18

3次元樹木状モデルのモデリング法

上原慎矢* 高井信勝
(北大工)† (北海学園大工)

1 はじめに

3次元グラフィックスシステムにおいて,自然物のモデルをモデリングするために,自然物のなかにフラクタル図形となっているものがあり,フラクタル図形が自己相似性という性質をもっている[1]ことを利用して,関数の再帰呼び出しを用いて各種の形状を得ることができる。[2]

ここで説明しているモデリング法も基本的には関数の再帰呼び出しを利用している。樹木状のモデルをモデリングするにあたり,まず枝・幹は,3角形を上・底面とする角柱に簡単化し,これをつなげていくことで3次元の樹木状のモデルを生成する。再帰呼び出しされる関数は,3次元空間での角柱の位置と向きを決めるパラメータを受け取り,これをもとに3次元空間に1つの角柱を配置するものとし,再帰呼び出しによって次の同じ関数に渡されるパラメータの与え方に,一定の規則を与えることによって,樹木状のモデルを作成する。

2 座標系

このプログラムの座標系の取り方は左手系で考えているので,以降の記述では,座標系として左手系を前提としている。

このプログラムで考える座標系は,枝座標系,ワールド座標系の2つの座標系であり,枝座標系で,表示する角柱(枝)の頂点座標を与える。この座標系は,角柱の形状を定義するために使う。ワールド座標系は,実際に角柱が配置される座標系である。即ち,枝座標系で定義された角柱の頂点座標を,変換によってこの座標系での表現にかえていくことになる。

3 枝の成長

枝の座標系では,底面をxz平面内に置き,(このとき,3角形の重心を原点と一致させる。これは,枝の連結を適切に行うためである。)上面はxz平面に平行で,+y方向にある量だけ平行移動したものを考える。つまり,このy軸方向の平行移動量が1回の関数呼び出しによって生成される枝の長さになる。このy軸方向の平行移動量を再帰呼び出しの深さが深いほど小さくすると,再帰呼び出しが深いほど,即ち枝の先端に近いほど配置される枝が短くなる。

プログラムでは,ユーザーが与える長さの初期値をL,再帰呼び出しの深さをNとするとき,長さをL/Nにしている。

4 再帰呼び出しに必要なパラメータ

4.1 枝の方向ベクトルg

ワールド座標系でベクトルgが与えられたとき,このgの方向と枝座標系のy軸方向を一致させるという考え方で,各枝をワールド座標に配置するようにする。このようにすると,ベクトルgの方向に枝が伸びていくので,このベクトルgを以下,枝の方向ベクトルと書く。また,このベクトルgの大きさを,3節でのy軸方向の平行移動量に用いている。従って,ベクトルgは,向きが枝の成長方向で,大きさが枝の長さになっていて,基本的に関数の呼び出し毎に成分が異なる。

4.2 ワールド座標系における枝座標系の原点の位置ベクトルm

枝の方向ベクトルgだけでは,ワールド座標系での各角柱の向きと長さは決められるが,位置は決められない。位置を決めるために,もう1つのベクトルmを導入する。このベクトルmは,ワールド座標系における枝座標系原点の位置ベクトルである。これらベクトルg,mがあれば,独自の枝座標系で定義された各角柱をワールド座標に配置することができる。

ここまでの項目を図にまとめるとFig.1のようになる。この図で,枝座標系をx'y'z'で,ワールド座標系をxyzで表している。

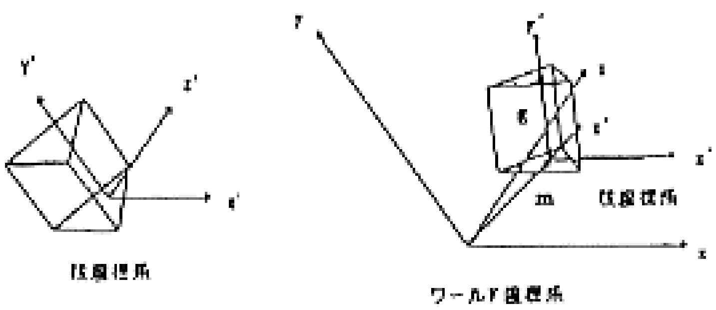

Fig. 1 枝,ワールド座標系とベクトルg,m

5 実際の各角柱の配置

各角柱をワールド座標に配置することを考える。はじめに,枝座標系とワールド座標系が一致していると考えると,角柱の底面はワールド座標系のxz面内にあり,原点と重心が一致している。

次に,ワールド座標系でのベクトルgの向きを回転によってy軸に一致させることを考える。Fig.2を用いて

*sinya_uehara@msn.com
†札幌市北区北13条西8丁目北海道大学大学院工学研究科

説明すると，先ず,Fig.2(b) からわかるように,z 軸の回りに角 $\varphi = \tan^{-1}(gx/gy)$ だけ回転するとベクトル g は,yz 平面上にのる。この状態が Fig.2(c) である。この図から，さらに x 軸回りに角 $\theta = \tan^{-1} gz/\sqrt{gx^2+gy^2}$ だけ回転するとベクトル g が y 軸に一致することが分かる。このことから逆に，枝座標系の y 軸をベクトル g に一致させるには，まず x 軸回りに図形を角 $-\theta$ だけ回転し，次に z 軸回りに図形を角 $-\varphi$ だけ回転すればいいことになる。実際はさらにこの後，ベクトル m の分だけ平行移動させる。

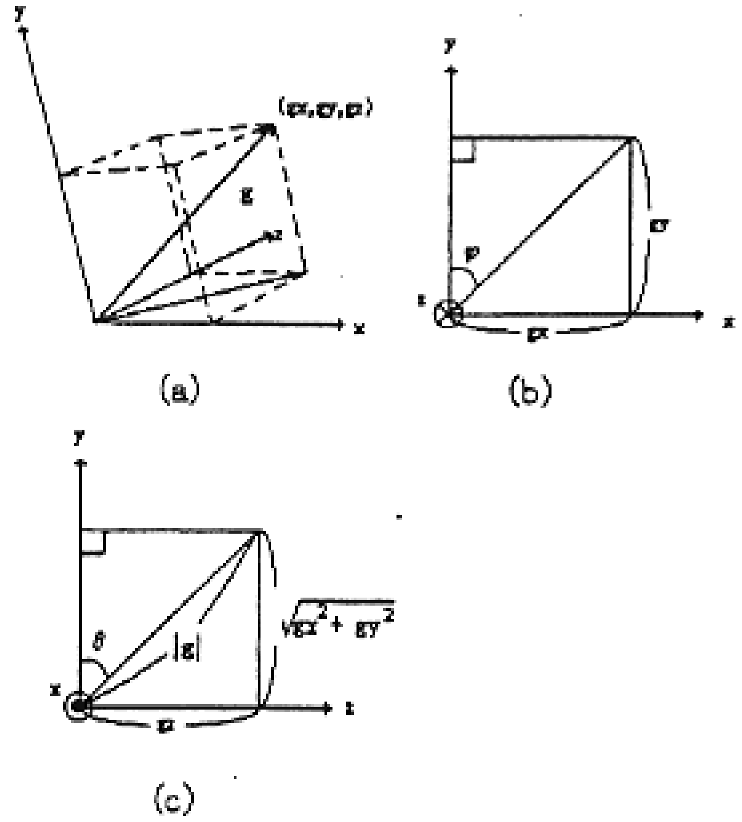

Fig. 2 ベクトル g と角 φ,θ の関係

6 枝の分岐

樹木らしいモデルを生成するために枝を分岐させる必要がある。この手順では，枝の伸び方を次の 3 つの場合に単純化している。

6.1 まっすぐ伸びる場合

まず，分岐しない場合を考える。この場合はきわめて簡単で，枝の方向ベクトルの向きは変えず，大きさだけを小さくする。また，枝座標系の原点の位置ベクトル m を,m + g に変更する。この新たなパラメータを用いて関数の再帰呼び出しを行う。この場合だけしか考えない場合の生成結果例は Fig.3 のようになる。

6.2 2 分岐する場合

この場合は,2 つの角柱が必要となるので，それぞれの角柱に対応して再帰呼び出しを 2 回することになり，そのためのパラメータを準備する必要がある。

2 分岐する場合,1 つ行わなければならないことが増える。それは，三角形の分割である。三角形の分割をしないまま角柱を 2 分岐させると，分岐を繰り返しても枝が

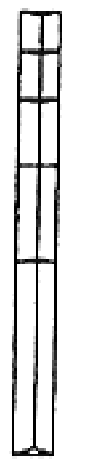

Fig. 3 まっすぐ伸びる場合

細くならない不自然なものになるので，三角形の分割を行う。この手順で用いている方法は，次のようなものである。三角形の最長の辺を選びその中点を算出する。次に，その辺の対になる頂点と，先の中点を結ぶ線分を考え，この線分で分割する。

分割した各三角形は，重心が原点と一致するようにする。これを怠ると，場合によっては三角形の外部の点を原点とした回転・移動が行われ，枝の連結がうまくいかないことがある。さて，三角形を 2 分割すると各三角形に重心があるので，これから分岐していく枝の前段の枝の底面の重心と，上面の 2 つの三角形の重心をそれぞれ結ぶベクトル ga,gb を考える。これを用いて枝座標系の原点の位置ベクトル ma,mb を，それぞれ $\mathbf{ma} = \mathbf{m} + \mathbf{ga}$, $\mathbf{mb} = \mathbf{m} + \mathbf{mb}$ とする。

最後に枝の方向ベクトル ga,gb を考える。ベクトル $\mathbf{ga} - \mathbf{g}$, $\mathbf{gb} - \mathbf{g}$(ベクトル g は，まっすぐ伸びる場合のベクトル。) を考えると，これはまっすぐ伸びる方向と内積が 0 になる方向に，しかもそれぞれが外側に広がるように伸びている。従ってこのベクトルの大きさをベクトル g の大きさと同じにしたベクトル ga′,gb′ を用いた次のベクトルを考える。ここで,α,β はそれぞれ 0 から 1 の範囲の任意の実数とする。

$$\mathbf{GA} = \alpha\mathbf{ga}' + (1-\alpha)\mathbf{g}$$

$$\mathbf{GB} = \beta\mathbf{gb}' + (1-\beta)\mathbf{g}$$

するとこれは,α,β の値により，まっすぐ伸びる場合と，その方向に対して垂直な方向に伸びる場合の範囲で向きが変わることになる。これをそれぞれの枝の方向ベクトルにする。また，大きさの変更もする。

これらのことを図にまとめると Fig.4 のようになる。また，この場合だけしか考えない場合の生成結果例は Fig.5 のようになる。

6.3 上記 2 つの場合の混合

上記 2 つの場合を組み合わせると，まっすぐ伸びる枝 (これは，太さが変わらない) のわきから，細い枝が伸びていく場合が再現できる。これはさらに，分岐した枝 2 本

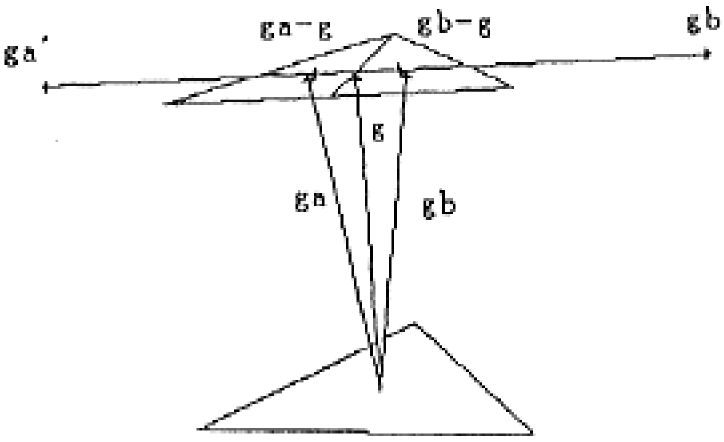

Fig. 4 各ベクトルの関係

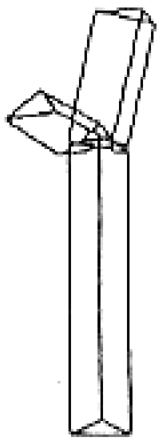

Fig. 5 2分岐する場合

をまっすぐ伸びる枝に混ぜて伸ばす場合(合計3本),分岐した枝1本だけを混ぜる場合(合計2本)の2通りがあり,1本だけを混ぜる場合は,どちらの枝を使うかでさらに場合分けができる。

この場合だけしか考えない場合の生成結果例はFig.6のようになる。

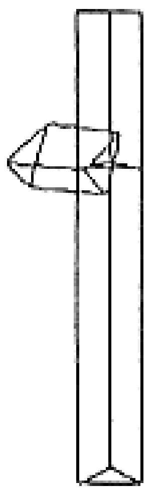

Fig. 6 まっすぐ伸びる枝と分岐する枝の混合

さて,これらの3通りのどれを使うかを乱数により決めれば生成毎に異なった樹木が得られる。さらに,上記3つの場合について,まっすぐ伸びる場合は乱数を使う余地がないが,他の場合乱数を使うことができる。α,βの値と,混合する場合のさらに内部での3通りのどれを選ぶかである。これらも乱数を使うことにより,複雑な成長をさせることができる。

7 実験・結果

この手順によって生成されたデータをもとに,表示を行った結果をFig.7に示す。

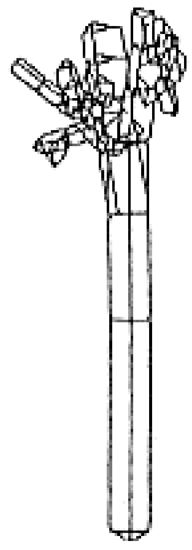

Fig. 7 生成されたデータからの表示例

8 おわりに

3角形の分割において中点変位法をもちいたり,枝の方向ベクトルの大きさをランダムに微小量だけ変化させるなどの方法で,さらに自然なモデルを作成できることは容易に想像できるが,ここでは行っていない。また,干渉問題や,特定の樹木を再現することなども考慮していないので,ここでの方法はまったく基本となる部分だけであり,今後改善していこうと考えている。

また,2次元フラクタル図形の生成・解析を,w変換という新しい考え方で行う研究が報告されており[3],このw変換理論を3次元フラクタル図形に応用することについても考えていきたい。

参考文献

[1] 高安秀樹"フラクタル"(朝倉書店,1986)

[2] 芹沢浩"3次元フラクタル紀行"(森北出版,1995)

[3] 青木由直"フラクタル図形生成におけるオペレータ法と葉脈曲線解析への応用" 電子情報通信学会論文誌D-II,vol.J82-D-II,No.4(April,1999)