

カラーグラフィックスを用いた地形シミュレーション及びフラクタルの応用

九州工業大学 正会員 山本 宏
九州工業大学 学生員 藤崎 文実博

まえがき

従来、地形のシミュレーションを行う場合、ワイヤーフレームモデルを用いたメッシュ図が多く使われていたが、このモデルは形状を把握することは容易であるが質感を伴わないので景観を評価しようとするには、あまりに情報量が少ない。そこで、本研究ではサーフェイスモデルを用いて地形のカラーシミュレーションを行うと共に、コンピュータ・グラフィックスの分野で最近注目されているフラクタルといった新しい理論を取り入れて考察をおこなった。なお、計算にはMELCOM COSMO 800IIIを、ディスプレイには JRC の NWX-235を用いた。

1, データの作製

サーフェイスモデルで地形をシミュレートするのに最終的に必要なデータは、単位要素を構成する各頂点の三次元座標であるが、ここでは地表面を三角形の有限要素に分割して近似しているため、1要素について9個のデータが必要になる。しかし、あらかじめメッシュを切って順序付け、有限分割を行うならば、必要になるデータは各交点の高さ及びメッシュの数と大きさだけになる。したがって地形をワイヤーフレームモデルで表現した場合と同じデータを用意すればよい。

2, 透視変換

透視変換は、三次元空間で定義された物体を二次元の平面に変換するもので、ここでは三次元左手系で定義したデータを視点 (X_f, Y_f, Z_f) を原点とする三次元右手系に変換し、それから注視点 (X_v, Y_v, Z_v) が、画面の中央に来るようにした二次元の平面に変換する。今、入力データを X, Y, Z とし、変換後のデータを X_p, Y_p と置くと座標変換マトリクスを R として

$$R = \begin{bmatrix} \cos(\alpha) \cdot \cos(\beta) & \sin(\beta) \cdot \cos(\alpha) & \sin(\alpha) \\ -\sin(\beta) & \cos(\beta) & 0 \\ -\cos(\beta) \cdot \sin(\alpha) & -\sin(\alpha) \cdot \sin(\beta) & \cos(\alpha) \end{bmatrix}$$

$$XD = L \cdot (R_{11} \cdot (X - X_v) + R_{12} \cdot (Y - Y_v) + R_{13} \cdot (Z - Z_v))$$

$$YD = R_{21} \cdot (X - X_v) + R_{22} \cdot (Y - Y_v) + R_{23} \cdot (Z - Z_v)$$

$$ZD = R_{31} \cdot (X - X_v) + R_{32} \cdot (Y - Y_v) + R_{33} \cdot (Z - Z_v)$$

$$X_p = (YD \cdot L) / XD \quad Y_p = (ZD \cdot L) / XD$$

$$L = \sqrt{(X_f - X_v)^2 + (Y_f - Y_v)^2 + (Z_f - Z_v)^2}$$

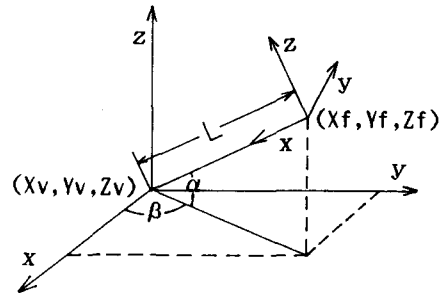


図-1 透視変換

以上の式で表すことができる。

3, クリッピング・陰線消去

クリッピングは、視野の外にある物体を画面から除く作業で、これには Sutherland のアルゴリズムを用いた。これは、空間を図-2 のような領域に分けて、線分を対象にして、その両端の点に4ビットのコードを与え、視野の内か外かを定めていくアルゴリズムである。

陰線消去には、ディスプレイの塗り潰し書き込みの機能を用いて、視点よりも遠くの要素から順に表示させていくことで解決した。

$$Y = -X \quad Y = X$$

1001	1000	1010	
			Z=X
0001	0000	0010	
			Z=-X
0101	0100	0110	

図-2 クリッピング・コード

4, 色の決定

三角形要素の色は、はじめに色相を定め、その平面の表向き法線ベクトルと、あらかじめ入力しておいた太陽の方向ベクトルとの内積の値によって濃淡を決定した。濃淡は14段階用意してある。

5, 影・水面

三角形要素の色を決める際に、太陽の方向ベクトルとの内積を取ったが山が地表や水面などに落とす影を考慮していない。そこで、別のアルゴリズムによって、影を付加することにした。まず、内積の値より影を作る要素群を判別する。次に、視点を太陽と同じ方向に取り、この影要素群に隠される三角形要素群を見つける。そして、この要素の色を暗くする。これで影が付加される。

地形の中に水域を含む場合は、水面の高さを与えて水面の平面式を作り、それよりも上側の三角形要素はそのまま表示させ、下側の要素は水面に変換し色を変えて表示させた。

6, フラクタルの応用

地表を有限分割してシミュレーションする場合、複雑な地形であっても分割数を増すことで、かなり高い近似を行える。しかし、その手間はメッシュの数の2乗に比例して増加して行くため、単にメッシュを細かく取るのではなく、その地形の特徴が埋もれないくらいの大きさに取り、メッシュの間を何等かの方法によって補間していくのが有効ではないかと思われる。ここではそのためにフラクタルを用いた。

フラクタルという言葉は、Mandelbrotが、ラテン語の「FRUCTUS」より造った人造語で、「半端な」といった意味を持つ。フラクタルと呼ばれる図形は、整数で与えられる次元からはみ出した図形のことを言い、この図形は自己相似という特徴を持っている。これは、図形のある一部分を取っても、また全体を取っても、それらが同じ構造をしているといった性質で、形状は変わってもその統計的性質が変わらないものを統計的自己相似と呼ぶ。地形シミュレーションに応用しようとするのは、フラクタルのこの性質である。地形は、様々にスケールを変えてみても同じような性質の形状が表われてくる。これは、地形の持つ自己相似性であるが、これを計算機でシミュレーションするには、あるスケールではその形状が簡単な数式で表わされる図形であると仮定し、スケールを変える毎にそれに則した乱雑さを加えながら重ねていくといった方法を用いる。具体的には、三角形要素を基本の図形に取り、図-3のようにその各辺の中点を乱数で高さ方向（重力方向）に変化させ、各頂点と結ぶ。これで新しい4つの三角形ができあがるが、この1つ1つに同じことを繰り返して行くことで地形としてのフラクタル図形ができあがっていく。問題となるのは、乱数の与え方で、スケールに則した乱雑さを再現するために、ここでは三角形要素の辺の長さに比例した標準偏差を持ち、中点を中央値に持つ正規分布乱数を用いた。

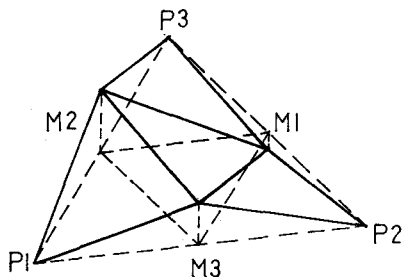


図-3 要素分割

あとがき

図-4, 図-5は、縮小したモノクロ写真で表示すると非常に判別しにくくなるため、OHPによってカラーで拡大表示する。

図-4 サーフフェイス

図-5 フラクタル

フラクタルを用いることで図-4,5に示したようになかなりリアルなシミュレーションを行うことができた。しかし、実際の地表のテクスチャーとは、まだかなり離れており考慮に入れているフラクタルも少ない。だがこの手法は、自然の地形といった数学では扱いにくいものを計算機を用いてシミュレーションするのに非常に有効な手段の1つであると思われる。

<参考文献>

- 1) 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)