

標本点で表された曲線の曲率による補間について

小野直樹 瀧山龍三

九州芸術工科大学

〒815 福岡市南区塩原4-9-1

あらまし

曲線の認識やデータの保存のためには、形状を損なわない程度にできるだけ少ない標本点によって曲線を記述することが望ましい。一方、そのような曲線を表す離散点の系列から逆に曲線を再現することがしばしば必要となる。通常は、簡単のために隣合う標本点間を直線で結ぶことによって曲線を再現することが多いが、それでは滑らかな曲線は得られない。又、曲線補間の代表的な方法としてスプライン関数によるものがあるが、各標本点間ごとに与える多項式関数の係数を求めるために方程式を解く必要があり計算のための負担が大きい。本稿では、標本点によって記述された曲線を曲率を用いて滑らかに補間する方法を提案する。この方法は、開曲線閉曲線ともに用いることができ、スプライン補間とほぼ同程度に滑らかな曲線を、より少ないメモリと処理時間とで生成することができる。

和文キーワード 標本点、補間、曲率、スプライン補間、曲線の生成

Interpolation of sampled curves by using curvature function

Naoki ONO and Ryuzo TAKIYAMA

Kyushu Institute of Design

4-9-1 Shiobaru, Minami-ku, Fukuoka, 815 Japan

Abstract

It is often required to reconstruct curves from sparse sample points. The interpolation by connecting adjacent points with straight lines is very simple, but it suffers from the lack of smoothness of resultant curves. Although the spline function is frequently used for smooth interpolation, it takes a longer computation time and much memory space.

We propose a method to interpolate dotted curves smoothly by using curvatures. This method is applicable to both open and closed curves, and needs less memory space and computation time than those by the spline interpolation.

英文 key words interpolation, sample points, curvature, Spline interpolation, curve reconstruction

1. まえがき

画像処理の重要な対象の一つに曲線があげられる。曲線の中でも特に図形の輪郭線には、対象の認識をする上で有効な情報が多くふくまれているためしばしば取り上げられる。そのような対象の同定及び認識処理には、基本となる既知のデータの蓄積、すなわちデータベースが必要となる。その際、認識に有効な形式のデータであることは勿論、保存に要するメモリが少なくすむようにデータの大きさが小さいことが望ましい。従って、輪郭線及び曲線データを保存する場合には、曲線を表しているすべての画素に関するデータを記述するのではなく、元々の形を損なわない程度に標本点を選び出して、少ない標本点で近似すると都合がよい^[1]。認識処理などもそれら選び出された標本点を用いるので効率的に行える。

さて、そのような標本点データから元の輪郭線を再現するときには、一般には簡単のために直線で補間することが多い。ところが、直線を用いると標本点において角が生じ、再生される輪郭線は多角形となるため視覚的に不自然なものになり易い。そこで、標本点間を滑らかに補間して輪郭線を再生することが望まれる。

標本点間を滑らかに補間する方法として、周波数空間による方法が考えられる。まず、標本点データ系列を座標そのものや角度などを用いて何らかの関数で表す。それを離散フーリエ変換してデータ点数と同じ数の周波数スペクトル成分を得る。こうして得られた周波数成分よりも高域の周波数成分を0とおいてデータ点数を増やし、それを逆変換することによって標本化密度を上げることができる。例えば、入力したデータ点の数と同じ数の0を高域に付加すれば、標本化密度は2倍になる。この方法によると本来存在していたものより高い周波成分は生じないで標本化密度を上げることができると考えられる。

実際、曲線を周波数空間において記述したものはフーリエ記述子と呼ばれ、その代表的なものとしてG型記述子、P型記述子がある。G型記述子は標本点座標を複素平面上で表したものをフーリエ変換したものである^[2]。これを開曲線に対して適用し、上の手順に従って補間すると閉曲線が生成されてしまう。一方、P型記述子は、曲線の各点における偏角を指数部に持つ関数をフーリエ変換したものである^[3]。P型記述子は、周波数0の成分を操作しない限り曲線の端点が保存されるという特長を有するが、上の方法によって補間すると、得られる曲線はデータとして与えられている標本点を必ずしも通らない。また、曲線の曲がり具合を

記述した関数である偏角関数や曲率関数をフーリエ変換したものも考えられる^[4]。しかしながら、それらを用いて得られる補間曲線はG型記述子と同様に端点の位置が保存されないため、曲線の形状が代わってしまう。従って、曲線を表す標本点を滑らかに補間するためには用いることができない。

一方、画像空間上で標本点を補間するための方法もいくつか提案されている。一般に、最もよく使われる曲線補間のための方法として3次スプラインによるものがある^{[5][6]}。これは、曲線を各標本点間ごとに3次関数によって補間するものである。各区間の接続点において、1次及び2次微分が等しくなるように補間に用いる3次関数の係数を各区間ごとに求めていく。この方法によって、全ての標本点を通りかつ曲率の2乗和が最小な曲線を得ることができる。しかしながら、この方法を用いると、3次関数の係数を求めるために例えばN個の標本点が与えられたときにはN元の連立方程式を解く必要がある。標本点が多いときこれを解くために要するメモリと時間とは非常に大きなものとなる。

本研究では、曲線を表す標本点間を画像空間上で滑らかに補間する方法を提案している。ここで提案する方法は、各標本点における曲率に基づいて標本点間を補間するものである。但し、標本点間を連続関数で近似するのではなく、周波数空間処理による補間の場合と同じく、図1に示すように補間によって標本点密度を上げるものである^[7]。

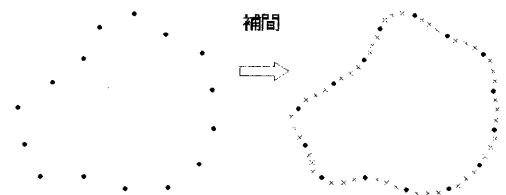


図1 標本点と補間点

2. 曲率の2乗和の最小化による補間

2.1 補間点の決定法^[8]

データとして、曲線に沿ってN個の標本点 $p_i (i=1, 2, \dots, N)$ の座標が与えられているものとする。なお開曲線の場合にはN+1個の標本点が与えられているものとする。ここで提案する方法は、曲率を用いて隣接した標本点間に新たに1つずつ補間点を定めていくものである。

図2に示すように標本点 p_i, p_{i+1} における曲率をそれぞれ c_i, c_{i+1} とする。また、 p_i と p_{i+1} との距離 $p_i p_{i+1}$ を d_i とする。このとき補間点 P_i を、 p_i からも p_{i+1} からも等しい距離にあって、 P_i における曲率 C_i が

$$C_i = ac_i + bc_{i+1} \quad (1)$$

となるように与えることにする。すなわち、点 P_i は、 $p_i P_i = p_{i+1} P_i$ でかつ曲率が C_i となる点である。

一般に、曲線上の各点における曲率の大きさが小さいときにはその曲線は滑らかである。従って、曲率ができるだけ小さくなるように補間点を選べば、滑らかに補間された曲線が得られる。そこで、パラメータ a, b を補間後の曲率の2乗和が最小になるように定める。すなわち、

$$J = \sum_{i=1}^N C_i^2 + \sum_{i=1}^N c_i'^2 \quad (2)$$

を最小にするように決める。ここで、 c_i' は点 p_i における補間後の曲率であって

$$c_i' = c_i - \frac{a}{2}(c_i + c_{i-1}) - \frac{b}{2}(c_i + c_{i+1}) \quad (3)$$

と記述できる。従って、式(2)は

$$J = \sum_{i=1}^N (ac_i + bc_{i+1})^2 + \sum_{i=1}^N \left\{ c_i - \frac{a}{2}(c_i + c_{i-1}) - \frac{b}{2}(c_i + c_{i+1}) \right\}^2 \quad (4)$$

となる。

さて、 J の値が最小になるのは、

$$\frac{\partial J}{\partial a} = 0, \quad \frac{\partial J}{\partial b} = 0 \quad (5)$$

の時である。ここで、 J の a, b それぞれに関する偏微分をとると次のようになる。

$$\begin{aligned} \frac{\partial J}{\partial a} &= 2 \sum_{i=1}^N c_i (ac_i + bc_{i+1}) - \sum_{i=1}^N (c_i + c_{i-1}) \left\{ c_i - \frac{a}{2}(c_i + c_{i-1}) - \frac{b}{2}(c_i + c_{i+1}) \right\} \\ &= a \left\{ 2 \sum_{i=1}^N c_i^2 + \frac{1}{2} \sum_{i=1}^N (c_i + c_{i-1})^2 \right\} \\ &\quad + b \left\{ 2 \sum_{i=1}^N c_i c_{i+1} + \frac{1}{2} \sum_{i=1}^N (c_i + c_{i-1})(c_i + c_{i+1}) \right\} \\ &\quad - \sum_{i=1}^N c_i (c_i + c_{i-1}) \\ \frac{\partial J}{\partial b} &= 2 \sum_{i=1}^N c_{i+1} (ac_i + bc_{i+1}) - \sum_{i=1}^N (c_i + c_{i+1}) \left\{ c_i - \frac{a}{2}(c_i + c_{i-1}) - \frac{b}{2}(c_i + c_{i+1}) \right\} \\ &= a \left\{ 2 \sum_{i=1}^N c_i c_{i+1} + \frac{1}{2} \sum_{i=1}^N (c_i + c_{i-1})(c_i + c_{i+1}) \right\} \\ &\quad + b \left\{ 2 \sum_{i=1}^N c_{i+1}^2 + \frac{1}{2} \sum_{i=1}^N (c_i + c_{i+1})^2 \right\} \\ &\quad - \sum_{i=1}^N c_i (c_i + c_{i+1}) \end{aligned}$$

従って、次の連立方程式が導かれる。

$$\begin{aligned} d_1 a + d_2 b &= e_1 \\ d_2 a + d_3 b &= e_2 \end{aligned} \quad (6)$$

但し、

$$\begin{aligned} d_1 &= 2 \sum_{i=1}^N c_i^2 + \frac{1}{2} \sum_{i=1}^N (c_i + c_{i-1})^2 \\ d_2 &= 2 \sum_{i=1}^N c_i c_{i+1} + \frac{1}{2} \sum_{i=1}^N (c_i + c_{i-1})(c_i + c_{i+1}) \\ d_3 &= 2 \sum_{i=1}^N c_{i+1}^2 + \frac{1}{2} \sum_{i=1}^N (c_i + c_{i+1})^2 \\ e_1 &= \sum_{i=1}^N c_i (c_i + c_{i-1}) \\ e_2 &= \sum_{i=1}^N c_{i+1} (c_i + c_{i+1}) \end{aligned} \quad (7)$$

とおいた。

これを解くことによって a, b は次のように求められる。

$$\begin{aligned} a &= \frac{d_3 e_1 - d_2 e_2}{d_1 d_3 - d_2^2} \\ b &= \frac{d_1 e_2 - d_2 e_1}{d_1 d_3 - d_2^2} \end{aligned} \quad (8)$$

また、距離 $p_i P_i$ 及び $p_{i+1} P_i$ の値を r_i とすると r_i は、 P_i における曲率 C_i と標本点間距離 d_i とを用いて

$$r_i = \frac{d_i}{2 \cos\left(\frac{C_i}{2}\right)} \quad (9)$$

によって求められる。

ここでは、どの区間についても同じ曲率の重み a, b を用いているが、各区間ごとに係数 a, b を変えて最適化を行えば、曲率の2乗和はより小さくなり、従って得られる曲線はより滑らかになると考えられる。しかし、そのためには3次スプライン関数を用いるときと同じく N 元の連立方程式を解かなければならなくなるため、処理に要する時間、メモリとも大きなものとなってしまふ。

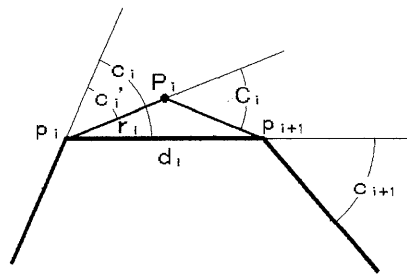


図2 曲率による補間点 P_i の決定

2. 2 閉曲線の場合のパラメータ a, b

データ点が閉曲線を表している場合には、データが周期的なので次の4つの関係式が成り立つ。

$$\begin{aligned} \sum_{i=1}^N c_i^2 &= \sum_{i=1}^N c_{i-1}^2 = \sum_{i=1}^N c_{i+1}^2 \\ \sum_{i=1}^N c_i c_{i+1} &= \sum_{i=1}^N c_i c_{i-1} \\ \sum_{i=1}^N (c_i + c_{i+1}) &= \sum_{i=1}^N (c_i + c_{i-1}) \\ \sum_{i=1}^N (c_i + c_{i+1})^2 &= \sum_{i=1}^N (c_i + c_{i-1})^2 \end{aligned}$$

従って、

$$\begin{aligned} d_1 = d_3 &= 2 \sum_{i=1}^N c_i^2 + \frac{1}{2} \sum_{i=1}^N (c_i + c_{i-1})^2 \\ e_1 = e_2 &= \sum_{i=1}^N c_i^2 + \sum_{i=1}^N c_i c_{i+1} \end{aligned}$$

となる。これを式(8)に代入すると

$$a = b = \frac{(d_1 - d_2)e_1}{d_1^2 - d_2^2} = \frac{e_1}{d_1 + d_2}$$

が得られる。分母の値 $d_1 + d_2$ は

$$\begin{aligned} d_1 + d_2 &= 4 \sum_{i=1}^N c_i^2 + \sum_{i=1}^N (c_i + c_{i+1})^2 \\ &\quad + 4 \sum_{i=1}^N c_i c_{i-1} + \sum_{i=1}^N (c_i + c_{i-1})(c_i + c_{i+1}) \\ &= 7 \sum_{i=1}^N c_i^2 + 8 \sum_{i=1}^N c_i c_{i+1} + \sum_{i=1}^N c_{i-1} c_{i+1} \end{aligned}$$

とまとめられるので、式(7),(8)は次のように簡単な形式に整理できる。

$$a = b = \frac{2 \left(\sum_{i=1}^N c_i^2 + \sum_{i=1}^N c_i c_{i+1} \right)}{7 \sum_{i=1}^N c_i^2 + 8 \sum_{i=1}^N c_i c_{i+1} + \sum_{i=1}^N c_{i-1} c_{i+1}} \quad (10)$$

従って、新たに与える補間点における曲率 C_1 は、閉曲線の場合には式(10)によって求められる a を用いて

$$C_1 = a(c_i + c_{i+1}) \quad (11)$$

によって計算される。なお、標本点から補間点までの距離は式(9)によって求められる。

更に特殊な場合として、曲率が全て等しい場合を考えてみる。このとき式(10)は

$$\begin{aligned} a = b &= \frac{4 \sum_{i=1}^N c_i^2}{16 \sum_{i=1}^N c_i^2} \\ &= 0.25 \end{aligned} \quad (12)$$

となる。すなわち、補間点における曲率は

$$C_1 = c_i / 2 \quad (13)$$

によって決められる。従って、曲率の変化が比較的小さい場合には、精度は多少低下するものの、曲率の重みパラメータ a, b を式(8)や式(10)を計算することなく、式(13)によって直接補間点の曲率が計算できるので、より簡便に補間することができる。

2. 3 補間の繰り返し

提案している方法では、 N 個の入力データに対して処理後の標本点の数が2N個になり標本点密度がほぼ2倍になるが、スプライン関数を用いたときのように、標本点間が連続関数で補間されるわけではない。しかしながら、図3に示すように得られた結果に対して再び同様の補間処理を繰り返すことによって、更に標本点密度を高めることができる。デジタル曲線の再現のためには、スプライン関数を用いた場合でも、曲線上のいくつかの点の座標値を具体的に計算する必要がある。このようなことからデジタル曲線を表示する上では、離散点を求めていく本方法でもスプライン関数を用いる場合と比較して、大きな不具合は生じないといえる。

実際にデジタル曲線を表示する際には、標本点間距離が3、4画素程度になるまで補間を行い、最終的にはそれらを直線で結ぶことによって曲線として表示することが処理時間の面からも適当であると思われる。

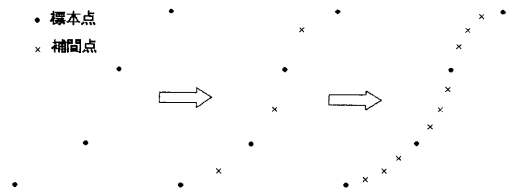


図3 補間の繰り返し

3. 曲線の部分的な補間

本方法によると、与えられた全ての標本点間に補間点を与えていくだけでなく、曲線の一部の区間を選んで、その区間のみ補間することも可能である。

例えば、曲率が小さい標本点間に対しては直線によって補間しても比較的 unnatural ではない。従って、曲率が大きい標本点間についてのみ本方法によって補間点を与えればよい(図4)。

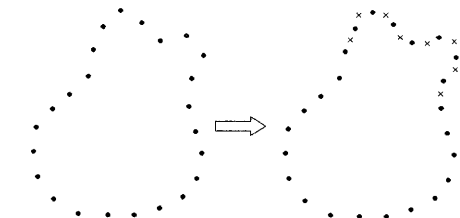


図4 曲率の大きな曲線部分の補間

又、標本化密度すなわち標本点間距離が場所によって異なる場合には、標本点間距離が比較的大きな部分だけ補間することにすれば、補間を効率よく行うことができる(図5)。なお、どの区間を補間するかは曲率もしくは距離についてのしきい値などを与えることによって自動的に選び出すことができる。このように補間区間を自動的に選択して、その区間だけを補間できることは、周波数空間による補間処理や一般のスプライン補間処理にはない機能である。

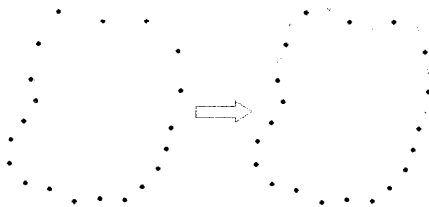


図5 標本点間距離の異なる曲線の補間

4. 実験

標本点で表された曲線の補間を行った。比較のために直線補間による方法、G型記述子による方法^[2]、3次スプライン補間による方法^[5]、以上3つの方法を用いた場合についても併せて行った。なお、P型記述子による曲線の生成もあるが、この方法では、補間後の曲線が与えられた標本点を通らないのでここでは除外した。

実験では、いかに滑らかな曲線が生成されるかに注目し、その指標として曲率を用いた。曲率の大きさが全体的に小さく、その変化が緩やかであれば、滑らかな曲線であるといえる。従って、生成された曲線の曲率の2乗和の大きさを比較した。又、曲率の変化を調べるために曲率関数のフーリエスペクトルについても比較した。

まず、N個の標本点からなる曲線データを曲率関数で表し、それに対してFFT処理を行い、N個の周波数パワースペクトル成分を得る。なお、標本点数としては、FFTを適用し易いように閉曲線の場合には2のべき乗個に、開曲線の場合には2のべき乗個+1個にした。つまり、標本点間の数が2のべき乗となるように与えた。補間後の曲線データは、標本点が増えるので周波数成分も高周波領域が増えることになる。曲線本来の形状を損なわないためには、元々存在していた周波数成分が変わらない方がよい。一方、滑らかな

曲線となるためには、新たに加わった高周波成分は、できるだけ小さいことが望ましい。以上のことに注目して、各補間方法による結果を比較した。

4. 1 閉曲線の補間

入力データは、図6に示す輪郭線を表す32点からなる標本点データである。このデータを各方法によって256点になるように補間した。それぞれの結果を図7、8、9、10に示す。それぞれ直線補間による結果、G型記述子による結果、スプライン補間による結果、提案している方法による結果を表している。又、それぞれの図において補間曲線、補間曲線の曲率関数、曲率関数のパワースペクトルの3つを示している。なお、入力データの曲率のパワースペクトルのグラフは、補間後のパワースペクトルのグラフと横軸の尺度を揃えている。

又、表1に各方法による補間曲線の曲率の2乗和の値と、元の標本点データの曲率関数の周波数スペクトルとの誤差を示す。ここで、表の低周波域とは、補間前から存在していた周波数成分の領域を意味しており、高周波域とは補間によって新たに生じた高周波成分の領域を表している。

図7から、直線補間によってできる曲線が折れ線になっていることがわかる。又、曲率関数がパルス状で周波数スペクトルも高周波成分が大きくなっていることから曲線が滑らかでないことがわかる。

G型記述子を用いた結果、スプライン補間による結果、提案している方法による結果は、ともに視覚的に滑らかな曲線となっており、曲率関数の変化も穏やかである。パワースペクトルも高周波成分がよく抑えられており、滑らかな曲線であることを表している。しかしながら、詳しくみるとG型記述子を用いたものは高周波成分はよく抑えられているものの、低周波領域での誤差が比較的大きくなっている。一方、スプライン補間によるものは、関数の性質から曲率の2乗和が他の2つに比べて小さくなっているが、提案している方法に比べて低周波のスペクトルの誤差は弱冠大きい。なお、G型記述子による結果とスプライン補間による結果はともに曲率の大きな部分の標本点密度が密になるように補間されており、標本点間の距離がほぼ等しくなるように補間される本方法によるものとは少し異なっている。

4. 2 閉曲線の補間

入力データは、図11に示す閉曲線を表す17点からなる標本点データである。このデータを各方法によって128点になるように補間した。それぞれの結果を図12、13、14、15に示す。それぞれ直線補間による結果、G型記述子による結果、スプライン補間による結果、提案している方法による結果を表している。

表2に各方法による補間曲線の曲率の2乗和の値と、元の標本点データの曲率関数の周波数スペクトルとの誤差についても示す。

直線補間による結果は、閉曲線の場合と同じく滑らかでなく曲率関数がパルス状で、パワースペクトルも高周波成分が大きい。また、G型記述子によって補間したものは閉曲線となっており、その形状も入力データから予想されるものとは大きく異なっている。

スプライン補間による結果と提案している方法による結果とは、ともに入力データから予想される滑らかな曲線となっている。曲率関数およびそのパワースペクトルもほぼ同様なものとなっている。曲率の2乗和と高周波成分の誤差は、やはりスプライン補間による結果が弱冠小さいが、低周波成分の誤差は、逆に提案している方法の方が小さい。

4. 3 パラメータ a, b について

パラメータ a, b は、先に述べたように曲率の変化がない場合には0.25となる。実際には、そのようなことはごく希であるが、補間処理を繰り返すと a, b の値は0.25に近づく。例えば、図10に示す補間曲線を得る際には、3回の繰り返し補間処理が必要であって、その際の係数の値は、

1回目 0.29728

2回目 0.27732

3回目 0.26077

となっており0.25に近づいている。これは、補間処理によって、各点における曲率が平均化されていくことを示している。

4. 4 処理に要する時間

スプライン関数による結果と提案している方法による結果とはほぼ同様なものが得られることが先の実験によってわかった。ここでは、処理に要する時間をスプライン関数による補間と提案している方法による補間とで実験によって比較した。実験結果を表3に示す。本方法によって32点のデータを64点に補間する際

に要する時間を1としたときに、それぞれの処理に要した時間の相対的な値を示している。少ない標本点を密に補間する必要があるときには、スプライン補間を用いた方が、提案している方法よりも処理に要する時間が小さくてすむ。しかし、入力標本点数が大きいときや、補間点の数が比較的小さいときには、提案している方法の方がずっと小さい処理時間で実行可能であることがわかる。

表1 曲率の2乗和とスペクトルの誤差
閉曲線の場合

	曲率の2乗和	スペクトルの誤差	
		低周波域	高周波域
直線補間	14.87	0	12.05
G型記述子	3.48	1.24	1.83
スプライン	3.14	0.40	2.00
提案方法	3.36	0.38	2.43

表2 曲率の2乗和とスペクトルの誤差
閉曲線の場合

	曲率の2乗和	スペクトルの誤差	
		低周波域	高周波域
直線補間	7.89	0	27.43
G型記述子	33.08	13.79	26.41
スプライン	1.61	0.33	2.55
提案方法	1.72	0.17	3.87

表3 処理時間の比較

標本点の数		処理時間	
補間前	補間後	スプライン	提案方法
32	64	2.0	1.0
	128	3.0	2.5
	256	5.5	6.5
	512	10.5	14.0
64	128	5.0	2.0
	256	9.0	5.5
	512	17.0	13.0
128	256	16.5	3.5
	512	30.5	11.0

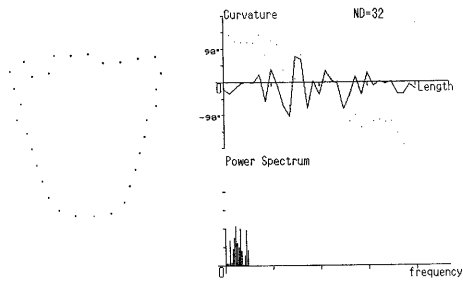


図 6 入力閉曲線座標データ

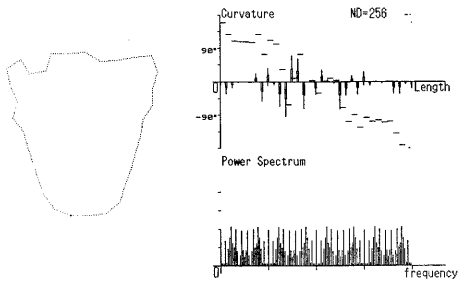


図 7 直線補間による結果

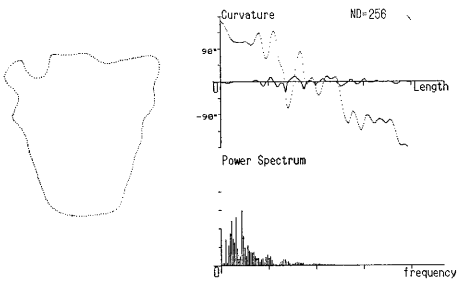


図 8 G型記述子による結果

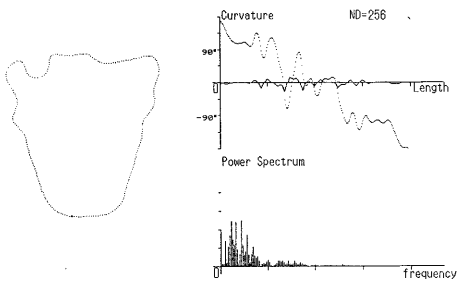


図 9 スプライン補間による結果

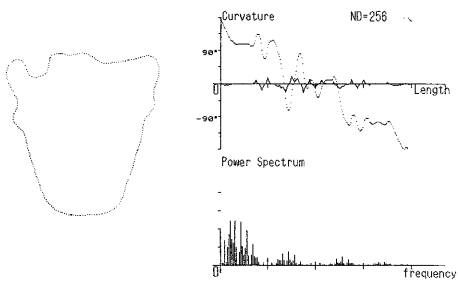


図 10 提案している方法による結果

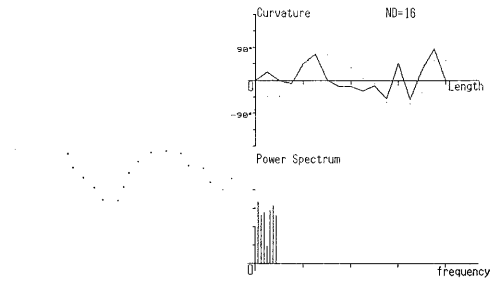


図 11 入力開曲線座標データ

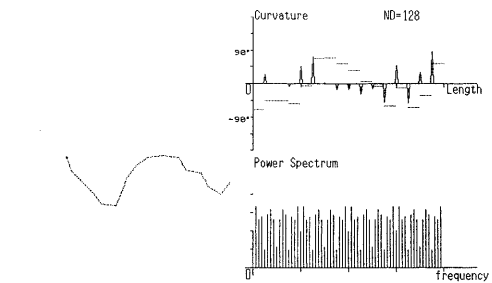


図 12 直線補間による結果

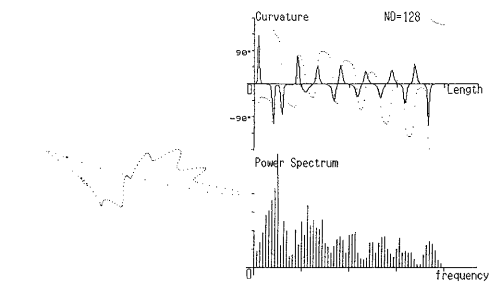


図 13 G型記述子による結果

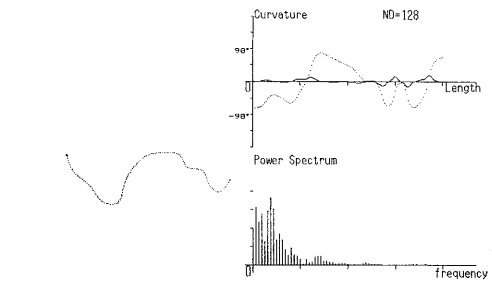


図 14 スプライン補間による結果

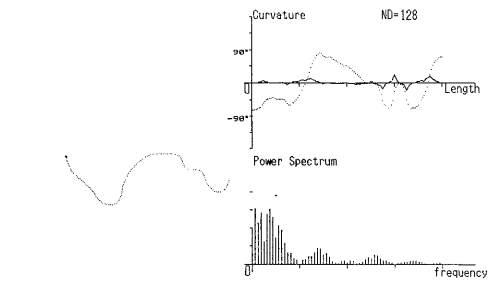


図 15 提案している方法による結果

5. むすび

3次スプライン関数による補間曲線は、理論的にも曲率の2乗和が最小になることが知られている。本補間方法によるとその値は、スプライン関数を用いた場合に比べて弱冠大きい。しかしながら、計算に要するメモリや処理時間が小さくてすむという利点がある。例えば、曲線を規定するN個の標本点が与えられたときにはスプライン関数による補間を行うためにはN元連立方程式を解く必要があるので、標本点の数Nが大きときには補間が実行できなくなる可能性がある。一方、提案している方法では、補間後の標本点とそれぞれの点における曲率とを保存できる程度の大きさのメモリがあれば実行可能である。又、処理時間については、ごくわずかの入力データに対して多数の補間点を与える必要があるときには、本方法はスプライン補間による場合よりも多くの時間を要する。しかし、入力データ数が多いときや、補間点の数が比較的少ない場合には、ずっと短い時間で実行できる。更に、曲線の一部の複雑な部分だけを選んで補間できることも本方法の特長である。

参考文献

- [1]小野, 瀧山: "線図形の等辺多角形近似とその認識への応用", 電子情報通信学会技術研究報告, PRU89-92, 1990.
- [2]G. H. Granlund: "Fourier preprocessing for hand print character recognition", IEEE Trans. Comput., Vol. C-21, pp. 195-201, 1972.
- [3]上坂: "開曲線にも応用できる新しいフーリエ記述子", 電子通信学会論文誌, J67-A, pp. 166-173, 1984.
- [4]安居院, 中島, 伊藤: "閉曲線情報処理手法による湖の識別", テレビジョン学会誌, 33, pp. 210-215, 1979.
- [5]市田, 吉本: "スプライン関数とその応用", 教育出版, 1979.
- [6]D. F. Rogers and J. A. Adams著, 山口訳: "コンピュータグラフィックス", 日刊工業新聞社, 1979.
- [7]小野, 瀧山: "標本点によって記述された曲線の補間法", 1992年電子情報通信学会秋季全国大会講演論文集, 分冊6, P. 294, 1992.
- [8]小野, 瀧山: "標本点によって表された曲線の曲率を用いた補間", 平成4年度電気関係学会九州支部連合大会講演論文集, P. 758, 1992.