

チェーン符号法による等高線のデータ圧縮

○ 日本大学理工学部 正員 鈴木芳朗
筑波大学電子・情報工学系 正員 星 仰

1. データ圧縮の構想

地図情報システムの中で大容量となる等高線データは、コンピュータ処理に対して保管スペースを占有するため、ハードウェア全体のシステム構成まで影響してくる。したがって等高線データのコンパクト化は、地図情報を取り扱う上で重要な事項と考えられる。最近、建設省国土地理院が提言しているデジタルマッピングのデータベース化でのコンパクト化は、データの間引き、もしくはデータ間隔をどの程度に設定すれば良いのかといった視点で研究されていて、本来の電算機処理時に重要なデータ圧縮手法に付いては触れられていない。そこで本研究では、何等かの基準によって与えられた既知の等高線デジタルデータを圧縮する方法について論述し、実験によって手法の効果を示す。

ファイルされた莫大な等高線データを、コンピュータ処理によって自動的に描画システムで出力表示していく場合に、全データ点上を通過する曲線補間では、処理時間が必要以上にかかる上、処理の工程からみて適切な方法と考えられない。各座標データのうち、同一方向ベクトルが多数配列している等高線を見ると、それらを全点通過して描いた等高線と、同一方向ベクトルをもつ連続したデータ群の内から最初の点と最終のデータ点間を補間させるようにし、その間にある全ての同一方向ベクトルを有する点を補間の対象から外して得る等高線の形状とは、ほぼ同一形をした線状になるものとする。この様にデータ圧縮から等高線の表現が適切に行えるならば、補間処理時間が短縮されるので、コンピュータ処理上から有効な利用法といえよう。ここで補間に必要としない多くの等高線データが存在することになるが、それらのデータはすべて廃棄されて消滅するのではなく、原データとして一次ファイル上に集録されている未利用データである。それらのデータは次の要求する図化縮尺の変更の際に利用することも考えられ、また、等高線上の単独の点データとして意味を持っており、別の用途に利用することも考えられる。データ圧縮法には種々の手法が考えられるが、本研究では単純なベクトル化によるチェーンコード法を用いることにする。等高線データのチェーンコード化によるデータ列は前処理部分であり、一種のデータ変換といえる。チェーンコードを圧縮するには可逆性を有する処理を念頭におく。この方法には、連続コードの圧縮法と、差分法による圧縮法などが考えられるが、ここでは前者の方法による圧縮を試みることにする。

データ処理には大容量のデータを取り扱う関係から、一般的には大型コンピュータを用いるのが、利用面から便利であるが、本研究ではハードウェアが低コストであり、利用面での汎用性を考慮してパーソナルコンピュータを用いた処理過程を取り入れることにする。したがって処理用のソフトウェアはベシック言語で作成する。

2. チェーン符号化

ファイルされている等高線データ群の内から、画像や線画処理の表示を行う場合、原データから同類データの選択を行う必要上、各データについて同一方向ベクトルを求めて各データの持つ方向性から種別する符号を付けて表すことができる。この符号(コード)を連続している各点データが持つことになるので各データにおけるチェーン符号化ということができる。このように一本の等高線には写真測量的手法によって抽出された座標データが連続してファイルされている。それらの各データに各点から次の点への方向角による認識を与えてデータに符号を付けて表すと、各点は連続しているのでチェーンを構成することになる。

このチェーンとして符号化された各点において、同類の符号が連続している場合には、その同類系の中から始点データと終点データを用いた補間処理を行っても全点を通る補間処理とほぼ同様の形状が得られる。このような処理方法がチェーン符号化によるデータ圧縮の基本的な考え方である。また、この手法では処理後に未使用となる各点データについては原データとして1次ファイルに保存されているので、他の目的に利用することは可能であり、いつでも原形データが再現できる。

以下に等高線データのチェーン符号化によるデータ圧縮の処理方法について詳述する。

2-1 データ圧縮の前処理 (符号化)

等高線上の任意間隔で採取された座標データ群において、始点から進行方向の次の点との座標差より始点の等高線に沿った方向角と距離を求めることができる。したがってこの処理では等高線上の連続する各点について同様に次の進行方向の点に対しての方向ベクトルとして求められる。ここで、各点における方向ベクトルから符号化を行うのであるが、そのためには方向ベクトルと符号化の基準を設定しなければならない。一般に符号化については、図-1に示すような方向ベクトルから分割する量子数に応じた方法があり、最も粗い8分割に始まり、16分割、32分割、のように細分化してゆく分割法が考えられる。

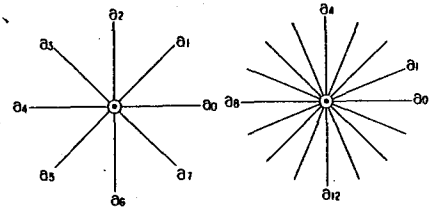


図-1

8方向の分割による処理について符号化を考えると、求められた各点の方向ベクトルから、コード a_0 より a_7 までの8量子の符号化ができる。この場合の符号化の範囲については次の様に表わすことができる。単位ベクトルを a とし、各方向ベクトル a_i ($i=0\sim7$) で表すことにすれば、各 a_i の範囲は次式のようなになる。

$$\frac{\pi}{8} (2i+1) - \frac{\pi}{4} \leq a_i \leq \frac{\pi}{8} (2i+1) \dots (1)$$

($i=0\sim7$)

したがって、方向分割を細かくすると量子が増加することになり、方向ベクトル a_i は、 n 量子に対し次式のようなになる。

$$\frac{\pi}{n} (2i+1) - \frac{2\pi}{n} \leq a_i \leq \frac{\pi}{n} (2i+1) \dots (2)$$

ここで、($i=0\sim n-1$), ($n=i+1$)

n 量子については、コードを2桁以内と考えれば、 $n_1=8$, $n_2=16$, $n_3=32$, $n_4=64$, の4種類となる。一般には、 $n_j=2^m$ ($m=j+2$), ($j=1,2,\dots$) となる。

この方向分割の処理では、式で表される範囲内については同一記号となるので、圧縮を行う場合、点ベクトル誤差を生ずることになる。予測される誤差としては、図-2に示す通り点1から点2に至るベクトルは符号化によってAからBまでの範囲で誤差が見込まれることになる。図-2より誤差は点1から点2までのベクトル L に比例することになり、コード角 $2\pi/n_j$ の $1/2$ 以内とすれば次式の範囲内が考えられる。

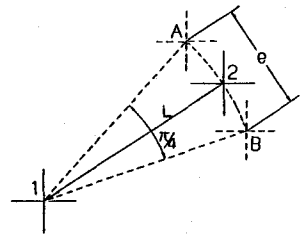


図-2

$$e < L \cdot \frac{\pi}{n_j} \quad (j=1,2,\dots) \dots (3)$$

したがって、量子 n が小さな値であると等高線の変化の著しい個所では点ベクトルは大きく変形する可

能性があり、本来の等高線と異なる形状となることが考えられる。

2-2 チェーン符号法によるデータ圧縮

方向ベクトルによる点データの符号化は図-1に示されているように、ある点から次の点へ方向に対して定められている角度の分割量子によって設定される。分割が8量子の場合には、各点を表すコードはa0からa7までとなるので、1桁の符号で対応できる。しかし、16量子や32量子になると2桁のコードを必要とすることになる。

図-3に示されている結合する等高線の場合を考えると、等高線上で採取された座標データはa1からa20までの20個のデータがある。その各点のデータについて次の点に対する方向ベクトル、表-1に整理されている値となっており、それらに対する各量子の違いによるコードが示されている。表から符号化は各方向を分割する量子数によって異なった領域に区分されることが理解できる。

これらの符号化された点データをその領域における同類値とみなして圧縮処理を次の様な形式で行うことができる。

例として、8量子分割の符号を用いてデータ圧縮を行ってみると表-2に示すような形体に圧縮処理することができる。

表-1 データの量子化

点 名	方向ベクトル (π)	8量子 (i=0~7)	16量子 (i=0~15)	32量子 (i=0~31)
1	0.523	1	1	3
2	0.785	1	2	4
3	0.698	1	2	4
4	0.523	1	1	3
5	0.349	0	1	2
6	0.174	0	0	1
7	0	0	0	0
8	5.938	0	15	30
9	5.502	7	14	29
10	5.236	7	13	28
11	4.886	7	12	26
12	4.363	6	11	22
13	3.491	6	9	18
14	3.317	4	8	17
15	3.317	4	8	17
16	3.142	4	8	16
17	2.967	4	8	15
18	2.793	4	7	14
19	2.443	3	6	12
20	1.047	1	3	5

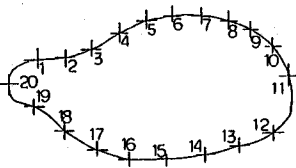


図-3 等高線と点データ

表一2 8量子分割によるデータ圧縮

点 名	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
8量子 a _i	1	1	1	1	0	0	0	0	7	7	7	6	6	4	4	4	4	4	3	1
圧 縮 コード	1 9 4				0 9 4				7	7	7	6	6	4 9 5				3 1		

点名1から4までは、a_iの同一符号が連続している。それを圧縮すると下段のように194として示すことができる。このコードでの1はa_iを表し、9は同一符号が連続していることを示す。また、最後の4は個数を表し、連続しているa_iが4個という意味である。この様にして圧縮すると下段に示しているような形となって、方向ベクトルから求められたチェーン符号を圧縮することができる。しかし、圧縮3コードを用いるので、連続する同一符号は4個以上が対象となる。

2-3 チェーン符号におけるデータの補正

チェーン符号を用いたデータ圧縮では、連続する等高線のデジタルデータより、各点において次の点に対する角度と距離によって逐次符号化されてゆくの、量子によって角度が異なってくる。そのため連続する2点以降の位置が変化することにより誤差が蓄積された形として現れてくることになる。そのことを考慮して次のような補正を試みることにする。

ある量子分割によりチェーン符号化されたデータに基づいて作図処理を行うケースとして図-4に示す連続したA, B, C, Dの線分を仮定する。各線分A-B, B-C, C-Dの量子がそれぞれ0, 1, 2であるとすれば、この粒子に対するベクトルA-B₁, B₁-C₁, C₁-D₁を用いて連続線分A-B₁-C₁-D₁の処理をすることができる。この様な処理では、図から明らかなように、B₁, C₁, D₁・・・と誤差が蓄積されることになる。このB-B₁間の誤差は、チェーン化に伴うものであるから、個々の調整は、量子数の増減及び最大移動量の増減により行うことができる。これに対し、C-C₁間とD-D₁間の誤差は前の許容誤差の外に誤差を増幅する要因が認められることになる。つまり、C-C₁間の誤差に関しては、B≠B₁であり、同様にD-D₁間の誤差についてもC≠C₁でないことがその要因である。その対策として、Cの方向ベクトルを決定する基準をBからB₁に移動している点B₁を基準としてCの量子を決定するとうな補正法を行うことにする。この様にしてチェーン符号化を行えば、図中のB₁-C₂, C₂-D₂の作図処理となる。補正の基本的方法としては、図-4より、点A, B, Cの各座標を次のように表す。

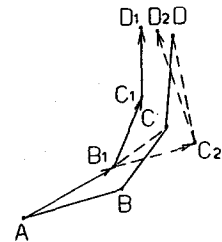


図-4

ある量子分割によりチェーン符号化されたデータに基づいて作図処理を行うケースとして図-4に示す連続したA, B, C, Dの線分を仮定する。各線分A-B, B-C, C-Dの量子がそれぞれ0, 1, 2であるとすれば、この粒子に対するベクトルA-B₁, B₁-C₁, C₁-D₁を用いて連続線分A-B₁-C₁-D₁の処理をすることができる。この様な処理では、図から明らかなように、B₁, C₁, D₁・・・と誤差が蓄積されることになる。このB-B₁間の誤差は、チェーン化に伴うものであるから、個々の調整は、量子数の増減及び最大移動量の増減により行うことができる。これに対し、C-C₁間とD-D₁間の誤差は前の許容誤差の外に誤差を増幅する要因が認められることになる。つまり、C-C₁間の誤差に関しては、B≠B₁であり、同様にD-D₁間の誤差についてもC≠C₁でないことがその要因である。その対策として、Cの方向ベクトルを決定する基準をBからB₁に移動している点B₁を基準としてCの量子を決定するとうな補正法を行うことにする。この様にしてチェーン符号化を行えば、図中のB₁-C₂, C₂-D₂の作図処理となる。補正の基本的方法としては、図-4より、点A, B, Cの各座標を次のように表す。

$$A (x_A, y_A) \quad B (x_B, y_B) \quad B_1 (x_{B1}, y_{B1}) \quad C (x_C, y_C)$$

点BのAに対する距離L_Bは

$$L_B = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \quad \dots \dots \dots (4)$$

x軸の正の方向に対する線分ABの角度θ_Bとすれば、

$$\tan \theta_B = \frac{(y_B - y_A)}{(x_B - x_A)} \quad \dots \dots \dots (5)$$

この値をある分割数に基づいて量子化した値を a_B とすると、点 $B_1 (x_{B1}, y_{B1})$ は、

$$\begin{aligned} x_{B1} &= x_A + L_B \cdot \cos (\arctan a_B) \\ y_{B1} &= y_A + L_B \cdot \sin (\arctan a_B) \end{aligned} \quad \dots \dots \dots (6)$$

この $B_1 (x_{B1}, y_{B1})$ は、点 $B (x_B, y_B)$ の補正点である。

点 C については、点 B_1 を基準に距離、角度、補正点 C_2 の座標を求めることができる。

一般式として点 M の補正点 M' を基準とした次の点 N については、式 (7)、(8) のようにまとめられる。

$$L_N = \sqrt{(x_N - x_{M'})^2 + (y_N - y_{M'})^2} \quad \dots \dots \dots (7)$$

$$\tan \theta_C = \frac{(y_N - y_{M'})}{(x_N - x_{M'})} \quad \dots \dots \dots (8)$$

式 (8) の結果量子 a_N とすると、また、上式の x_N, y_N は式 (9) で書き表される。

$$\begin{aligned} x_N &= x_{M'} + L_N \cdot \cos (\arctan a_N) \\ y_N &= y_{M'} + L_N \cdot \sin (\arctan a_N) \end{aligned} \quad \dots \dots \dots (9)$$

3、原データの抽出

等高線のデジタルデータは、写真測量用の実体図化機に取りつけられているデジタル座標記録装置によって抽出されるデータを用いる。この座標記録装置はパーソナルコンピュータ PC-9801 とインタフェースによってオンライン化されているので、等高線データは直接 PC-9801 に入力してファイルすることができる。座標記録装置の抽出モードには自動連続抽出モードとマニュアル抽出モードとがあり、前者には一定時間の間隔で等高線上の座標データを抽出する方法と XY の座標採取間隔を任意に設定して等高線上のデータを抽出する方法とがある。一定時間による抽出は曲率半径が小さく凹凸のはげしい地形に適用することができる。一方マニュアル抽出モードは図化技術者が必要と思われる特徴ある等高線上の座標モードを不定間隔で抽出するもので、一般の地形に適用される。

本研究では後者のマニュアル抽出方法によるデータを実験に用いる。実験用のモデル地域として伊豆大島を選定し、サンプリング縮尺 1/25,000 で 50m インターバルの等高線デジタルデータを抽出して PC-9801 のフロッピーディスクに 1 次ファイルを行った。

4、データファイルのフォーマット

データのファイルとしては、実体図化機 (A-10) より標高データ (x, y, z) を抽出したデータは絶対座標となっている。このままではデータ量が多いので、相対座標にデータ変換することにより、多少のデータのコンパクト化が望めるものと期待できる。ここでは前者の絶対座標のデータファイルを第 1 次ファイル、後者の相対座標のデータファイルを第 2 次ファイルとする。この第 2 次ファイルからチェーン符号化データ (第 3 次ファイル) を作成する。これらの 3 種のデータファイルのフォーマットについて以下詳述する。

4-1 絶対座標ファイルのフォーマット

絶対座標ファイルはレコード長固定に順編成されたファイルであり、記述はすべて文字型となっている。ファイル形式は図-5 に示す通り、座標読み取り装置のコードから、8 桁のマニュアル指示コード、6 桁

のX座標、そして6桁のY座標であり、その各々をダブルクォーツ(”)で囲んだ上、キャリッジリターン(CR)とラインフィード(LF)を付けた構成となっている。したがって1セットのレコード長は32バイトの使用となる。

4-2 相対座標ファイルのフォーマット

この相対座標ファイルは、レコード長が可変の順編成ファイルであり、ファイルヘッダにはファイル番号、縮尺、最大増減幅、サイズ記号や用紙サイズが記録されている。図-6に示すとおり、各レコードは1桁目に等高線が開曲線か閉曲線についての識別する文字としてOまたはCが記録され、次に等高線番号が整数型2バイトで、そして最終桁にレコードの終了を示す文字Lが記録される。1本の等高線の終了の場合にはLはEで記録されるようになる。各レコードのデータ部分は、単精度型でX方向の相対座標値、Y方向の相対座標値の順にデータが配列されている。ただし、第一のデータ(図ではD1)に限って数値は絶対座標値となっている。

4-3 チェーン符号化ファイルのフォーマット

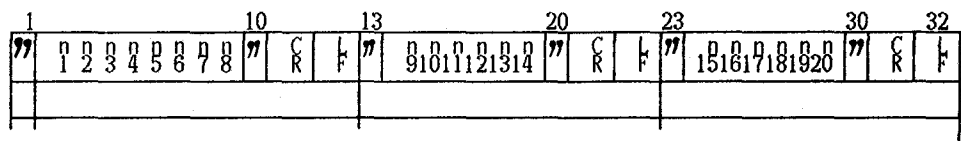
チェーン符号化ファイルは、レコード長が可変の順編成ファイルとなっている。ファイルヘッダは相対座標値ファイルのファイルヘッダに量子数の情報を追加したものであり、各レコードは相対座標値ファイルのフォーマットと同一となっている。ただし、データはX方向の相対座標値、Y方向の相対座標値の代わりに量子番号と前の点からの距離が記録されるようになっている。そのことは、データにおいて同一の量子番号が連続する場合において、量子の連続数と番号によって、さらにデータを圧縮するためであり、そのために前の点からの距離を一定にとらなければならなくなる。現在、創成されている原データファイルは距離をランダムにとったデータであるためにチェーン符号化ファイルは前の点からの距離という情報をもたざるを得ず、その結果としてチェーン符号化ファイルでは相対座標ファイルとほぼ同じ規模となることになる。

5、データの圧縮率

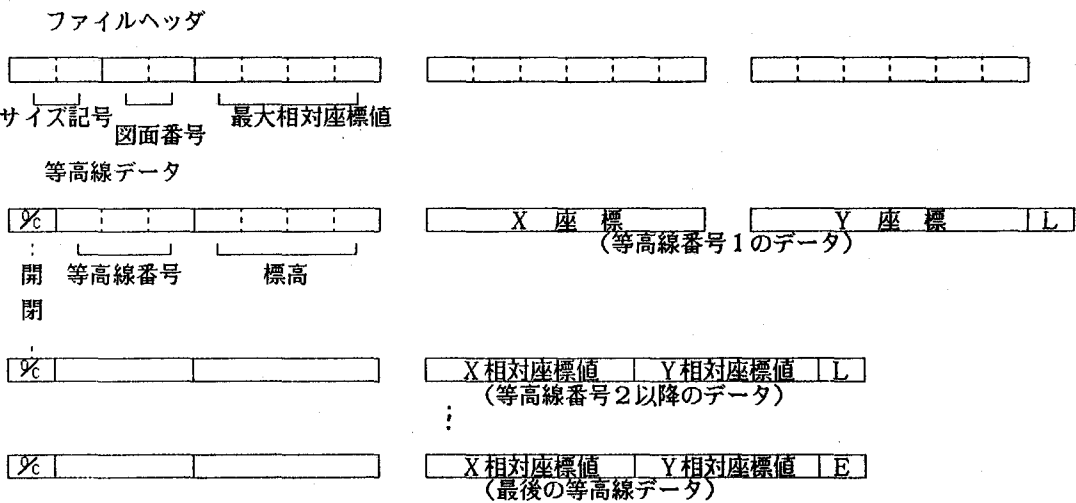
原データのファイルフォーマットが1セット32バイトの形式で構成されているならば、仮にサンプルデータファイルのレコード数が416の場合を想定するとファイル規模は32バイト×416レコードで13,312バイトとなる。これを図-7に示すようなファイルフォーマットに変換するとデータの取扱がやりやすくなると考えられる。この場合では、1セット24バイトであるからサンプルファイルをこの形式に変換するとファイル規模は24バイト×416レコードであり9,984バイトになる。したがって、この形式からでもデータ変換によってデータのコンパクト化ができる。

まず、増減値ファイルの規模については、ファイルの基本情報をヘッダーとして等高線1本について図-6に示すようなフォーマットになっている。この座標データは座標の採取数によって不定であるからレコードの長さは一定にはならない。先のサンプルファイル416の場合では数値をすべて単精度実数として取り扱っているために4バイトを占有することになり、この増減値ファイルの規模は3,767バイトとなってしまふ。しかし、ここでデータを1量子単位に変換することによって最終的に取り扱うデータとしては、分割数を8量子の場合でみるとサンプルデータは圧縮数が135となるので、量子ファイルは279に減少することになる。ここでデータ長は量子数を3桁とった場合、距離に要する3桁と区切り符号をあわせると7バイトとなるから、全体では、1978バイトに圧縮されることになる。サンプルファイルのデータを1セット24バイトの規模9,984バイトからみた場合、8量子では圧縮率は約1.9

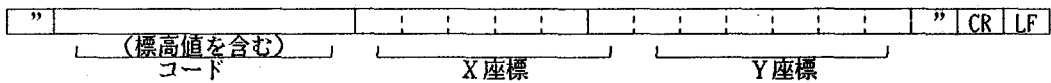
に縮小されたことになり、16量子場合には、1.68の圧縮率となる。この圧縮率については更に率を高めることは可能であるが実際には地形形状によって変化する等高線の表現から圧縮率としては形状に応じて変化した範囲をとるものと考えられる。



図ー5 絶対座標ファイル第1次ファイルフォーマット



図ー6 相対座標値ファイル第2次ファイルのフォーマット



図ー7 24バイトのファイル規模 (パソコン用第2次ファイル)