

データフロー型モジュール指向ソリッドテクスチャジェネレータ

5V-6

永廣ますみ 奥嶋明希 川田亜矢子 市川哲彦 佐藤浩史 藤代一成

お茶の水女子大学 理学部

1 はじめに

CG制作において、物体の材質感を与えるテクスチャは重要な役割を果たす。従来、関数型テクスチャジェネレータでは、テクスチャ記述言語を用いるためにパラメータや関数の変更・再実行に時間がかかるだけでなく、テクスチャ生成プロセスの把握も容易ではなかった。そこで、本論文ではこれらの問題点を解決するために、K. Perlinが提案した Noise 関数によるテクスチャ生成法 [1] に示されている基本関数とその加工機能をモジュール化し、データフローの概念をとり入れた対話型ソリッドテクスチャジェネレータを開発する。実装にあたり、試験的に AVS (Application Visualization System) Ver.4 を使用した。

本稿は、まず次節で Perlin のテクスチャ生成法と AVS を対比させ、3 節では本ジェネレータの特徴について述べる。4 節では Perlin の primitive 関数の基本モジュール化を説明し、5 節では本ジェネレータでのテクスチャ生成方法を述べ、6 節では生成したテクスチャの例を示す。そして最後に、まとめと今後の展望を述べる。

2 Perlin のテクスチャ生成法と AVS

K. Perlin が 85 年に提案した Noise 関数によるテクスチャ生成法 [1] では、一様乱数を返す Noise() を含む primitive 関数や turbulence() 等の多くの composite 関数を用意し、それらの関数を選択・合成することにより、様々なテクスチャを作り出すことができる。

一方 AVS は、データフローパラダイムとモジュール指向という 2 つの特徴を有している。すべてのプログラムはモジュールをノードとする有向グラフ（ネットワーク）で表わされ、モジュール間を結ぶアーク上をデータが加工されながら流れていく様子が可視化される（デー

タフローパラダイム）。AVS のモジュールライブラリは、利用頻度の高い組込みモジュールや、ユーザが独自に作成・登録したモジュールから構成されている。ユーザは、ここから自由にモジュールを再利用してネットワークを作成する。さらに、その作成したネットワークをマクロモジュールとして登録しておくことも可能である（モジュール指向）。

以上の説明から、Perlin のテクスチャ生成法における primitive 関数を AVS 上の基本モジュールに、composite 関数をマクロモジュールにそれぞれ対応させることができ、データフローとモジュール指向という概念によって、関数型ソリッドテクスチャ生成法の能力を大きく引き出すことができると考えられる。ただし、AVS のデータフロー機能の限界により、ネットワークとして作成できない composite 関数もあるため、ここではそのような関数も primitive 関数として扱い、基本モジュールとして作成することにした。

3 本ジェネレータの特徴

本ジェネレータの特徴は次の 2 点にまとめられる。

1. テクスチャ生成プログラムの編集をビジュアルな環境で行なうことにより、生成プロセスの把握が容易になり、プログラミング時間の短縮化が期待できる。
2. 本ジェネレータは、ユーザが要求に応じて選択・結合したモジュールネットワークを、逐次的に評価することによりテクスチャを生成する。1つ1つのモジュールやパラメータの変更は、それ以降のネットワークを再評価することにより直ちに結果出力に波及させられる。そのため、実行のターンアラウンド時間を短縮することができる。

4 primitive 関数の基本モジュール化

モジュールにはパラメータがあり、それによりテクスチャの拡大・縮小や凹凸などを制御することができる。ここでは、後掲の図に示されているテクスチャを生成する際に用いた、2つの基本モジュールについて説明する。

- `Noise()`

三次元ベクトル（座標値）をとりランダムなスカラ値を返す関数。

↓

`noise` モジュール

離散化された空間の各点に対して `Noise()` を施す。スケールと X 方向、Y 方向、Z 方向の長さの4つのパラメータがあり、発生させた乱数をスケールにより拡大・縮小している（格子間の値の補間には trilinear 補間 [2] が用いられている）。

- `turbulence()`

三次元ベクトルをとり `Noise()` を用いてゆらぎを生成する関数。例えば、縞模様のテクスチャにこの関数を用いることにより、大理石のテクスチャが作り出せる。

↓

`turbulence` モジュール

離散化された空間の各点に対して `turbulence()` を施す。noise モジュールの出力値を入力とし、パラメータのスケール値を変えることによりゆらぎ具合の調節が可能となっている。

5 本ジェネレータでのCG作成方法

物体の形状モデリングには、メタボール的手法 [3] を用いる。形状は、物体の存在しない領域の値を0、存在する領域の値を非0として内部に向かうほど値の増加する3次元のボリュームデータとして与える。そのため、基本物体のボリュームデータの拡張集合演算を行なうことにより、複雑な形状モデリングを行なうことができる。一方、前節で述べたテクスチャ作成モジュールの出力も、スカラ値かベクトル値の3次元のボリュームデータになっている。

現バージョンでは、これら2つのボリュームデータを入力とする isosurface モジュール（AVS 組み込みモジュール）を使い、等値面上のテクスチャを抽出することによって、ソリッドテクスチャリングを実現させている。

6 例

下図は、本ジェネレータで生成した大理石テクスチャのネットワークとその出力結果である。画面左には、Noise モジュールのパラメータを調節するための4つのダイヤルが表示されている。isosurface モジュール入力までのテクスチャ生成ネットワークをマクロモジュールとして登録することにより、大理石テクスチャ生成モジュールとしての再利用が可能となる。

7 まとめと今後の展望

データフローとモジュール指向の概念に基づいて関数型テクスチャジェネレータを構築したことにより、効率的かつ効果的なテクスチャ生成プログラムの編集環境を実現した。今後の展望としては次の3点が挙げられる。

- モジュールライブラリの充実
- より良い形状モデリングの追求
- 他のテクスチャ生成法の研究 [4]

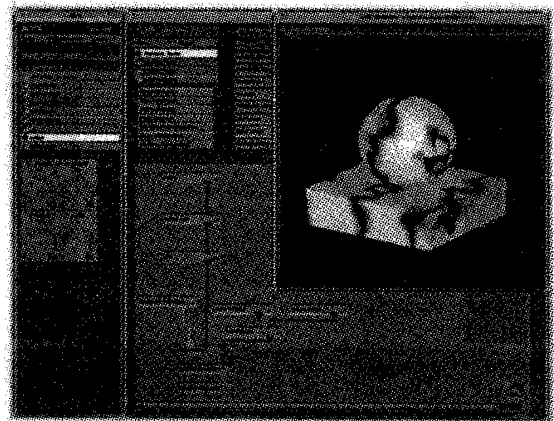


図 本ジェネレータの編集環境とテクスチャ生成例

参考文献

- [1] Perlin, K.: "An Image Synthesizer," in *Proc. SIG-GRAPH'85*, pp. 287-296 (1985).
- [2] 藤代一成、茅暁陽：“ボリューム・ビジュアライゼーションの基本アルゴリズム”，*PIXEL*, No.119, pp. 127-132 (1992)
- [3] 高沖英二：“みんなのメタボール”，*PIXEL*, No.88, pp. 106-109 (1990)
- [4] 宮田一乗：“骨格線分を用いたテクスチャ生成法”，情報処理学会 グラフィックスとCAD研究会, 65-8 (1993)