

複合属性型を用いたレンダリング法

4Q-7

青野 雅樹

日本アイ・ビー・エム株式会社 サイエンス・インスティテュート

1. はじめに

3次元オブジェクトのリアルな表現法のひとつとして、複合属性型を用いたレンダリング法と、これのCSGモデル[文献1]への適用例について述べる。

従来のレンダリングでは、CSGモデルのプリミティブと属性の関係は1対1であることが多かった。複数の属性を持つ物体を表現するためには、異なる属性をもつプリミティブを複数個用意してそれらに和・差・積に代表される集合演算を繰返して対応していた。しかし、集合演算の反復使用は、冗長なプリミティブの定義を増大させ、しかもレンダリングの効率低下を招く。

そこで、プリミティブと属性の関係を1対多で与える複合属性型を導入し、複数の属性を持つ物体をより少ない集合演算の組合せで定義し、効率的にレンダリングできる手法を報告する。

2. 複合属性型プリミティブとその要件

CSGモデルにおいて、プリミティブ対属性の関係を1対1で与える従来の方法により定義されるプリミティブを単属性型のプリミティブと呼ぶことにする。これに対して、プリミティブ対属性の関係を1対多で与えるものを複合属性型のプリミティブと呼ぶことにする。単属性型のプリミティブでは、プリミティブの形状と属性の関係が固定されてしまう。属性の観点から見ると、立方体、円柱、あるいは球といったプリミティブ単位でしか定義されないため、属性が立体の表面上に複雑に入り組むような物体例えば自然界では花崗岩の表面とか、砂場の表面など、また人造物ではアルミ箔のシールを貼った容器などは、複数の属性が存在し、これらを単属性型のプリミティブで定義することは困難である。また、複数の単属性型のプリミティブの集合演算の組合せで表現できる場合もあるが、この際プリミティブの数が増大し、レンダリングの効率が低下する。

これに対して、複合属性型のプリミティブを導入すれば、単属性型のプリミティブの組合せで表現するよりも効率は向上し、しかもそれだけでは表現できなかったリアリズムを増すことができる。

3. 属性マッピング

リアルな画像を生成する手段として Catmull [文献2]によってテクスチャマッピングが導入されたが、このテクスチャマッピングの概念の一般化として属性マッピングを定義することができる[文献3]。これは、パラメトリックに表現された3次元物体の表面を単位正方形 $[0,1] \times [0,1]$ のウィンドウにマッピングして正規化変換を行ない、逆にこのウィンドウで定義された属性領域を、逆変換して3次元物体の表面の領域にマッピングする手法として定義できる。ここで属性とは、面の色やテクスチャだけでなく、面の光学的性質や凹凸具合など、面を特徴づける性質を表す。

正方形のウィンドウで定義される属性は、それが有効な固有の領域内で定義されるものとし、その領域は以下のような面領域を分割する構造(以下PDS = Planar Division Structureと呼ぶ)で表現されるものとする。

(1) CPG (Constructive Planar Geometry)

CSG (Constructive Solid Geometry) の2次元版とも言えるもので、平面上で定義される幾つかのプリミティブ、例えば長方形、楕円、多角形などに対して、和・差・積の集合演算および移動・回転などのアフィン変換演算の組合せとして、図形を定義しようとするものである。属性は、各プリミティブごとに与えられ、その領域は任意形状のものが任意個、概念的に生成される。

(2) Pixel Array

離散的な最小の単位 pixelの集合として、図形を定義する。属性は各 pixelごとに割り当てられる。領域の数は、pixelの数に支配される。

(3) Cell Decomposition

tessellation functionを利用したtilingや, mosaicに代表される表現法と、4分木[文献4]に代表される階層的な再帰的表現法がある。属性は、各 cellごとに割りあてられ、その数はcellの数に依存する。

(4) その他

閉曲線で囲まれた閉領域を表す手法としてBallard [文献5]によるstrip treeやB-spline曲線などのパラメトリックな表現も定義できる。属性は各閉領域ごとに割りあてられ、その閉領域の数に依存する。

A Rendering Method using Multi-attribute Primitives

Masaki Aono

Science Institute, IBM Japan Ltd.

どのPDSが適切であるかは、応用に依存することが多く一概には決められない。例えば、効率が問題となる状況では、pixel arrayを用いると、任意の与えられた点がどの領域に属するかの判定(point membership problem)は $O(1)$ で解決される。一方、記憶領域の節約が問題となる状況では、他のPDSの方が有利である。

本報告では、円や矩形領域の定義しやすさから、CPGを用いて面の領域分割を行った。計算時間の観点からみると、N個のプリミティブからなるバランス木構造のCPGでは、point membershipの判定は、 $O(\log N)$ 回必要となる。一方、属性の領域分割の観点からみると、次の問題が生じる。すなわち、2つの属性の異なるプリミティブが交錯する場合、交錯する部分の属性をどう取り決めるべきか、という問題である。ここでは、簡便のため、演算子に対して作用される被演算子の現れる順序に優先度を与えるように取り決めた。こうすることによって、あいまい性をなしに、2次元平面上での属性に基づく領域分割が可能になり、この属性によって識別される領域を属性マッピングで3次元物体の表面に貼りつけることができる。

4. 属性マッピングの実現例

レイ・トレーシングは、屈折、反射に代表される光学属性を表現するのに優れた手法である。属性マッピングは、テクスチャー属性、光学属性、凹凸属性などの属性を柔軟に表現できることが前提条件であり、この実現にあたって、レイ・トレーシングを利用することは有意である。

レイ・トレーシングに属性マッピングを組み込んだ手法をアルゴリズム1に示す。図1は、数個の属性からなる円柱を、このアルゴリズムによってレンダリングしたものである。円柱の上面部は3つの円形状の異なる属性領域から成り、側面部は"CG"というロゴの鏡面反射する矩形領域と、凹凸属性を持つ帯状の領域と、その他のタイル状のパターン領域が和集合演算で定義されてきている。

5. おわりに

本報告では、CSGモデルに基づくレイ・トレーシング法において、物体の詳細な属性を表現する方法として、プリミティブ対属性の関係を1対多で与える複合属性型のプリミティブを用いた属性マッピングの手法を示した。本手法を組み込んだ3次元物体のモデリングとレンダリングを行うシステムは、ホスト計算機IBM3081を用い、IBM5080グラフィック装置を使用して、VM/CMSの下で実現されている。

参考文献

- [1] A.A.G.Requicha and H.B.Voelcker, "Constructive Solid Geometry," Tech.Memo 25, Production Automation Project, 1977
- [2] E.A.Catmull, "A Subdivision Algorithm for Computer Display of Curved Surfaces," UTEC-CSc-74-133, 1974

- [3] M.Aono and T.L.Kunii, "Attribute Mapping," to be submitted to CG International '87
- [4] Hanan Samet, "The Quadtree and Related Hierarchical Data Structures," Computing Surveys, Vol.16, No.2, June, 1984
- [5] Dana H. Ballard, "Strip Trees: A hierarchical Representation for Curves," CACM, Vol.24, No.5, 1981

アルゴリズム 1 (属性マッピング)

- ステップ1: レイと物体の交点を求める。もし、何とも交わらなければ、背景色をセットして終り。そうでなければ、視点から最も近い物体との交点(x,y,z)と、その物体のプリミティブのid (PID)をセットする。
- ステップ2: 交点座標(x,y,z)を2変数パラメータ(u,v)に正規化変換する。
- ステップ3: PID に対して定義される各面につき、それぞれのPDSを探索し、どの面領域に、その(u,v)が含まれるかを調べる。求まった面領域に対して定義される属性値をセットする。
- ステップ4: 交点(x,y,z) が影に入るかどうか計算し、明度値をセットする。
- ステップ5:
- (1) 凹凸属性に基づき、交点上での法線ベクトルを求める。
 - (2) テクスチャー属性に基づき、(R,G,B) 値をセットする。
 - (3) 光学属性に基づき、シェーディングモデルを選択し、鏡面反射又は透過モデルの場合反射光、透過光に対し、ステップ1より更に再帰的に処理する。そうでなければ、得られた(R,G,B) 値をもって終了。

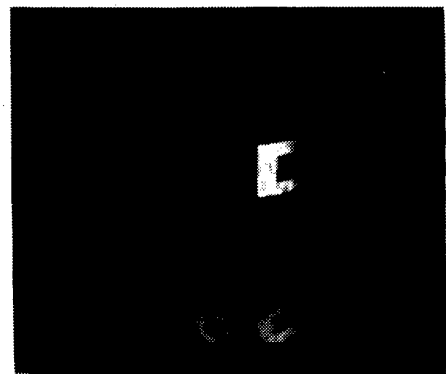


図1. 属性マッピングの適用例