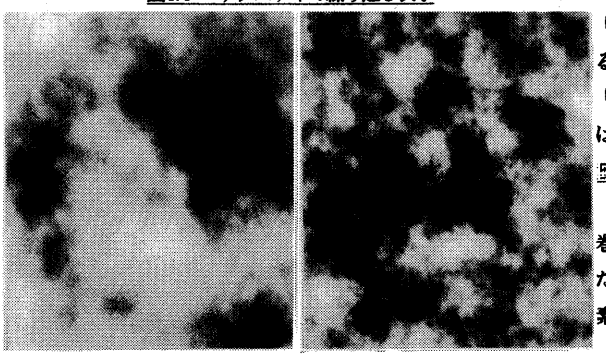
レベル１からレベルｎまで一貫して中点変位を行うのが一般的であるが、今回は変位を開始するレベルを指定するようにした。

各分割レベルで中点変位を行うと、そのレベルに応じた広さの凹凸が加えられていく。このうち特に、レベル１で生じる配列全域に及ぶ広い凹凸は、テクスチャを繰り返し表示する際、周期性を目立たせる原因になり、問題となった。そこで、変位開始レベルを指定することにし、これ以前のレベルでは、変位させずに同一の初期値を与える。これによって、それ以前のレベルに起因する幅の広い凹凸を除去し、それ以降の細かい特徴だけを利用することができる。開始レベル１と４の実例（部分）を図4.6 に示す。分割レベルｎは８であり、画像サイズは 256×256 画素（全体）、計算時間は25秒程度である。



(a)テクスチャの例（部分）　(b)接合部分付近（部分）

図4.5　テクスチャの繰り返し表示



(a)開始レベル１（部分）　(b)開始レベル４（部分）

図4.6　変位開始レベルの効果

５．　制作例

ここでは、テクスチャを素材としたアニメーションの制作例として「炎」［5-1 ］［5-2 ］・「星雲」・「台風」の３つを取り上げ、そのダイアグラムと出力映像を示す。これらのダイアグラムは、試行錯誤の末に得られたものである。現在、研究の初期段階で、ダイアグラムの記述についてなど未完成の事項が多いので、ダイアグラム各部分の詳細についてなどの説明は割愛した。

５．１　「炎」

「炎」のためのダイアグラムを図5.1 に示す。ダイアグラム前半の７つの座標系変換関数で、炎の「流れ」と「揺らぎ」を表現する。Ⅰ・Ⅱ・Ⅲの各段階における座標系変換の効果を図5.3 ～5.6 に示す。そして、微細構造（Ｄ）と大局形状（Ｃ）とを合成して炎の単色映像（Ｅ）を得る。Ａ～Ｅにおける各映像を図5.7 ～5.11に示す。さらに、ダイアグラムの最終段階で、３つの濃度値変換関数によって単色映像から色彩映像の赤・緑・青成分を得る。身近に見られる焚き火やろうそくなどの黄色い炎は、熱せられた炎中の煤が黒体放射によって輝いて見えるものである。ここでは黒体放射にこだわらず赤から黄白色へと感覚的に配色したが、比較的良い結果を得ている。「炎」から４フレームを図5.21～5.24に示す。画素数は 128×128 、計算時間は１フレーム約20秒である。

５．２　「星雲」

「星雲」のためのダイアグラムを図5.2 に示す。座標系の変換によって渦をつくり、時間の経過とともに星雲を回転させた。Ⅰ・Ⅱ・Ⅲの各段階における座標系変換の効果を図5.12～5.14に示す。また、微細構造（Ｂ）と腕（Ａ）から大まかな映像（Ｃ）をつくり、徐々に星雲を形成していく。Ａ～Ｆにおける各映像を図5.15～5.20に示す。濃度値変換関数"star"は、入力された濃度値を「確率」として扱い、乱数で「星」を発生する。星の明るさは、星の発生と同時に別の乱数系列で計算される。「星雲」から２フレームを図5.25,5.26 に示す。画素数は 256×256 、計算時間は１フレーム約４分である。

５．３　「台風」

座標系の変換によって、時間の経過とともに雲が渦を巻きながら中心部から周辺部へと吹き出す様子を表現した。「台風」から２フレームを図5.27,5.28 に示す。画素数は 256×256 、計算時間は１フレーム約３分である。

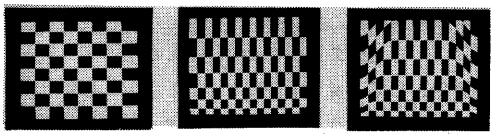


図5.3 テストパタン　図5.4 炎：Ⅰ　図5.5 炎：Ⅱ

図5.12 星雲：Ⅰ　図5.13 星雲：Ⅱ　図5.14 星雲：Ⅲ

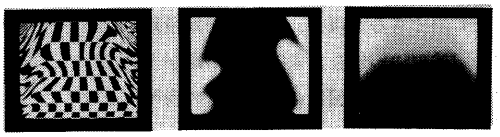


図5.6 炎：Ⅲ　図5.7 炎：A　図5.8 炎：B

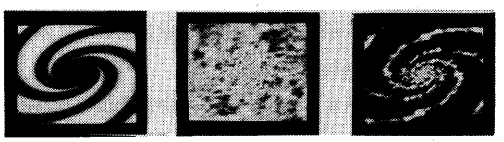


図5.15 星雲：A　図5.16 星雲：B　図5.17 星雲：C

図5.9 炎：C　図5.10 炎：D　図5.11 炎：E

図5.18 星雲：D　図5.19 星雲：E　図5.20 星雲：F

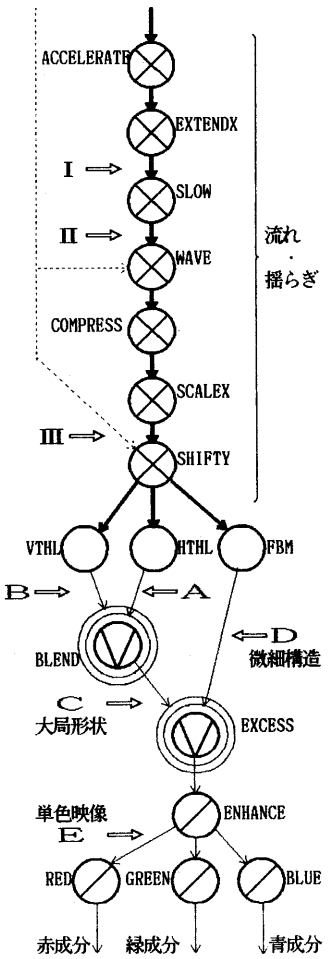


図5.1「炎」のためのダイアグラム

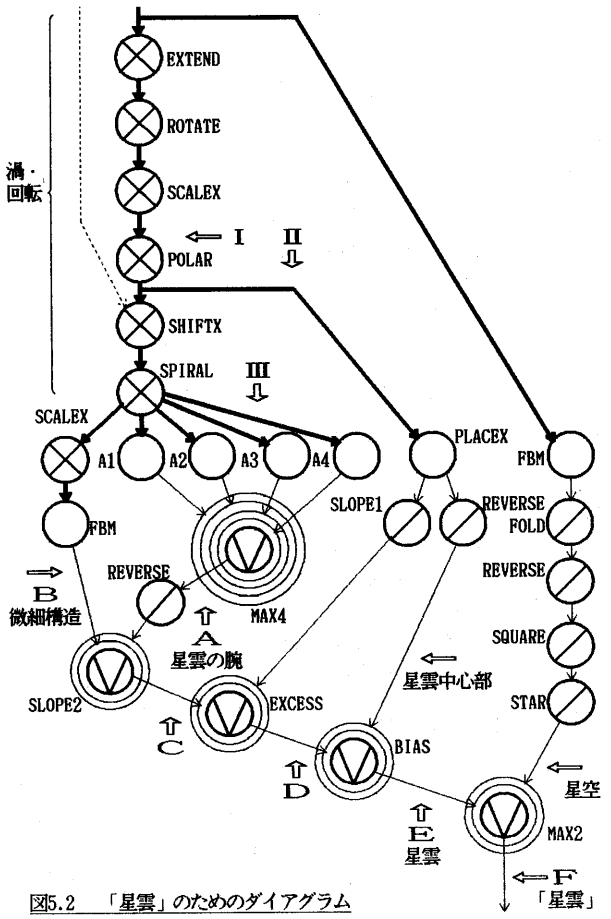


図5.2 「星雲」のためのダイアグラム

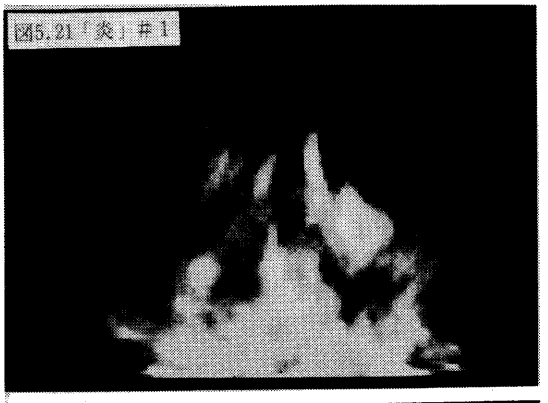


図5.21「炎」＃1

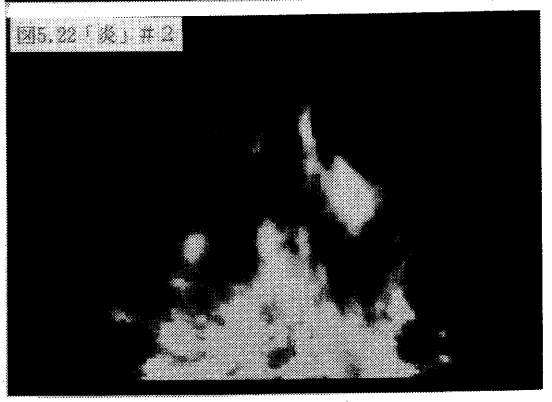


図5.22「炎」＃2

図5.23「炎」＃3

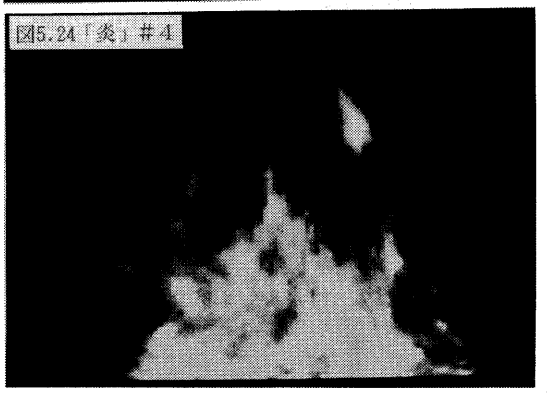


図5.24「炎」＃4

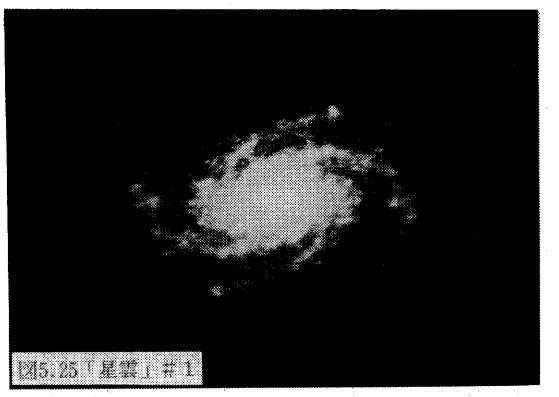


図5.25「星雲」＃1

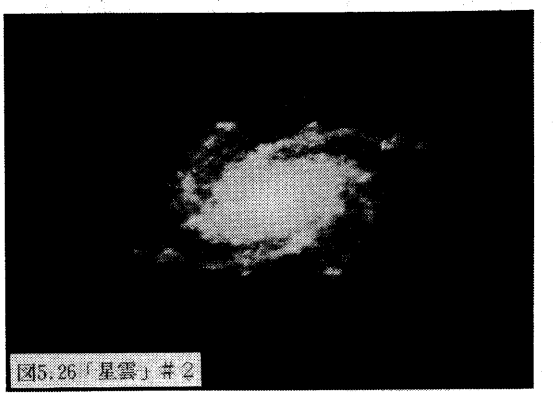


図5.26「星雲」＃2

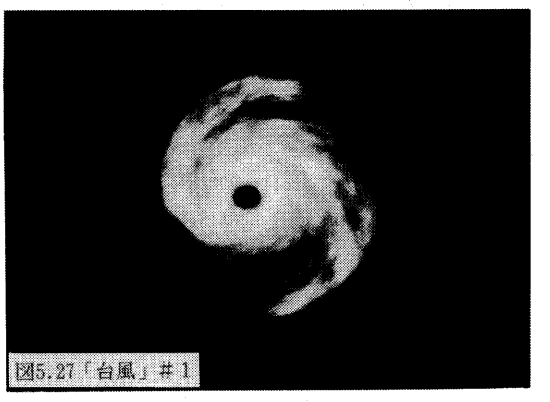


図5.27「台風」＃1

図5.28「台風」＃2

6. むすび

テクスチャを素材とした様々な対象の表現とそのアニメーションの制作を行うために、既存の手法によって得られるテクスチャをベースとして、簡単な関数の組み合わせによるテクスチャ関数の合成を行った。実際に幾つかのアニメーションを制作したが、構造化ダイアグラムの採用で、テクスチャのアレンジ手順を簡潔に記述することができ、デザインおよびコーディングを効率的に進める上でたいへん役に立つことがわかった。また、アニメーションの試作を通じて、従来の一般的なモデリング手法では難しかった雲・炎・煙など形状のあいまいな対象の「流れ」の表現が、テクスチャを用いることによっても低いコストで実現できる可能性が確認された。さらに、中点変位法によるテクスチャは、期待した通り柔軟性に富み、様々な対象の表現に使えることが確認された。

今後の課題としては、

① 関数ライブラリの充実

② 基本的なアレンジ・ルールの整理

があげられる。最終的な目標としては、ディスプレイ上でダイアグラムを対話的に編集するだけで映像を制作できるようなシステムに発展させて、テクスチャを用いた新しいタイプのアニメーション・デザイン・ツールとして完成させたい。

本研究における環境を以下に示す。

① 画像生成：

ＣＰＵ	Apollo DOMAIN SERIES 4000
使用言語	Ｃ（DOMAIN C）

（Apollo, DOMAIN: Apollo Computer 社登録商標）

② 表示：

ＣＰＵ	NEC PC9801VX
ディスプレイ	NEC PC-453n
フレーム・バッファ	SIG 501-FB

謝辞

出力映像の批評など研究過程で貴重な御意見を頂いた研究室の諸氏に感謝します。

参考文献

[1-1] Catmull,E. "A subdivision algorithm for computer display of curved surfaces," Ph.D. dissertation,University of Utah,1974.

[1-2] Blinn,J.F.and Newell,M.E."Texture and reflection in computer generated images,"CACM,19, 10 (Oct.1976),542-547.

[1-3] Blinn,J.F."Models of light reflection for computer synthesized pictures,"Computer Graphics, 11,2(1977),192-198.

[1-4] Blinn,J.F."Simulation of wrinkled surfaces "Computer Graphics,12,3(Aug.1978),286-292.

[1-5] Gardner,G.Y."Simulation of natural scenes using textured quadric surfaces,"Computer Graphics 18,3(July 1984),11-20.

[1-6] Cook,R.L."Shade trees,"Computer Graphics, 18,3(July 1984),223-231.

[1-7] Crow,F.C."Summed-area tables for texture mapping,"Computer Graphics,18,3(July 1984),207-212.

[1-8] Perlin,k."An Image Synthesizer,"Computer Graphics,19,3(July 1985),287-296.

[4-1] Fournier,A.,Fussell,D.,and Carpenter,L.C." Computer rendering of stochastic models,"CACM,25, 6(June 1982),371-384.

[4-2] Mandelbrot,B.B."The Fractal Geometry of Nature,"W.H.Freeman&Co.(1982).

[4-3] 高安秀樹「フラクタル」、朝倉書店、1986.

[4-4] 別冊数理科学「形・フラクタル」、サイエンス社、1986.

[4-5] 金子、「画像特徴としてのフラクタル次元」、信学技報、PRL85-30,pp.31-39,(1985).

[5-1] 大島、板橋、「２次元テクスチャを用いた炎の簡易表示の試み」、情報処理学会研究報告、グラフィクスとＣＡＤ 30-5,(Nov. 1987).

[5-2] 大島、板橋、「２次元テクスチャを利用した炎の簡易表示の試み」、NICOGRAPH'87論文集、pp.175-181 (Nov. 1987).