

等高線からの山岳形状の再生

福 島 誠*

Makoto FUKUSHIMA

A Reconstruction of Mountains from Contour Lines

Abstract: In this paper, a method is proposed for reconstructing mountains from 3-Dimensional(3D) data obtained from the tracing of contour lines.

The thinning and the noise reduction are carried out after feeding the contour lines to the personal computer through image scanner. Then the 3D data of the points on the contour lines can be obtained from the tracing of the contour lines. Moreover, the 3D data of the points between the contour lines are calculated by using the linear interpolation and the lattice model based on the contour lines. Since these points are regarded as the vertices of rectangular patch, the surface model using the rectangular patches can be reconstructed with hidden-surface and shading algorithms.

1. ま え が き

コンピュータグラフィックス（CG）の応用はいろいろ考えられるが、とくにこれによる自然形状の再生は、フライトシュミレーション、景観シュミレーション等の分野に加えて教育のための教材作成にも有用であると考えられる。

CGによる自然形状の再生のうち特に等高線による山岳形状の再生は、より現実的な把握が容易な教材としても利用できる可能性がある。ただし、自然の等高線の形状は簡単なものから複雑なものまで多岐にわたっており、これを全てCGとして忠実に再生することは容易ではない。しかし、等高線の形状をある程度限定すれば3次元サーフィスマodelの作成はパーソナルコンピュータ（パソコン）でも可能である。

等高線データからの3次元モデルの再生上重要で、場合によっては面倒なのが等高線の座標データの入力である。これには、通常2種類の方法が考えられる。すなわ

ち、1. デジタイザによる入力、2. ドラムスキャナ（イメージスキャナ）からの入力である。1.の方法の場合は、入力されたデータはそのまま座標データとして処理可能であるが、2.の方法の場合は、スキャナから入力された等高線を追跡して座標データに変換する作業が必要になる。等高線の形状が複雑になるほど、1.の方法はデータ入力作業にかなりの労力を必要とするのに対して、2.の方法では等高線追跡のプログラムなどが用意されていれば、等高線の入力ははるかに簡単である。しかしスキャナの読取ドット数の増加に比例して座標データの数が増加することと、スキャナ入力の際の雑音対策も必要であるなど問題点もある。ただし、このうちデータ量の増加については等高線の特徴点を抽出して、ベクトルデータとして近似する手法も提案されており⁽¹⁾、必ずしも不利とはいえない。

本報告では、データ入力の容易さを優先して、2.のスキャナ入力の方法を採用する。すなわち、等高線入力後、パソコンで細線化と雑音の除去及び等高線の追跡を行い、得られた3次元データよりサーフィスマodelを再生する方法を提案する。この方法によりかなり複雑な等高

* 島根大学教育学部技術科研究室

線でも、等高線の点に対応する座標データを得ることがより少ない労力で可能である。ただし、データ量の圧縮をするよりもできるだけ自然形状に近い3次元モデルの再生を目的としたので、データの圧縮は行っていない。

2. 等高線入力と追跡によるサーフィスモデルの再生

2.1 システムの処理の概要

等高線の入力から山岳形状の再生までの、処理の流れの概要は図1に示すとおりである。また、処理システムの構成は図2に示すとおりである。

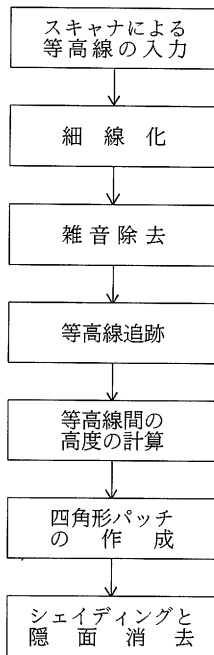


図1 処理の概要

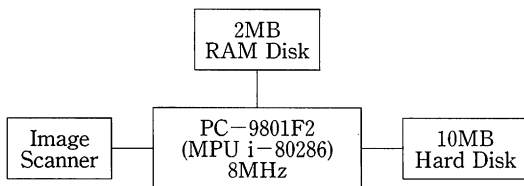


図2 システムの構成

まず等高線入力のためのイメージスキャナにより、等高線図を200dpiの解像度で2値画像データとして入力する。入力されたイメージデータの画素数は、640×400ドット

で、1画面分のデータは、32KBになるが、イメージスキャナはカラー対応のため、実際には96KBになる。

2.2 細線化と雑音除去

入力された等高線のイメージデータは、原図の等高線の太さによって線幅は1ドット以上になる場合が多く、また幅も一定ではない。従って後述するように、等高線を追跡して各ドットを座標データに変換するには、このままでは適さないで細線化を行う。細線化のアルゴリズムはいくつか提案されているうち、Hilditch⁽²⁾のアルゴリズムを採用した。このアルゴリズムでは良く知られているように線幅がある程度太くなると、細線化の過程で髭状の雑音が発生する⁽³⁾。等高線を追跡するためにこの髭状雑音は不都合なので取り除く必要がある。これを取り除くためのアルゴリズムは以下に示すような簡単な方法を採用した。

まず前提条件として、{等高線は全て閉曲線であり、交差することはない。}と仮定する。

- ① 画面上の3×3ドットの局所パターンにおいて、端点を見つけ、その(x, y)座標を配列に格納する。
- ② ①と同様な3×3ドットの局所パターンにおいて交点を見つけ、その(x, y)座標を配列に格納する。
- ③ 端点の配列と交点の配列との距離を計算し、最短距離の端点と交点の組み合わせから、端点から交点までの線を消去する。

端点と交点の組み合わせを確認するのは、等高線の途切れている点を端点と見なして消去するのを防ぐためである。従って、上記のアルゴリズムでは等高線の途切れ点、雑音の端点よりも交点近くに存在する場合には不適であるが、ここではアルゴリズムの簡易さを優先して採用した。

2.3 等高線の追跡

雑音除去の終了後、等高線を画面上で追跡して各ドットの(x, y, z)座標の値を決定する。(x, y)座標は画面上のドット位置の座標を基にして決定する。この場合、画面の中央を原点として、座標系は図3のような位置関係とする。z軸方向の値、すなわち、等高線の高さのデータは等高線(閉曲線)が同心円状であれば、基準等高線の値と、等高線間の偏移値を与えれば自動的に決定できる。しかし、峰などが複数あるような場合には等高線間の包含関係を定義する必要があり簡単ではない⁽⁴⁾。本報告では、等高線を1本追跡するごとに、その等

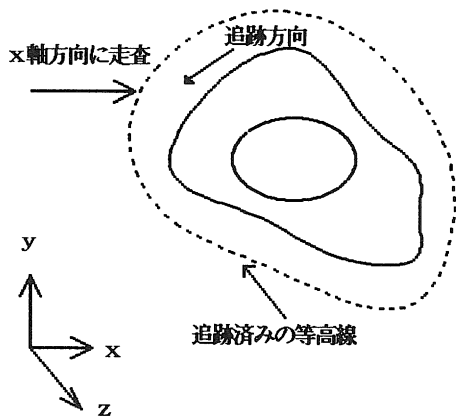


図3 等高線の追跡

高線の高さ (z 座標値) を入力するようにしている。また、 (x, y, z) 座標のデータ長にはそれぞれ、2, 2, 1バイトを割り当てているが、 (x, y) 座標は配列、またはファイルのデータ位置に対応させると、追跡後の640×400ドットの画面の座標データは最大で256KBになる。本論文で採用したFORTRANコンパイラの制約上256KBの配列は確保できないので、座標データは外部ファイルとして処理した。

なお、等高線追跡のアルゴリズムは次に示すようなものである。(図3参照)

- ① 画面上のドットデータを x 軸に沿って左上から右下に順次走査して行く。
- ② 最初に出会った等高線上の点を始点として追跡を開始する。追跡方向は 3×3 ドットの近傍で、始点を中心にして反時計方向に連結点を探索する。最初に発見した点を連結点として、最初の点はマークして探索済みであることを記憶しておく。
- ③ 次に、この連結点を中心として②と同様に反時計方向に探索して次の連結点を捜す。見つければ、中心の点をマークする。
- ④ 3×3 ドットの近傍で連結点が見つからない場合は、 5×5 ドットの近傍まで拡大して探索する。これでマークされていない点が見つければ、この点は途切れ点として線をつなぎ、追跡を続行する。もし 5×5 ドットの近傍でも見つからなければ1本の等高線の追跡は終了する。

①から④の操作を繰り返して全ての等高線の追跡を行うが、このアルゴリズムの場合④の操作で明らかのように、等高線の途切れ点が5ドット以上離れている場合、

また、等高線間の距離が 5×5 ドットの近傍内まで接近している場合には不適である。前者の場合は、追跡を再開することは可能であり、また、後者の場合は等高線をスキヤナで読み取る場合にチェックすることができる。④の途切れ点における探索方法については、厳密にはベクトルによる方向を考慮して探索するのが望ましいが、ここでは簡単のため省略した。

2.4 等高線間の z 座標の計算

等高線データから3次元サーフィスマデルを再現するには、等高線の各点を頂点とした多角形を設定し、これらの多角形によって表面を構成するのが一般的な手法である。この手法としてよく用いられるのが、等高線の特徴点を抽出し、特徴点を頂点とする三角形パッチによって面を構成する手法である⁽⁴⁾。この手法はデータ量の圧縮という点では有利であるが、等高線の形状が複雑に変化しているような場合には、三角形パッチのはりめぐらし方に例外を設定したアルゴリズム⁽⁵⁾を必要とし、簡単にはいかないようである。これに対して本論文で採用した格子点モデルによる手法は、一定の間隔の格子を仮定し、格子と等高線の交点を基にしてその他の z 座標を計算し、これらの点によって形成される面を四角形のパッチと見なして3次元モデルを再生する手法であり、アルゴリズムも簡単である。しかし、一定間隔の格子をはりめぐらす場合、格子の間隔が広ければ等高線に忠実な形状の再生はできないので、格子点の間隔をある程度狭くする必要が生じる。従って、データの圧縮という点では不利であるが、ここではアルゴリズムの簡単さを優先させてこの格子点モデルを採用した。この格子点モデルの場合、等高線と格子の交点間の座標値を求めるアルゴリズムについては、既に程⁽⁶⁾らが提案しているが、ここでは

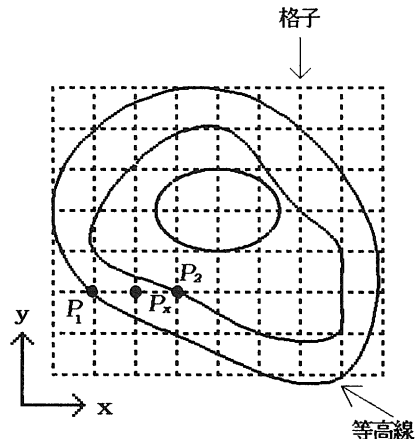


図4 等高線間の z 座標の決定

このアルゴリズムを参考にして、より簡単なスキャンライン型のアルゴリズムを提案する。(図4参照)

- ① x軸方向に左から右に操作してゆき、等高線との交点を探索する。
- ② 等高線との最初の交点 $P_1(x_1, y_1, z_1)$ を記憶し、次の交点 $P_2(x_2, y_2, z_2)$ のz座標の値と比較する。
- ③ z座標の値が異なれば、次式により、2点間の P_x のz座標の値 z_x を計算する。

$$z_x = \frac{z_2 - z_1}{x_2 - x_1} x_1 + z_1$$

- ④ x軸方向の走査が終了したら次にy軸方向に操作し、等高線間の未決定の点を探索し、その点からそれぞれy軸方向で反対方向の最短距離にあるz座標の既知の点を探す。これらの2つの点のz座標の値がお互いに異なっていれば、この2つのz座標の値により、前述の式を使って計算する。

以上のような操作を三沢⁽⁷⁾らの等高線モデルに適用すると、等高線の間隔が疎な部分、特に頂上にあるような平坦な部分(図5のaの部分)、或は半島上に突出した部分(図5のbの部分)では、前述のアルゴリズムではz

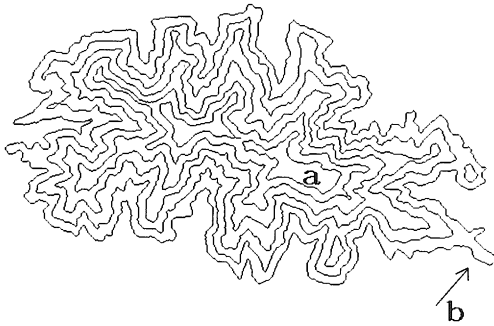


図5 等高線図(三沢⁽⁷⁾らによる)

座標の値が決定できない場合がある。これについては次の2.5で検討する。

2.5 等高線の間隔が疎な部分

2.4で述べたように、図5のa、または同図bのような部分ではこれまでの方法では等高線間の値が決定できない。このうちbの部分については、全て同一の高さとして処理することが考えられるが、まだ有効な対策は考えていない。本システムでは等高線の高さは1バイト整数として処理しているので、これを2バイト以上また

は実数として処理すれば値が決定できるかもしれないが、この場合はデータ量が增大するという問題がある。

これに対してaの部分については、x軸方向に走査してゆき、未決定点の直前の数点によってx軸方向への傾斜を計算する。これは図6に示すようにそれぞれ向かい

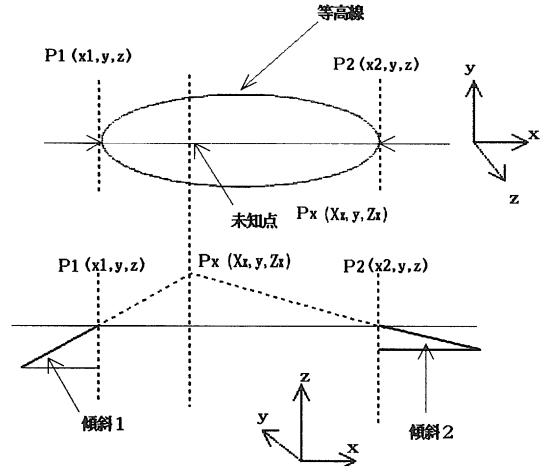


図6 頂上近辺の平坦な部分の高度決定

合った2点、 P_1 、 P_2 点において行う。この場合、 P_1 、 P_2 点の傾きが、それぞれ逆で、かつ P_1 、 P_2 の傾きから求めた直線の交点 P_x のx座標が P_1 、 P_2 の間にあれば P_x 点のz座標の値をその点の高さとする。ただし、この場合、 P_x 点のz座標と P_1 、 P_2 点のz座標の差は、等高線の偏移差以内でなければならない。これ以外の場合はいずれも P_1 (P_2)の高さと等しいと仮定した。

2.6 サーフィスマodelの再生

2.4によって等高線間の座標が決定できれば、これからサーフィスマodelを作成するための四角形パッチを作成しなければならない。この作業は、前述したように、三角形パッチを等高線に沿って作成するのと比較すると簡単である。四角形の大きさは、ここでは 3×3 ドットとした。

四角形パッチは等高線に沿ってはりめぐらすのではなく、x、y軸に沿って、格子状に作成する。従って、山岳部分の境界となる等高線の部分では、等高線の形状を完全に再現はできないが、 3×3 ドットのパッチなので実用的には十分と考えられる。また、この境界部分では、四角形の4個の頂点のうち、2点以上が山岳部分に属していれば、残りの点はすべて境界上の等高線のz座標値と仮定した。

隠面除去のための各面のz座標値は4頂点の平均値を

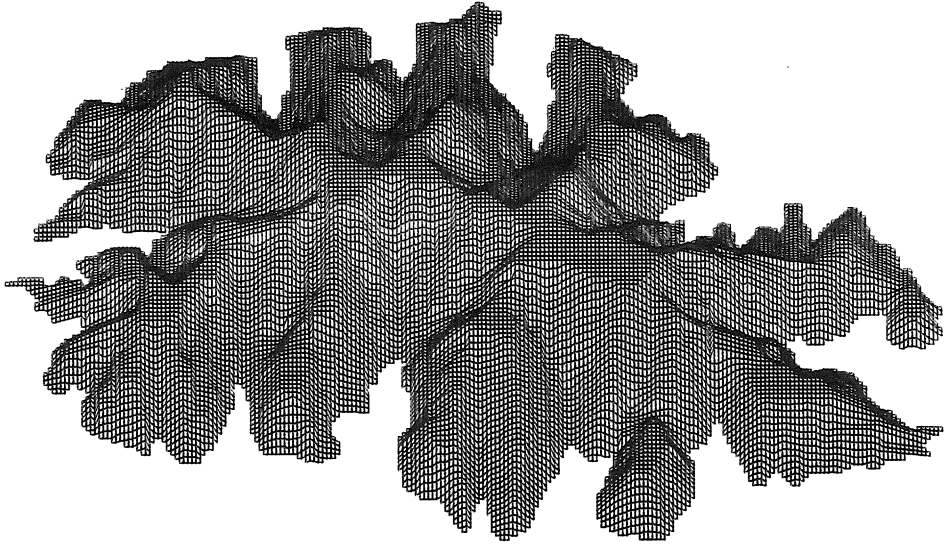


図 7(a) ワイヤーフレームモデル

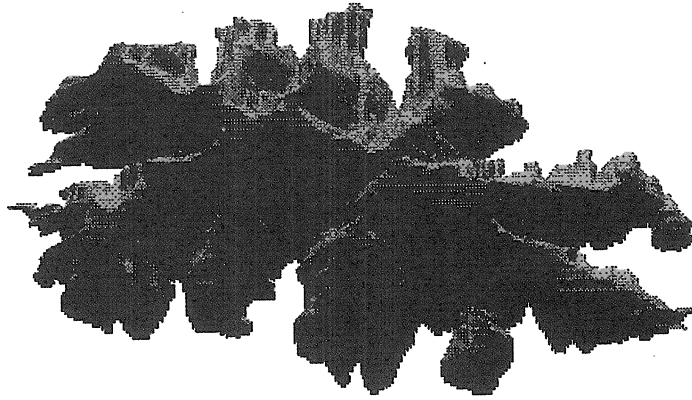


図 7(b) サーフィスマodel 1

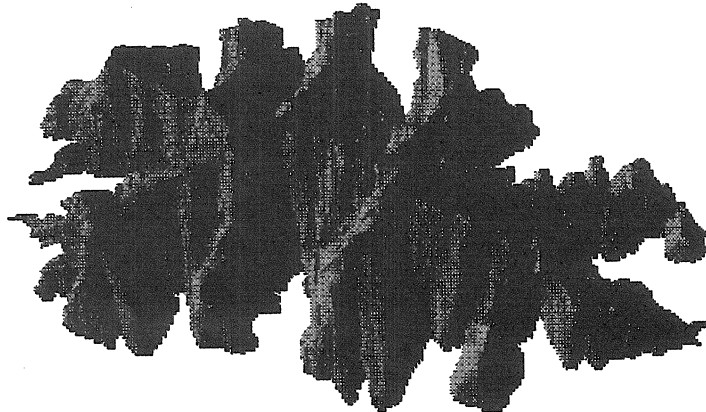


図 7(c) サーフィスマodel 2

使用し、シェイディングのために面の中心から、2つの頂点に向かうベクトルから法線ベクトルを求めて使用した。

システム作成のための言語ソフトウェアとして使用したのは、Lattice-CとPC-FORTRANで、面の塗りつぶしにはc-tool98のPolygon-fillルーチンを使用し、シェイディングの表現にはやはりc-tool98のRGB各16段階のタイリングパターンを使用した。

図7(a), (b), (c)は図5の等高線を上記の方法で再現したワイヤフレームとサーフィスマodelの図である。サーフィスマodelの光源はそれぞれy, x軸の正, 負の方向に存在する点光源で、視点はz軸の正の方向に存在する場合の透視変換による図形である。

3. む す び

イメージスキャナから読み込んだ等高線を追跡して、3次元サーフィスマodelをパソコンで再現する方法を提案した。等高線データの読取はデジタイザによる方法よりも簡単になり労力も少ない。また、読み取った等高線データから3次元サーフィスマodelの再生をするために、格子点モデルによる3×3ドットの四角形パッチによる方法を提案し、簡単に3次元モデルの再生ができることを示した。システムのインプリメントは16ビットパソコン上で行い、より実用的なシステムをめざしたので、現実的な教材作成システムとしての利用も可能と考える。現時点では、等高線の入力から、3次元モデルの再生までには、半日程度の時間がかかるが、64KB以上の配列が使用可能な言語ソフトウェアの利用、16ビットパソコンの高速化、あるいは32ビット化により、この時間は大幅に短縮されることは確実である。例えば、サーフィスマodelの視点を変えた再生にかかる時間は本システムでは15分程度であるが、本学部のHP-350SRXシステムでは2～3秒程度の時間で可能である。

今後の課題としては、3次元モデル再生のために必要なデータ量の圧縮と2.5で述べた等高線間の未決定点のz座標を計算する方法を考える必要がある。

文 献

- (1) T. AGUI, M. NAKAJIMA, and K. MATSUBARA: "An automatic generation algorithm of closed curves in contour map processing", Trans. IECE Japan, Vol. E-64, 9, pp.610-611 (1981).
- (2) C. J. Hilditch: "Linear skeletons from square cupboards", University Press, Machine Intelligence IV, pp.403-420 (1969).
- (3) 森, 板倉: "画像認識の基礎(1)—前処理と形の特徴抽出—", オーム社 (昭61).
- (4) 安居院, 宮田, 中嶋: "三次元山岳形状の等高線からの自動作成法", 電子通信学会論文誌(D), J 69-D, 12, pp.1905-1912 (昭61).
- (4) 安居院, 長, 中嶋: "等高線地図に忠実な山岳形状再生に関する研究", 情報処理学会研究会報告, グラフィクスとCAD, 27-7 (1987).
- (6) 程, 出沢, 相馬: "等高線形状からモデルから格子点モデルへの変換", 情報処理学会研究会報告, グラフィクスとCAD, 21-4 (1986).
- (7) 三沢, 千葉, 斉藤: "自然の画像生成—山の手続き的形状定義法—", 情報処理学会研究報告, グラフィクスとCAD, 21-6 (1986).