

一般化シリンダを用いた枝分れ物体のモデリングとマッピングの手法

柿本 正憲* 林 伸彦** 大口 孝之** 山藤 真二*** ネルソン・マックス****
*富士通研究所 **富士通 ***山陰中央テレビ放送 ****光キネマ

ねじれない一般化シリンダのモデリング手法と、一般化シリンダの枝分れ形状を補間によって生成する手法と、マッピングを施す手法について述べる。一般化シリンダを多面体近似によって表現する場合、中心軸となる折れ線の経路によっては途中に不自然なねじれが生じることがある。これを、折れ線の各線分ごとにねじれ補正を行なう座標変換を施すことによって回避した。また、そのようにして作った一般化シリンダを複数組み合わせると枝分れ部分となる曲面パッチを新たに生成した。このような枝分れ物体は、四角形の画像をそのままマッピングしてもどこかで画像の不連続が起こる。この問題を、三角形のテクスチャ画像を用い、パッチの接合部分を合わせるように原画像を描くペイントシステムによって解決した。

Methods for modeling and mapping branched surfaces using Generalized Cylinders

Masanori Kakimoto* Nobuhiko Hayashi** Takayuki Oguchi** Shinji Santoh*** Nelson L. Max****
*Fujitsu Laboratories **Fujitsu ***San-in Chuo Television Broadcasting ****Hikari Cinema
*1015 Kamikodanaka, Nakahara-ku, Kawasaki 211, Japan

This paper describes methods for modeling generalized cylinders, generating branch surfaces, and mapping texture on surface of the shape. When modeling generalized cylinders with polygonal surfaces, a certain kind of path for the center line causes unexpected rotation between adjacent contours. We resolved this problem by giving a corrective rotation transformation to each of the contours. We gave a method to generate new patches which fill in branching points among three generalized cylinders arranged in Y-shape. Texture mapping is required in order to give reality to these shapes. Rectangular texture image will cause unavoidable discontinuity on the branch surface at the points where boundary of the unit texture is mapped. We resolved the problem by using two special triangular textures in addition to rectangular textures. We also developed a tool to paint these triangular textures with neighboring textures displayed at the right position on the display.

1. はじめに

我々は、樹木・血管・植物の道管師管などの不規則で枝分れした物体をCGで表現する研究を行っている。管状の三次元物体を表現する強力な手法として一般化シリンダ [1] [3] がある。一般化シリンダは、二次元閉曲線を、三次元空間内にある曲線に沿って移動させたときにできる軌跡として表現できる。この際、閉曲線の形状は、移動に伴って連続的に変化してもよい。通常のシリンダ (円柱) が、円を直線に沿って平行移動させた軌跡の形状であるのに対して、円を閉曲線に一般化し、直線を空間曲線に一般化したものであることからこの名称がある。

我々は一般化シリンダを用いて会話的に種々の形状をモデリングできるシステムを開発した。また、現実感を与えるために、ペイントプログラムによって作ったテクスチャをマッピングするシステムを開発した。全体の処理の流れを図1に示す。

各処理ステップにはそれぞれに解決すべき問題点がある。一般化シリンダのモデリングの段階ではねじれによって形状が歪むという問題がある。枝分れ形状の生成ではシリンダとの滑らかな接続が必要である。マッピングではテクスチャの連続性が必要である。本論文では、これらの問題点をどのように解決したかを述べる。

2. 一般化シリンダのモデリング

2. 1. 一般化シリンダの定義

一般化シリンダは、二次元閉曲線を、中心軸となる空間曲線に沿ってスweepすることによって形状が定まる。これを直接レイトレーシングで表示する手法 [2] も考案されている。しかし、直接レイトレーシングを行えない枝分れ形状とすることやマッピングを行なうことを考慮すると、多面体によって形状を近似するのが現実的である。そこで、閉曲線の代わりに多角形を、空間曲線の代わりに折れ線 (空間折れ線) を用いることにした。ここで、スweepされる多角形のことを便宜上リングと呼ぶ。また、移動の軌跡である折れ線は、最終的な形状の骨組み

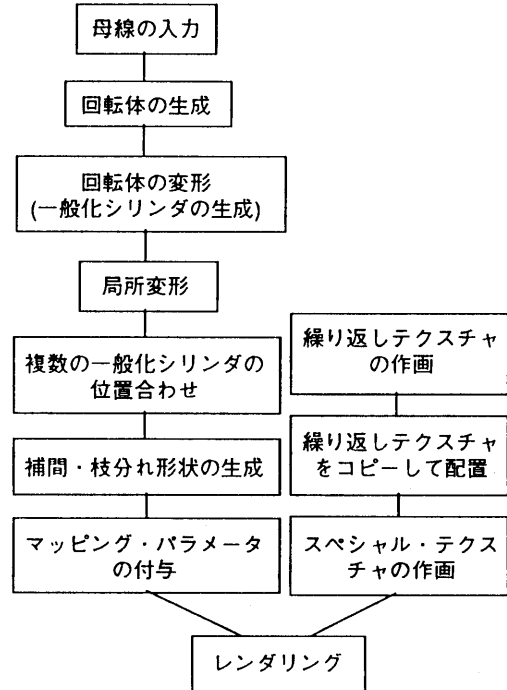


図1. 全体の処理の流れ

となることから、スケルトンと呼ぶこととする。

一個の一般化シリンダは、次のようなデータ構造を持つものとする。

$$\begin{aligned}
 \text{Skeleton} &= \{ S_0, S_1, S_2, \dots, S_{m-1} \}, \\
 S_i &= (s_{xi}, s_{yi}, s_{zi}), \quad (i = 0, 1, 2, \dots, m-1) \\
 \text{rings} &= \{ \text{ring}_0, \text{ring}_1, \text{ring}_2, \dots, \text{ring}_{m-1} \}, \\
 \text{ring}_i &= \{ r_{i0}, r_{i1}, r_{i2}, \dots, r_{i,n-1} \}, \\
 r_{ij} &= (r_{xij}, r_{yij}, 0), \quad (j = 0, 1, 2, \dots, n-1) \\
 &\quad (2-1)
 \end{aligned}$$

スケルトン (Skeleton) は $m-1$ 個の線分からなる折れ線であり、世界座標系上にある m 個の節点 S_i の点列によって表現される。一個のリング ring_i はローカル座標系の x - y 平面上にあって原点を内部に含む n 角形で、 n 個の頂点 r_{ij} の点列によって表現される。この時の原点は世界座標に変換されたときにスケルトンの節点上に来る。 ring_i を m 個並べたものがリング列 (rings) となる。ここで、便宜上すべての ring_i はお互いに相似形であるという制約条件をつけておく。

ここで、 r_{ij} は各リング毎のローカル座標系でのリングの頂点位置を表す。リングはローカル座標系ではx-y平面上に乗るから、すべての r_{ij} のz座標は値は0になっている。 $ring_i$ の各頂点 r_{ij} が世界座標系に変換されて、 S_i の周りのリングRingとなったときの頂点を R_{ij} と表記することとする。

図2にデータ構造(2-1)を説明するための一般化シリンダの例を示す。

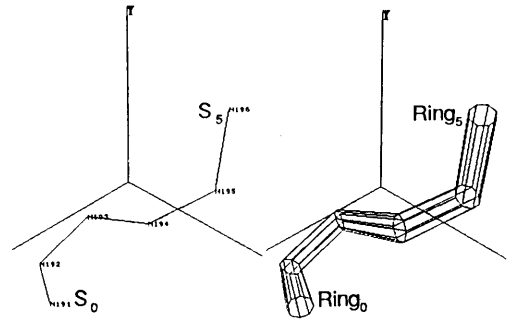


図2. Skeleton $S_0 \sim S_5$ (左)とRing $_0 \sim$ Ring $_5$ (右)

2. 2. 解決すべき問題点

一般化シリンダを生成する際には、フレネー閉包と呼ばれる一種の座標変換によってリングを移動させて形状を作る手法が知られている。しかし、この方法を多面体近似の場合に応用すると、リングの開始点を決めるのにスケルトンの折れ曲がり角を二等分する直線の向きを基準として用いることになる。そのため、スケルトンの曲がり具合が反対方向に変わる点においてはリングの開始点が180度反転し、リングのねじれが起こってしまうという問題点が生じる。この例を図3に示す。また、スケルトンの折れ線の折れの角度が0度の場合、リングの回転角が定められなくなってしまう。

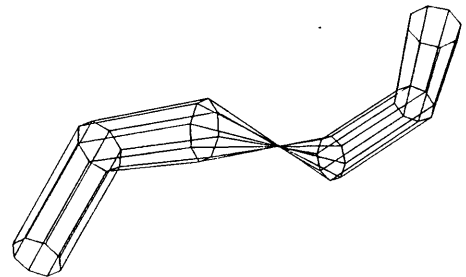


図3. フレネー閉包適用時のねじれ

2. 3. ねじれをなくすための処理手順

ねじれをなくすための基本的なアイデアは次のようなものである。すなわち、Ring $_{i-1}$ まで正しく求められていたとして、スケルトンの次の節点 S_i のまわりに来るRing $_i$ を、一回の回転変換でいったん求める。さらにRing $_i$ 上の点 R_{ij} に関して、四点 $S_i, S_{i-1}, R_{i,0}, R_{i-1,0}$ が同一平面上に来るようにRing $_i$ をねじれ方向に回転させる。(図4)

実際に一般化シリンダの形状を生成する手順について詳しく説明する。この問題は、各リングを相対座標から絶対座標に変換する問題に帰着することができる。すなわち、各ring $_i$ についてそれぞれ 4×4 の変換行列 M_i を求め、頂点 r_{ij} ($j = 1, 2, \dots, n$)を M_i によって R_{ij} に変換すればよい。そこで、Ring $_0 \sim$ Ring $_{i-1}$ の頂点の絶対座標がすでに求められている状態にあるときに、 M_i をどうやって求めるかを説明する。 M_i は、原点を通る軸回りの二回の回転移動

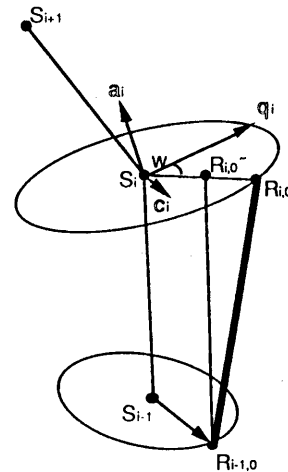


図4. 補正角 w によるねじれの補正

(step 1とstep 2) 及び一回の平行移動 (step 3) を合成した変換となる。

step 1.1 節点 S_i における進行方向ベクトル a_i を求める。

step 1.2 x-y平面上にあって a_i に垂直なベクトル

ル c_i を求める。

step 1.3 c_i を軸にして、 e_z と a_i のなす角度 v だけ回転する。

ここで e_z は z 方向の単位ベクトルである。このようにして求めた 3×3 の回転変換行列を M_{ii} とする。また、 M_{ii} によって $r_{i,0}$ (ベクトル $r_{i,0}$) が写されるベクトルを $q_{i,0}$ とする。次のステップでは、 M_{ii} の変換によって生じたねじれを補正する回転変換を行なう。

step 2.1 一つ前のリング Ring_{i-1} 上の一点 $R_{i-1,0}$ を $S_i - S_{i-1}$ の方向に平行移動して、 S_i を通って a_i に垂直な平面と交わる点を R_i^- とする。

step 2.2 $R_i^- - S_i$ と $q_{i,0}$ のなす角 w を求める。

step 2.3 a_i を軸として、 $q_{i,0}$ を w だけ回転する。

このようにして求めた 3×3 の回転変換行列を M_{2i} とする。最後のステップでは、リングの中心を原点からスケルトンの節点に平行移動する行列を求める。

step 3.1 O (原点) を S_i に平行移動する。

このようにして求めた 4×4 の平行移動変換行列を M_{3i} とする。ここで、 M_{2i} と M_{ii} を 4×4 に拡張して、

$$M_i = M_{3i} \cdot M_{2i} \cdot M_{ii} \quad (2-2)$$

を求め、ringのすべての頂点に対して変換 M_i を施し、 Ring_i を求める。その処理を、すべての i について繰り返す。こうすることによって、スケルトンが空間内でどんな経路をとってもねじれの最小な一般化シリンドラを生成することができる。図5に、step 2.1 ~ step 2.3の補正処理を行なった場合と行なわなかった場合の違いを示す。なお、この方式では、スケルトンの急激な屈曲によっておこる表面の食い込みについては考慮していない。

2. 4. 各処理ステップの詳細

次に各ステップでの処理を数式を用いて説明する。

step 1.1は前後の節点座標との位置関係によって次のように求められる。

$$a_i = (s_i + s_{i-1}) / |s_i + s_{i-1}| \quad (2-3)$$

ここで、 $s_i = (S_{i+1} - S_i) / |S_{i+1} - S_i|$ である。次の step 1.2は

$$c_i = (a_y, -a_x, 0) \quad (2-4)$$

として得られる。ここで、 $a_i = (a_{ix}, a_{iy}, a_{iz})$ である。

step 1.3の回転角 v は、

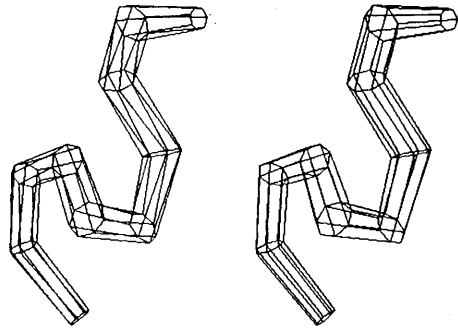


図5. ねじれ補正のない場合 (左) とある場合 (右)

$$v = \text{Arccos}(a_i \cdot e_z) \quad (2-5)$$

となる。ある軸のまわりにある角度だけ回転する行列は、文献 [5] を参照されたい。step 2.1では次のような連立方程式を解くことによって R_i^- を求める。

$$R_i^- = R_{i-1,0} + t(S_i - S_{i-1}) \quad (t \text{ はスカラー}) \quad (2-6)$$

$$a_i \cdot (R_i^- - S_i) = 0 \quad (2-7)$$

式 (2-6) は四点 R_i^- , S_i , S_{i-1} , $R_{i-1,0}$ が同一平面上にあることを意味している。step 2.2の計算は、 $r_i^- = R_i^- - S_i$ とおくと、

$$w = \text{sign}(a_i \cdot q_{i,0} \times r_i^-) \cdot \text{Arccos}(|r_i^- / |r_i^-|| \cdot (q_{i,0} / |q_{i,0}|)) \quad (2-8)$$

となる。step 2.3はstep 1.3と同様に計算する。step 3.1の計算は省略する。

2. 5. 一般化シリンドラの会話モデリング手順

実際のモデリングの手順を図6に示す。大きな流れとしては、まず回転体を作り、次に回転軸を折り曲げてほしいの形を作り、最後に個々の点を少し移動する微調整を行なうという作業を行なう。さらに必要ならば、補間によってより細かい多角形に分割することもできる。

回転体を作る手順は以下のようになる。まず y 軸に沿って、回転前の母線となる折れ線を描く。これはマウスを使って x - y 平面図上でカーソルを動かし

ながらボタンを押して描く。描き終わった後実行メニューをクリックすると、y軸まわりの回転体が生成される。厳密には回転によって、母線の各節点は円ではなく、正多角形を描く。これらの正多角形が一般化シリンダのリングとなる。

次に、回転体の軸すなわちスケルトンを表示させる。スケルトンは直線状になっているが、実際は複数の線分からなる。スケルトンの各節点は、最初に作った折れ線の各節点の回転中心点となっている。スケルトンの節点をマウスで選択して移動することにより、スケルトンの形状を変えることができる。x軸方向、y軸方向、z軸方向のそれぞれからこの操作を行なうことにより、スケルトンの形状は、任意に変えることができる。スケルトン形状の変更後、実行メニューをクリックすることにより、回転体がスケルトンに沿って変形する。このようにして変形した回転体は、一般化シリンダと見なすことができる。

変形後の各頂点、すなわちリング上の各頂点をマウスで選択して少しずつ移動することにより、形状の局所変形が可能である。この際、リングが常に同一平面上に乗るという制約条件を加える。

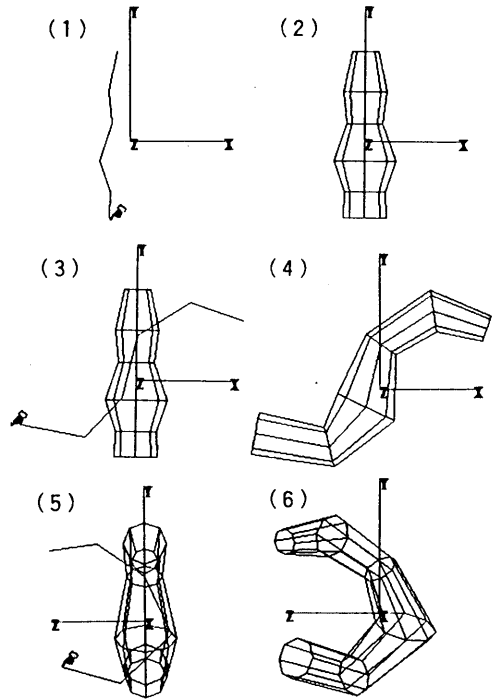


図6. モデリング手順

3. 枝分れ形状の生成

以上のようにして作った一般化シリンダを三つ集めてY字型に位置合わせを行ない、枝別れ部分の形状を補間により新たに発生させる。

3. 1. 一般化シリンダの位置合わせ

モデリングした一般化シリンダは、平行移動と回転によって任意の位置に任意の向きで配置することができる。そのような操作によって、枝分れ形状を作る前の位置合わせを行なった状態を図7に示す。

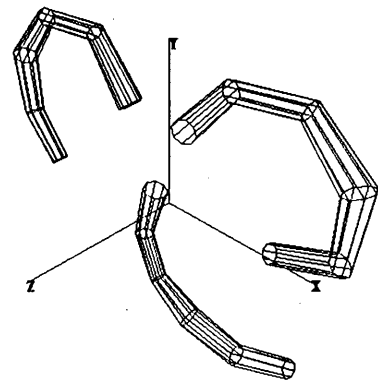


図7. 一般化シリンダの位置合わせ

3. 2. 枝分れ形状の構成

枝分れ形状は五枚のバッチによって構成する。そのうち三枚は円柱を縦に半分に割ったようなサドル型をしており、それぞれ二つの一般化シリンダを接続する。残り二つは枝分れの中央部分に位置するデ

ルタ型のバッチで、手前側と裏側に位置する。この様子を図8に示す。ここで同時にシリンダ部分も補間する。

ここで、三本のシリンダがお互いになす角度が最

も小さい分岐の対面のシリンダを主シリンダとし、これをcylinder0と呼ぶ。他の二つのシリンダをそれぞれcylinder1, cylinder2と呼ぶ。cylinder0, 1, 2の対面にあるサドル型パッチをそれぞれsaddle0, saddle1, saddle2と呼ぶ。デルタ型のパッチをdelta1, delta2と呼ぶ。

3. 3. 補間方法

パッチによる補間方法は、Ebomentalの方法 [4] を用いた。すなわち、図9に示すように、はじめに、半円状の曲線セグメントABCを求める。この曲線セグメントはパッチの位置を定める。次にsaddle1をcylinder2とcylinder0の間の補間によって求め、saddle2をcylinder1とcylinder0の間の補間によって求める。点A、点Cはsaddle1, saddle2がねじれてしまわないように選ぶ。点Bの位置はcylinder1とcylinder2の形状から補間して決める。delta1とdelta2の補間は、本来なら、三角形パッチ用の補間 [7] を用いるべきである。しかし、補間後のポリゴンとしてあまり細かいものを想定していないため、実用上は四角形パッチで十分であった。補間は3次のB-スプライン関数 [6] を用いた。

曲線セグメントABCの形状を調整することによって、枝分れ部分に一種のテンションを与えることができる。この様子を図10に示す。

4. マッピング

4. 1. 解決すべき問題点

一般に、二次元のテクスチャを三次元物体の表面に貼り付ける場合、次のようなことを考慮する必要がある。

- (1) できるだけ少ない種類の原画像テクスチャを用いる。
- (2) 同じテクスチャを繰り返し貼り付けたときに境界が連続になるようにする。
- (3) 滑らかな面上にはテクスチャの境界が現れない。

枝分れ物体の場合、パッチの境界面はすべて滑らかであるから、テクスチャの不連続な部分が現われて

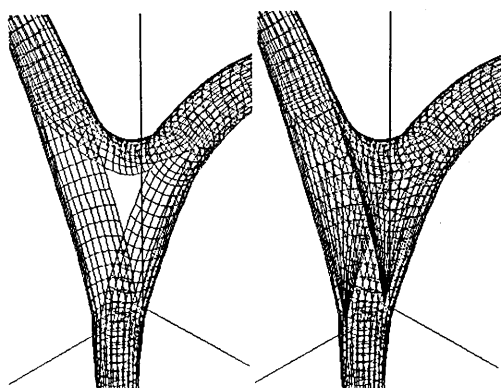


図8. 枝分れ形状を構成するパッチ

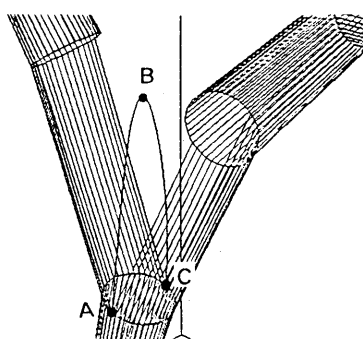


図9. 枝分れ中央の曲線セグメントの生成

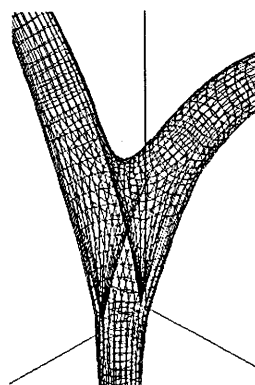


図10. テンションの付与による形状の変化

はならない。しかし、四角形の画像を貼り付けていく通常の方法では、枝分れ部分のどこかで必ず不連続な境界線が生じる。これは、3. で説明したdelta1, delta2のような三角形のパッチを使用しているため

である。

4. 2. 繰り返しテクスチャ

図11に枝分れ部分のパッチの構成を示す。このうち、表面の白い部分は、種類の四角いテクスチャを繰り返し貼り付けることによって、すべての点において連続にすることのできる部分である。

本システムのレンダリング・アルゴリズムでは、物体表面を近似するポリゴンのすべての頂点に (u, v) のパラメタを与えることによってマッピングを行っている。 u および v の値は0以上の実数とする。ここで、物体表面上で一枚のテクスチャ画像が貼り付けられる領域は

$$R = \{(u, v) \mid i \leq u \leq i+1 \text{ かつ } j \leq v \leq j+1, i, j \text{ は } 0 \text{ 以上の整数}\} \quad (4-1)$$

で表現される表面領域 R となる。同じテクスチャ画像を繰り返し貼り付けたときに全体が連続になるには、そのテクスチャの上端と下端、左端と右端がそれぞれ連続的につながるように作画する必要がある。このようにして作ったテクスチャを繰り返しテクスチャと呼ぶこととする。なお、パッチ同士の境界部分は u, v ともに整数となるように u, v の値を与えれば、異なるパッチの間でもテクスチャは連続的につながる。

4. 3. スペシャル・テクスチャ

図11で網かけで示したパッチ、すなわち、 delta1 , saddle0 , delta2 の三つについては、4. 2. で使用した繰り返しテクスチャと同じものを使うことはできない。 saddle0 に関しては一見そのまま貼り付けてもよさそうに見える。しかし、 cylinder1 との境界と、向い側の cylinder2 との境界では、パラメタ u の値の増減が逆向きになる。そのため繰り返しテクスチャは使用できない。

ここで、枝分れ部分のパッチの展開図を描くと図12のようになる。同じ矢印をつけた辺同士はその向きで接続されることを示す。 delta1 , saddle0 , delta2 からなる六角形の部分は各辺がすべて繰り返しテクスチャと接している。もし、繰り返しテクスチャの貼られかたが決まれば、六角形部分をうまく作画す

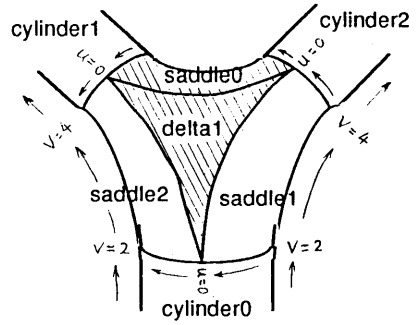


図11. パッチの構成

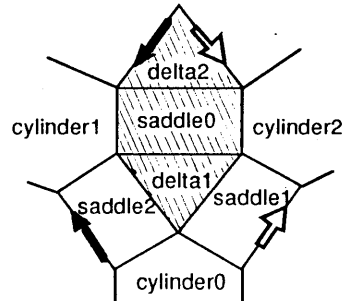


図12. 枝分れ部分のパッチの展開図

ることによって、繰り返しテクスチャの部分と連続になるようにできるはずである。

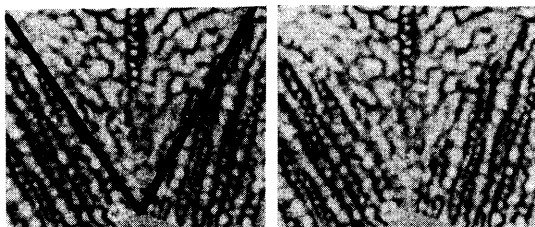
これを実現するために、六角形部分を作画する前に、そのまわりに繰り返しテクスチャを適切に描くようなペイントプログラムを作った。デザイナーは空白になっている六角形部分を埋めるように作画すればよい。実際には、六角形部分を二つの三角形パッチと一つの四角形パッチとに分けて作画している。

4. 4. マッピング・パラメタの与え方

枝分れ物体全体の形状データは各パッチ毎に別々にファイルに収める。各パッチはすべて $m \times n$ (m, n は正の整数)のメッシュ状に配置された頂点からなっている。それぞれのパッチについて、各コーナーの四点における u, v の値を与えれば、残りのすべての頂点についてリニアに補間されて自動的に計算される。 delta1 , delta2 の一端(図9における頂点AおよびC)については、複数の頂点と同じ位置にあるものと見なしている。

4. 5. マッピングの例

図13に、本方式を用いてマッピングを行なった例を示す。上の二つの写真は同じもので、中央の三角形の部分がdelta1にマッピングするスペシャル・テクスチャである。両側に繰り返しテクスチャのコピーが配置されている。左下は繰り返しテクスチャである。右下は実際に枝分れ物体にマッピングした例である。



5. まとめ

一般化シリンダを用いた枝分れ物体のモデリングとマッピングの手法について述べた。今後の課題として、一般化シリンダの自己干渉（食い込み）の問題や、三分木以上の一般的な分岐の扱いなどが挙げられる。

6. 謝辞

モデラの開発に貢献してくれたSSLの寺田光男氏、富士通花の万博推進本部システム部の岡野秀樹氏、講演等を通じて協力してくれたDoug Lerner氏に感謝いたします。また、日頃からお世話になる富士通研究所の石井部長、森田部長、ヒューマンインタフェース研究部CGグループの村上氏、林氏、広田氏、藤田氏に感謝いたします。

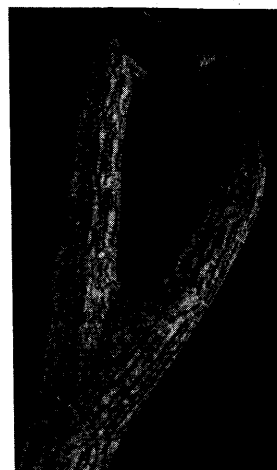
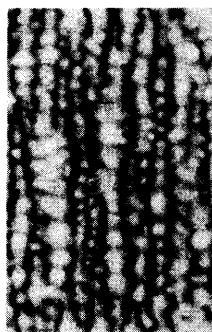


図13. マッピングの実現例（上二つ：スペシャル・テクスチャ、左下：繰り返しテクスチャ、右下：マッピング結果）

【参考文献】

- [1]. Shani, Uri and Dana H. Ballard, "Splines as Embeddings for Generalized Cylinders," Computer Vision, Graphics, and Image Processing 27 (1984), pp.129-156.
- [2]. Bronsvoort, Willem F. and Fopke Klok, "Ray Tracing Generalized Cylinders," ACM Transactions on Graphics, Vol.4, No.4, October 1985, pp.291-303.
- [3]. Klok, Fopke, "Two moving coordinate frames for sweeping along a3D trajectory," Computer Aided Geometric Design 3 (1986), pp.217-229.
- [4]. Bloomenthal, Jules, "Modeling the Mighty Maple,"

ACM Computer Graphics, Vol.19, No.3, July 1985 (SIGGRAPH '85), pp.305-311.

- [5]. 山口富士夫, 「コンピュータディスプレイによる図形処理工学」, 日刊工業新聞社, 1981年6月.
- [6]. 山口富士夫, 「コンピュータディスプレイによる形状処理工学[II]」, 日刊工業新聞社, 1982年12月
- [7]. Catmull, E. and J. Clark, "Recursively generated B-spline surfaces on arbitrary topological meshes," Computer-Aided-Design Vol.10, No.6, November 1978, pp.350-355.