

段階的クラスタを用いた描線順序の効率化†

雄 山 真 弓**

一般に作図は、2つの端点を持つ直線分、または連続する直線分からなる折れ線が複数個組み合わせられて構成されると考えることができる。これを一筆書きでその描線軌跡が最短になる経路を求める問題は、組合せ最適化問題として扱うことができる。線分が増加することによって、その組合せ数は増大し、計算量が指数関数的に増大するため、最適解は実用的には解けない場合が多い。したがってこれにかかわる近似解を求めることは重要である。本論文は、クラスタを段階的に行って近似解を求める方法を提案し、そのアルゴリズム、計算機実験の結果、2次元データについては、既に行われている方法との比較を行う。本方法の最大の特長は、3次元空間への適用ができることであるが、そのほかにも作図データを点、直線分からなる単純な構成として捉えていること、時間計算量、領域計算量について効率的な方法であること、さらに、高性能な近似解が得られることを述べる。

1. ま え が き

2次元平面または3次元空間に点、直線分、連続する直線分の組合せである折れ線が複数個組み合わせられて作られている作図がある。これらをすべて1回通り、その軌跡を最小にする描線順序を求めることは、組合せ最適化問題として定式化することができる。作図を構成する点、直線などの数が増加すると描線順序が指数関数的に増大するため、厳密解は求められなくなる。したがって近似解を求める研究が必要である。2次元平面については、すでにいろいろの研究がなされてきた。特に、その中で Iri らの方法は、作図をグラフ構造として最適化を行い高性能な近似解を得ている^{1)~3)}。また、Leipälä らの方法は、直線分または連続する直線分の組合せのすべてについて最短結合を求めて最適化を行っている⁴⁾。前者は、地図データのように、グラフ構造のデータ構成として、表現できるものについては、かなり良い結果が得られている。しかし作図データによっては、効率の悪い例もあげている。後者は、単純に連続する線分を見つけてその端点間の最短距離を求める方法である。端点の数が多い場合は組合せの数が非常に多くなり実用的でない。

以上のことから2次元のみならず3次元にも適応しうること、あらゆるデータに適用可能であること、さらに高性能な近似解が求まるアルゴリズムが望まれる。

本論文は、クラスタを段階的に行って、描線順序を求める方法を提案し、そのアルゴリズム、計算機実験の結果、2次元データについては、既に行われている

方法との比較を行う。本方法の最大の特長は、3次元空間への適用ができることであるが、そのほかにも、作図データを点、直線分からなる単純な構成として扱えること、時間計算量、領域計算量を十分考慮していること、作図データの形によらず高性能な近似解が得られることを述べる。

2. 効率的作図処理の条件

2.1 作図のための素データの構成

ほとんどすべての作図は、点、直線分、連続する直線分の組合せで構成されることができる。連続する線分は、直線分の集合であり、点は直線分の両端点が接近して同じ座標をもつと考えることができる。したがって、作図の基本構成は、2端点をもつ直線分 g が k 本ある場合、(1)式に示す集合 G として表される。

$$G = \{g_1, g_2, g_3, \dots, g_i, \dots, g_k\} \quad (1)$$

g_i は、2端点 p_i, p_i を持つ。これを(2)式に示す。さらに、 p_i, p_i は、それぞれの座標値 (x, y, z) と、 g_i の両端点である関係を示す値 b で構成される。これを(3)式に示す。

$$g_i = \{p_i, p_i\} \quad (2)$$

$$p_i = \{x_i, y_i, z_i, b_i\} \quad (3)$$

$$p_i = \{x_i, y_i, z_i, b_i\} \quad (3)$$

ここで、 b_i, b_i は、(4)の関係を持つ。

$$b_i = t \quad (4)$$

$$b_i = s \quad (4)$$

2.2 素データの整理と組合せの減少

描線順序の最適化を行うことは、線分の端点から他の線分の端点へ移動する距離の和を最小にすることである。素データは、直線分を表す2端点の座標値とそ

† Efficient Drawing Sequence by Stepwise Clustering Method by MAYUMI OYAMA (Information Processing Research Center, Kwansai Gakuin University).

** 関西学院大学情報処理研究センター

の連結関係の情報しか与えていない。したがって描線順序を求める場合、連続する線分の扱いは、次の2つの方法が考えられる。

1つは、①連続する線分を1線分とみなして描線順序を決定する方法である。この場合、連結した途中の座標値は考慮せず、2端点のみを考慮する。もう1つの方法は、②作図データを、連続不連続にかかわらずすべて直線分の組合せとして、各直線分の端点を用いて最短の組合せを求める。①は、プロッタによる作図の場合のようにペンの上下にかかわる時間を要する場合は、有効である。また、端点の減少になるため、組合せによる時間計算量を少なくすることができる。②の方法は、これができないが一般的なデータの扱いとして必要である。

一般に2端点をもつ k 本の線分を、その軌跡が最短となるように一筆書きするには、 $2k$ 個の端点の組合せについて調べることになる。 k が2の場合でも可能な組合せは4通りとなる。 k 個の線分を連続する1本の線分として結ぶには、それぞれの線分の方角と順序を考慮すると、全体の組合せ数 K は、(5)で与えられる。

$$K = 2^{k-1} \cdot k! \quad (5)$$

①を適用し n 個の端点が連結によって m 個に減少したとすると、端点比較組合せ数の比は、(6)式の K_r で与えられる。端点の減少は、時間計算量の減少となることは、明らかである。

$$k_m = m/2 \quad k_n = n/2 \quad \text{ここで} \quad n \geq m$$

$$K_r = K_n / K_m = 2^{n-m} \cdot (k_n! / k_m!) \quad (6)$$

m と n の差が大きいほど K_r の値は、大きくなる。ただし、②の場合も重要であり両者について計算機実験を行う必要がある。

2.3 最短距離組合せの方法

最短の描線順序となる線分の組合せは、単純に最適な端点を結んで行くだけでは最適な描線順序を求めることはできない。例えば、図1[1]の g_A, g_B, g_C の3つの線分の間で最短結合を考える場合、 g_A と g_B の間で端点の距離比較が4通りあり、 g_B, g_C と g_C, g_A に

ついてそれぞれ4通り、合計12通りの中から最短のものを選んで1回目の結合をする。2回目に残る2線分の端点の組合せで、4通りとなる。1回目と2回目の結合距離の合計を最短にする端点組合せが3線分の最短軌跡を求めることになるが、図1[2]のように、必ずしも1回目に最短となる端点の組合せを行ったものが、合計を最短にするとはかぎらない。図1[3]は、2回目の結合を考慮して1回目の結合を行い最短軌跡を求めている。そこで、線分の数 k が3以上の組合せについては、線分結合距離の合計が最短となるアルゴリズムを考えなければならない。

2.4 距離計算の方法

端点間の距離計算は、ユークリッド距離 L_2 を用いる。ユークリッド距離は、実距離を計算するのに最適であるが、各端点間の距離比較を行うには、相対的な値で十分である。したがって、絶対距離の中で距離の差の絶対値の最大値、最大値ノルム L_∞ を用いることもできる。 L_∞ は、距離比較を行う場合は、計算処理の容易さでは他の方法より優れている。したがって計算機実験では、 L_2, L_∞ の両方について、計算を行う必要がある。ただし、軌跡が決った後の実距離の計算についてはユークリッド距離を用いる。

2.5 近似解の最適化度を計る方法

近似解の精度を計る方法として以下に述べる最適化率を用いる。

χ_i = 厳密解の軌跡, χ_i = 入力データの軌跡

χ_0 = 最適化を行った近似解の軌跡

χ_i, χ_0 は、目的とする作図を描くために入力データの全体軌跡から線分の部分を除いた空間移動距離の合計である。厳密な最適化率 P_i は(7)式となる。

$$P_i = \{(\chi_i - \chi_0) / (\chi_i - \chi_0)\} \cdot 100 \quad (7)$$

厳密解 χ_i は端点数が非常に少ない場合を除き求めることはできない。したがって $(\chi_i - \chi_0)$ の代りに χ_i を用いた(8)式を最適化率 P として用いる。

$$P = \{(\chi_i - \chi_0) / \chi_i\} \cdot 100 \quad (8)$$

$P_i > P$ であり、本報告は最適化度として P を用いる。

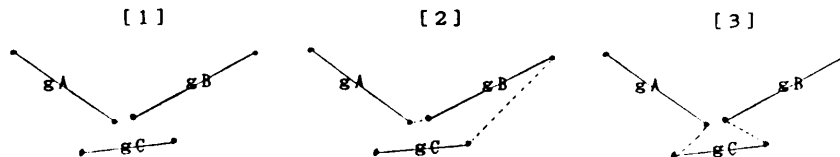


図1 3線分の接続と最短軌跡

Fig. 1 Combination of 3 segments and length of locus.

3. クラスタ化による描線順序の最適化

3.1 段階的クラスタを用いる意味

段階的クラスタ化は、クラスタ化を繰り返し行う方法を用いている。これは、以下に示す5つの項目について特に優れた特長をもっている。

- ①入力する作図データの構造は、あらゆる作図に適用できるように単純な形である。
- ②端点間距離計算のための組合せを少なくし時間計算量を減少する方法を用いている。
- ③領域計算量を少なくする方法を用いている。
- ④高性能な近似解を求めるため、端点間組合せを巨視的に捉える方法を用いている。
- ⑤2次元平面にかぎらず3次元空間への適用も可能である。

①から⑤の特長にたいして欠点をあげるならば、クラスタ化に伴うパラメータの設定が難しい。次に段階的クラスタ化の方法と上記の特長について述べる。

3.2 段階的クラスタ化法とアルゴリズム

散在する直線分の2端点の関係値と座標値を入力データとして、端点をクラスタに分けて、各クラスタ内での最短結合を総あたりで求め、端点を順次結んでいく。その際、線分がクラスタ間に跨がる端点同士は結合から除外する。残る端点を集めて再クラスタ化を行い最短結合を行う。再クラスタ化は残る端数が十分少なくなるまで繰り返す。最後に残る端点をすべて用いて、結合残りの端数が2になるまで最短結合を行う。

この方法は、クラスタ化によって端点間距離比較のための組合せを大幅に少なくし、さらに、クラスタ内での結合で、クラスタ間に跨がる長い直線分は残し、次段階のクラスタ化へ回す、いわゆる端点組合せを巨視的に行うのがポイントである。各段階のクラスタ化では、クラスタ数を決定するセルの大きさは、残る端点数とパラメータλを用いて算出される。パラメータλは、端点数nと端点の分散および各線分の長さなど、種々の値によって決定すれば、より有効なクラスタ化が行われるはずである。しかし、時間計算量や、領域計算量を十分考慮しなければならない。本報告ではクラスタ数と残る端点数の関係をフィードバックさせパラメータをダイナミックに変化できる方法を採用している。

段階的クラスタ化のアルゴリズムを図2に示す。

- [1] 作図データの入力
2.1で述べた各直線分の両端点の関係値bと座標値(x, y, z)を入力する。

- [2] 初期値パラメータλの設定
端点数と作図の大きさによって決定する。(λ>0)

- [3] クラスタのためのセルの決定
n個の端点の座標値を用いて、それぞれの最大値、最小値を計算する。

セルの大きさを決定するhは、(9)式で与えられる。

$$h = (x_d \cdot y_d \cdot z_d / \lambda \cdot n)^{1/3} \quad (9)$$

ここで $x_d = x_{\max} - x_{\min}$

$y_d = y_{\max} - y_{\min}$

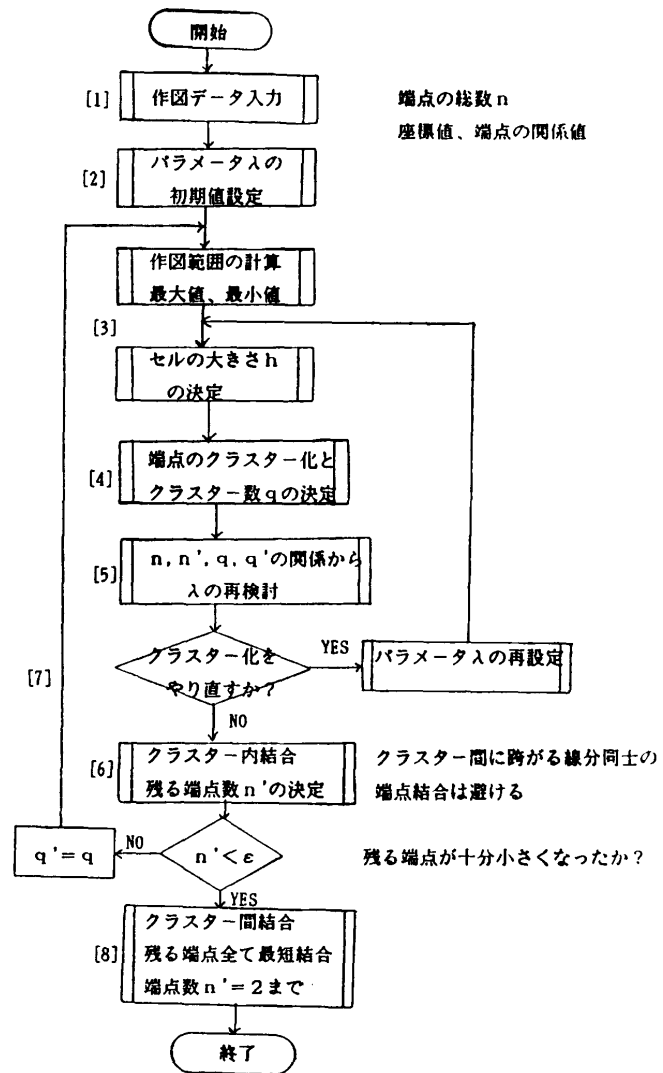


図2 段階的クラスタ化法のアルゴリズム
Fig. 2 Flowchart of stepwise clustering method.

$$z_d = z_{\max} - z_{\min}$$

[4] クラスタ分けとクラスタ数の決定

1辺が h の立方体で分け、 $\lambda \cdot n$ 個のセルを作る。セルを順次サーチして端点のあるセルが連なっている場合は、同一クラスタと見なす。

$\lambda \cdot n$ 個のセルをサーチすることによって、グループ化は可能である。 q 個のクラスタに分けられたとする。

[5] パラメータ λ の再検討

クラスタ数 q と端点数 n が以下の条件を持つ場合パラメータ λ を変更する。

$$q = n \text{ ならば } \lambda = \lambda - \alpha \quad (\alpha > 0)$$

$$q = 1 \text{ ならば } \lambda = \lambda + \alpha$$

段階数が2回以上の場合、

$$n' = n \text{ ならば } \lambda = \lambda - \alpha \quad (n' \text{ は前回の端点数})$$

[6] 各クラスタ内での最短結合

各クラスタを C_j で表す。 $\{C_j, j=1, q(1 < q < n)\}$ とする。

各クラスタ C_j の端点数を a_j とする。

(a) a_j のうち、線分の両端点がクラスタ C_j に含まれる端点数 a_{j1}

(b) 異なるクラスタに跨がる線分の端点数 a_{j2} ここで、 $a_j = a_{j1} + a_{j2}$ とする。

(b)に属する端点同士の結合は除き最短結合を行う。 q 個のクラスタについて行う。

結合終了後、クラスタ C_j に残る端点数 b_j は、 $b_j = a_{j1}$ となる。残る全端点数 n' は、(10)式で与えられる。

$$n' = \sum b_i = \sum a_{j1} \quad (10)$$

[7] クラスタ化の繰り返し

n' を n として[3]から繰り返す。 n' が十分小さな値 ϵ (実際には、使用するコンピュータの処理速度を考慮するとよい) になったら[8]の処理に移る。

[8] クラスタ間結合

残る端点 n' について、端点の数が2になるまで最短結合を行う。

2.2の①で述べた連続する線分をあらかじめ連結してから行う方法は、作図データの特徴に左右される。アルゴリズムでは、[1]と[2]の間に挿入する。

3.3 段階的クラスタ化とクラスタ手法

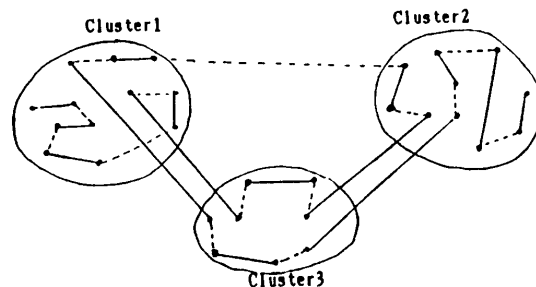
作図データをクラスタ分けするには、種々のクラスタ手法⁵⁾について検討する必要があるが、段階的クラスタ化におけるクラスタ手法については、筆者は既に報告した⁶⁾。クラスタ化を行う場合の作図データの扱

いは、次の2つの方法がある。1つは、①端点をそれぞればらばらに扱ってクラスタ化する方法と、②線分の2端点を1組のデータとしてクラスタ化する方法である。①は、同じクラスタに線分の両端点が含まれる場合と、含まれない場合ができる。つまり1線分が2つのクラスタに属し、曖昧さ(Fuzzy)を持つクラスタ化となる。両端点、同じクラスタに含まれる場合は最短結合をすることができるが、含まれない場合は、他方の端点が含まれるクラスタの状態と関連するため、やみくもに結合すると、線分がループを作ってしまう場合が生じたり、クラスタ間を結ぶ空間軌跡が増加する場合がある。したがって、段階的クラスタ化では、クラスタ内結合においては、クラスタ間に跨がる端点同士の結合は残し、次のクラスタ化に回す方法を用いて順次端点数を減らしていく。結果的には、②の線分の方法を段階を分けて行っていくことと類似している。

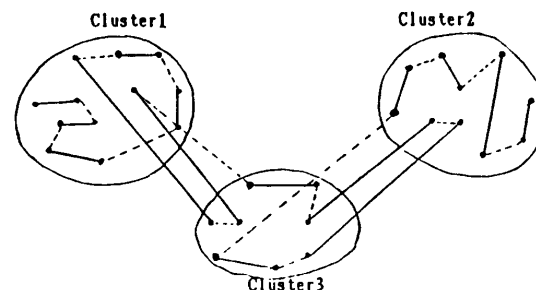
クラスタを用いて行う場合に問題となるのは、以下の場合である。

端点数 n 、前段階の端点数 n' が十分大きく、クラスタ数 q とすると、

(1) $q=1$, (2) $n=q$, (3) $n'=n$ の場合である。



[1] クラスタに跨がる線分同士の結合を次段階に回した場合クラスタ間の空間移動は1回となる



[2] クラスタ内ですべての端点に最短結合を行った場合クラスタ間の空間移動は2回となる

図3 クラスタ内、クラスタ間での最短結合の例
Fig. 3 Two cases, combining nearest point of intra-cluster and inter-cluster.

これらの対策としてはアルゴリズムで既に述べた。しかし、(1)の状態がパラメータの変化で変わらない場合は、クラスタの数を指定してクラスタ分けを行う手法⁷⁾をとれば、処理できることは、筆者が既に発表した論文⁶⁾で計算機実験を行って検証した。指定するクラスタの数については、クラスタ化の段階数が1回の場合に、組合せが最小となるクラスタ値の概算値を算出している。

実際は、パラメータ λ は(11)式で表現される値であることが予想される。

$$\lambda = \phi(v, r1, r2, d) \tag{11}$$

v : 端点の分散

$r1$: 線分の長さの分散

$r2$: 線分の長さの平均

d : 端点の密度 (体積にたいする端点数)

関数 ϕ を決定する値が見つけれれば、各段階における端点数の絞り込みが容易になる。しかし、関数 ϕ を求める時間計算量について考慮しなければならない。本報告では、アルゴリズムで述べた方法で行った。

3.4 段階的クラスタと時間計算量

クラスタを段階的に行って、組合せ数を少なくする方法は、分割統治法⁸⁾を段階的に行っていることになる。分割統治法に係わる手間は $O(n \log n)$ である。段階的に行う方法は、その効果をより大きくしている。ただし、クラスタ化のための計算量について考慮しなければならない。クラスタを求める方法として $n \cdot \lambda$ のセルに分割して各セルに含まれる端点を探しグループ化するが、2次元データについては、隣合う8方向のセルをサーチしていく。したがってサーチの回数 S は、(12)式で与えられる。

$$S = 4(2n\lambda + (n\lambda)^{1/2} - 5) \tag{12}$$

サーチに係わる手間は、 $O(n)$ である⁹⁾。さらに、クラスタ化の段階数は、高々2桁になることはない。

3.5 段階的クラスタ化と領域計算量

計算に必要な領域は、入力時に用意する各端点の座標値と端点の接続関係を示す値、順次連結を行った場合に新しい線分の端点関係を示す値、およびクラスタのグループ値を入れる値であり、特別に大きな領域を必要としない。入力時の端点数 n の数倍のオーダーの領

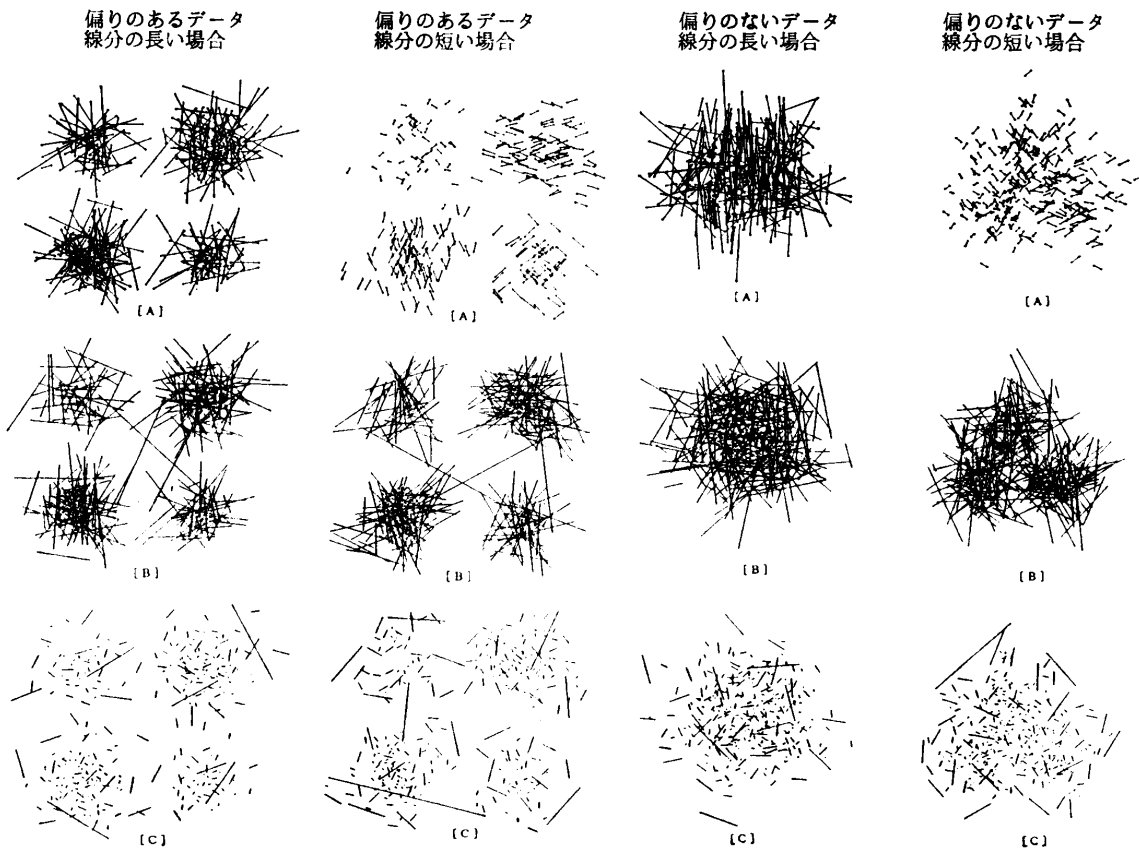


図 4 乱数を用いて作成したデータの実験結果

Fig. 4 The experimental results of 2 dimensional data using random number.

域で十分計算可能である。

3.6 段階的クラスタ化と近似解

高性能な近似解を求めることの基本となる最短の端点間距離を探索する方法は、実際はクラスタ内線分の連結を最短にする軌跡を求めることを繰り返し行うことになる。その際、クラスタ間に跨がる線分をどう処理するかが段階的のクラスタ化を繰り返す場合問題となる。図3は、クラスタ化を行った後、[1]クラスタ間に跨がる端点を後回しにした場合、[2]最短の端点のみに注目して連結を行った場合の比較を行っている。

いずれの場合もループにならないように連結している。[1]の方法は、クラスタ間の空間移動軌跡が1本であるのに対し[2]では2本になっている。一般に、クラスタ内距離<クラスタ間距離の関係から、[1]の方法は、より高性能の近似解を求める条件となる。しかし、点データや、線分の長さの短い線分からなる場合は、クラスタ内の最短結合のみが近似解を求める条件となる。

4. 計算機実験の結果

4.1 乱数で作成したデータを用いた実験

以下のデータを乱数で作成し、計算機実験を行った。

- ①クラスタ化しにくい偏りのないデータ
- ②クラスタ化しやすい偏りのあるデータ

①②について、(a)線分の長い場合、(b)線分が短い場合、(c)線分に長・短のある場合、(d)点データの場合について端点数を変化させて実験を行った。2次元・3次元データ双方について行ったが、表示の容易さから2次元データの結果について図4に結果の一例を示す。端点数は、600で①、②で(a)、(b)について行った結果である。300本の線分からなる作図データ[A]、作図を一筆書きするための空間移動軌跡(計算前)[B]、計算後の空間移動軌跡[C]を示す。最適化率はすべて83%以上であった。

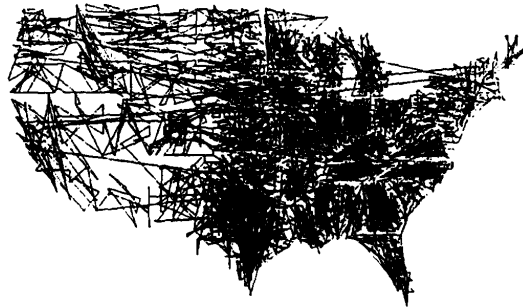
4.2 実際のデータを用いた結果

2次元データとしてアメリカの郡地図を入力データとして用いた結果を図5に示す。

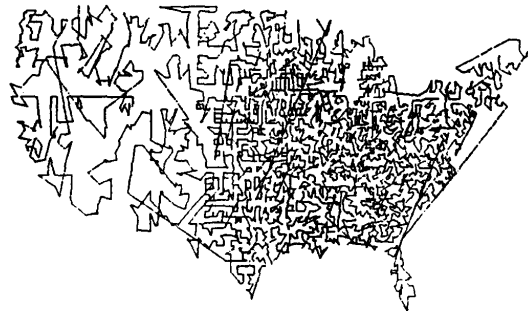
端点の総数は、85044である。[A]は、入力データのすべての線分を通る軌跡をペンを下げて描いたものである。したがってペンが上って作図処理には描かれない軌跡についてもその動きを示すため描いている。[B]は、[A]の作図線分のうち連続するものを1線分として端点数を減らしたもので、ペンの空間の軌跡の部分は同じである。[B]では端点数は、6166で



[A] 入力データ (端点の数=85044, 線分の数=42522)
[A] Input data. (Number of end points=85044, Number of segments=42522)



[B] 段階的クラスタ化を行うため連続する線分を1線分として処理したデータ (端点の数=6166, 線分の数=3083) の最適化前の軌跡
[B] Continuous segments are regarded as one segment. (Number of end points=6166, Number of segments=3083)



[C] [B]を最適化した後の一筆書きの軌跡
[C] After optimization.

図5 すべての線分を1回通り一筆書きした軌跡
Fig. 5 The locus of the pen passing all the segments once.

ある。[C]は、最適化後の[B]の軌跡である。この計算では、6段階のクラスタ化が行われた。6段階目は9個のクラスタとなり端点数が減少した。ペンを上げた軌跡の長さは、入力データは、149657であり最適化後は、21172である。最適化率は、85.01%である。

3次元データとして蛋白質の分子構造のデータを用

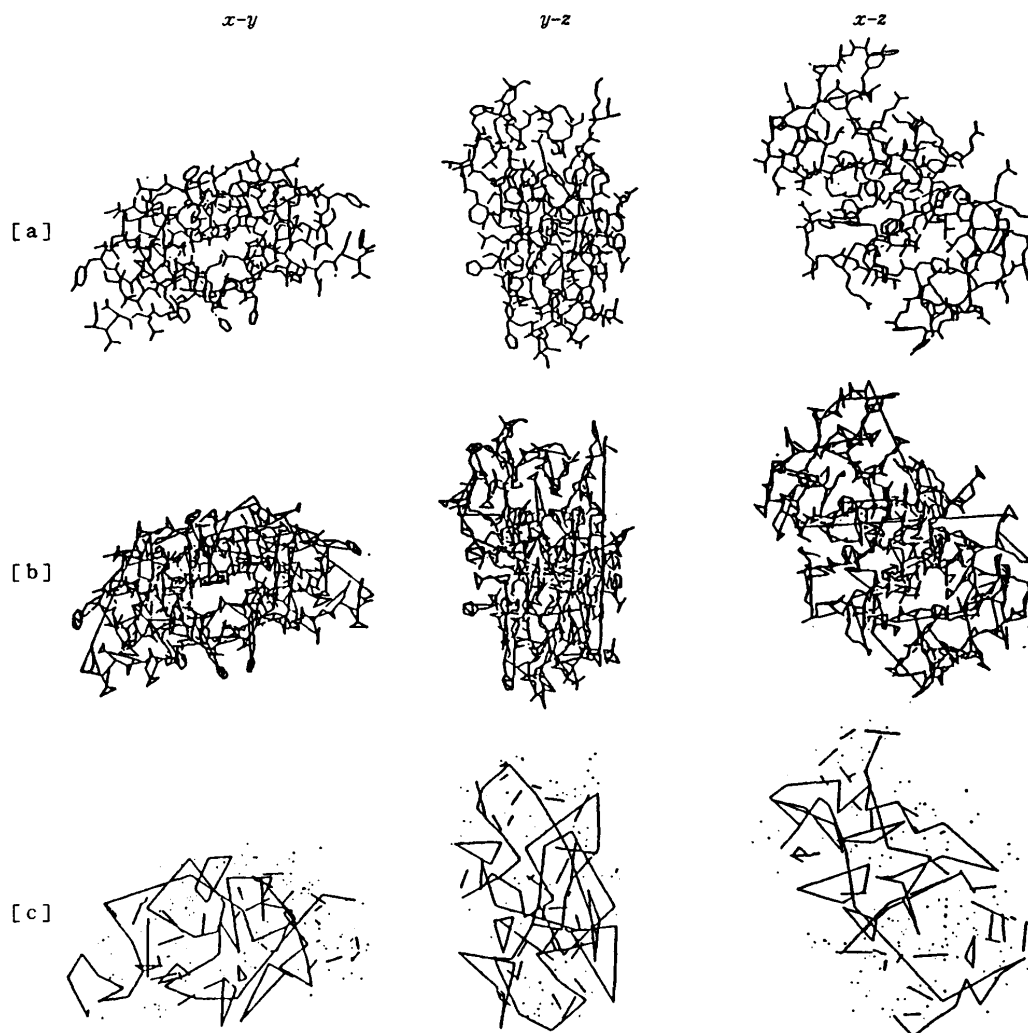


図 6 3次元データの例 (蛋白質の分子構造データ)

Fig. 6 The optimized results in the case of using the molecular structure data of protein. ($n=3312$)

[a] shows the structure of protein.

[b] shows the loci of a moving pen for writing protein.

[c] shows the optimized loci of movement.

いて行った結果を図 6 に示す。最適化率は 74.26% であった。

4.3 λ の変化に伴う最適化度と組合せの関係

図 7 は、段階的クラスタ化の方法で行った結果を示す。パラメータ λ の値を変化させて最短経路を求めるための組合せの数と最適化率の関係を示している。入力データは、筆順による漢字データを縦横 10 個ずつ並べたものである。端点の数は 3058 (連続した点を含めないと 2012) である。 λ を 0.4 から 7 まで変化させてクラスタ内 [a], クラスタ間 [b], 合計 [c] ($=[a]+[b]$) それぞれの端点の比較回数を示している。比較回数は、 λ の値が 2 以上になると、急激に

減少しているのがわかる。また、最適化率 [c] は、60% から 70% で、 λ の変化による顕著な変化は見られない。したがって、組合せの数が少なくなる λ があることが実験事実としてわかった。

4.4 2次元データについて他の方法との比較

段階的クラスタ化の結果の良否を示すには、地図データで優れた結果を得ている Iri らの方法との比較を行うのが良い。比較する方法のプログラムは、OPlot User's Manual のプログラム¹⁰⁾を用いて計算した。端点の数の多いデータの場合と、少ないデータの場合について双方の比較を行った結果を図 8 に示す。いくつかのデータについて比較を行ったが、いず

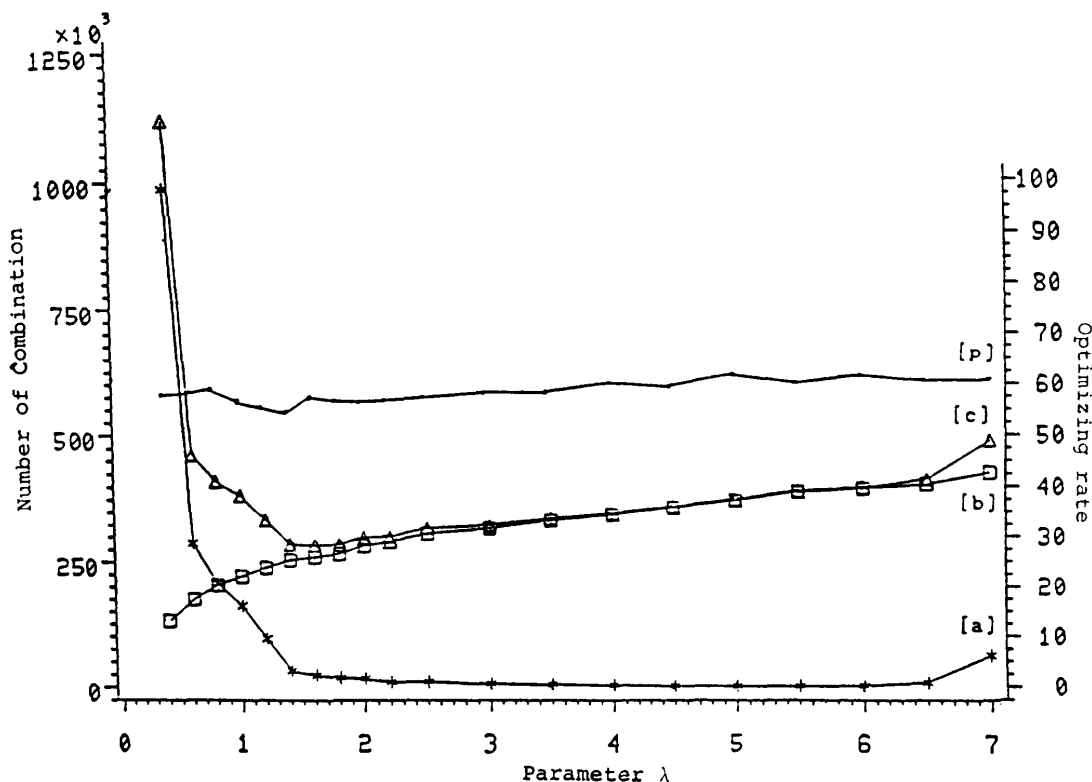


図7 段階的クラスタ化によるパラメータλと組合せの数(左側縦軸目盛)と最適化率(右側縦軸目盛)の関係(100個の漢字データ, 端点の数3058)

Fig. 7 Relationship between parameter λ, the number of combinations (vertical axis), and the rate of optimization (horizontal axis) for gradual clusterization. (Number of end points=3058)

- [a]=Number of intra-cluster combinations
- [b]=Number of inter-cluster combinations
- [c]=[a]+[b]
- [d]=Ratio of optimization

表1 グラフとクラスタの各方法を行った場合の最適化率の比較

Table 1 Comparison of optimization ratios with graphic and clusterization methods

[s: number of vertexes, l: number of segments, b: number of branches, n: number of end points, C: cluster, G: graph]

番号	方法	データ	cpu (1/100 sec)	最適化率%	番号	方法	データ	cup (1/100 sec)	最適化率%
1	G	s=26, b=13	0.27	69.75	5	G	s=1248, b=762	4.47	9.29
	C	n=26, l=13	0.17	94.26		C	n=1010, l=505	12.46	53.89
2	G	s=24, b=12	0.27	50.50	6	G	s=2498, b=1529	13.52	35.48
	C	n=24, l=12	0.15	92.20		C	n=2012, l=1006	46.23	60.39
3	G	s=52, b=26	0.32	61.99	7	G	s=3194, b=4032	40.17	58.06
	C	n=52, l=26	0.22	81.64		C	n=580, l=290	10.13	69.29
4	G	s=1106, b=680	3.78	27.33	(注) 使用したコンピュータは日立 M-240H				
	C	n=886, l=443	9.03	63.03					

れも段階的クラスタ化による方法は、最適化率について良い結果が得られた。表1に結果の比較表をあげておく。特に端点の少ない場合で、最適化ができない例

として報告¹⁾されていたものを、表1のデータ番号1と2にあげているが、段階的クラスタ化による場合は、最適化率、CPU時間ともに優れている。しかし、

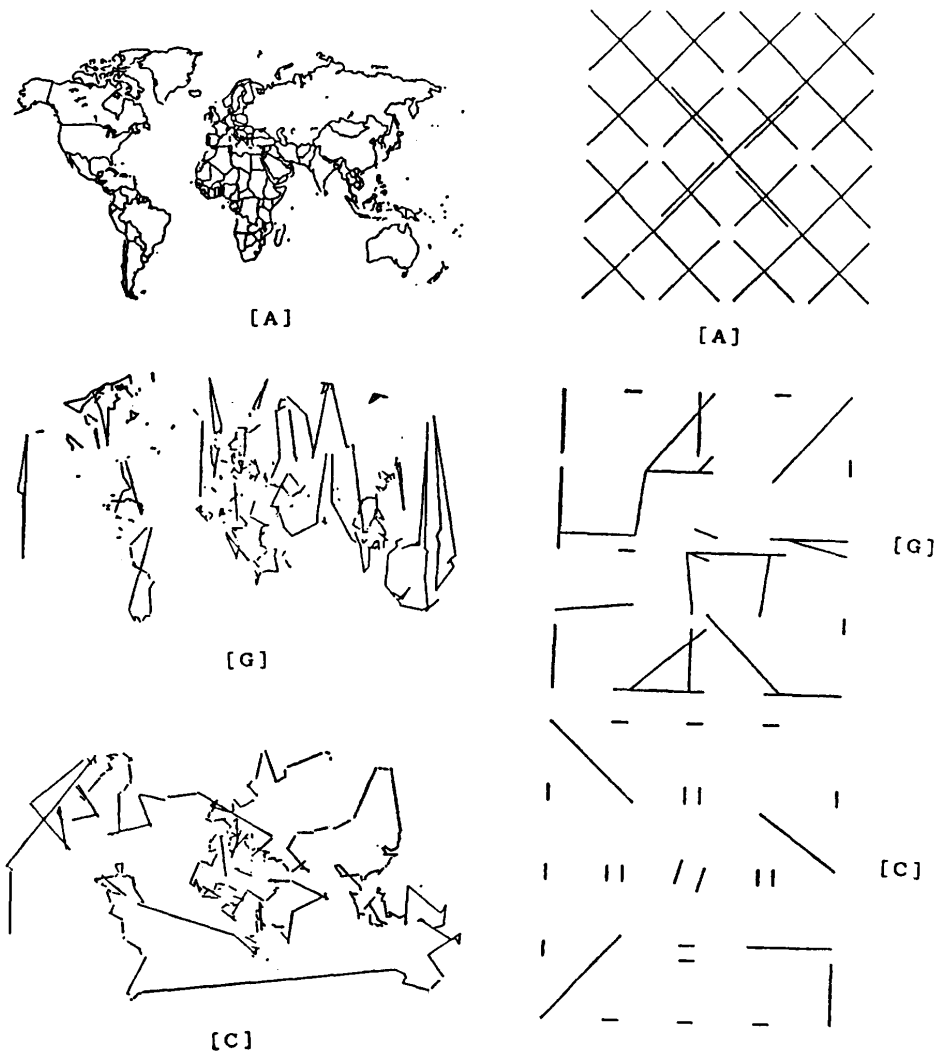


図 8 入力データ [A] を一筆書きで描く軌跡の最適化. [G] はグラフによる結果, [C] はクラスタによる結果
 Fig. 8 Optimization of locus of pen when drawing input data [A] in one stroke, [G]: Result of graphic method, [C]: Result of clusterization.

CPU 時間については、端点の数が多くなると、処理時間が長い場合もある。また、図 6 についても比較を行ったが、OPLLOT はリソースの点で処理できないこともわかった。そこで連続する線分を 1 線分として端点数を 6166 に減らしてから行った結果、最適化率は、81.3% であった。しかし、端点の計算上での削除は、本来のグラフ構造を用いる意味に反するため、この結果は、あまり比較にならない。

作図が地図データのように表現できる傾向が大であるものについては、比較を行った方法は効率がよいことがわかる。しかし、一般の作図に用いるデータは、2つの端点を持つ直線分または、折れ線分が、いくつか組み合わせられて作られているものが多い。段階的ク

ラスタ化は、一般の作図データをそのまま用いて処理を行うことができる。また作図の形によらず安定した結果を得ることができる。以上のことから、最適化処理を行うには、作図の特徴をとらえて処理にかかるのが理想的といえるが、そのような処理は一般には行わないのが通常である。どのような作図データが与えられても処理が簡単にできて、安定した最適化率を得ることができるのが段階的クラスタ化の最大の特長である。計算には、日立 M-240 H と M-280 D を使用した。

5. むすび

描線順序の最適化は、必ず通らなければならない複

数の線分を、いかに無駄（空間のあそび）を少なくして通るかの問題を解くことである。線分が3次元空間に存在する場合の最適軌跡も同じ問題である。段階的クラスタ化による描線順序の最適化は、作図データをそのまま処理できること、作図の形によらず安定した高い最適化率が得られること、処理効率が良いこと等について報告した。さらにこの方法は、3次元空間への拡張が可能なのは最大の特長であり、これによって種々の応用が考えられる。今後の課題として線分に結合制約条件が与えられた場合の描線軌跡を最適化する方法について研究を進めていく。

謝辞 貴重なお意見をいただいた統計数理研究所の水野欽司教授および上智大学機械工学科の浅野孝夫助教授に感謝いたします。蛋白質のデータをご提供くださいました住友化学工業株式会社の吉田元二氏に感謝いたします。

参 考 文 献

- 1) Iri, M., Murota, K. and Matsui, S.: Linear-Time Approximation Algorithms for Finding the Minimum-Weight Perfect Matching on a Plane, *Inf. Process. Lett.*, Vol. 12, pp. 206-209 (1981).
- 2) Iri, M., Murota, K. and Matsui, S.: An Approximate Solution for the Problem of Optimizing the Plotter Pen Movement, *Processing of the 10th IFIP Conference on System Modeling and Optimization, Lecture Notes in Control and Information Science*, 38, pp. 571-580 (1982).
- 3) Iri, M., Murota, K. and Matsui, S.: Heuristics for Planar Minimum Weight Perfect Matchings, *Networks*, Vol. 13, pp. 67-92 (1983).
- 4) Leipälä, T. and Nevalainen, O.: A Plotter Sequencing System, *Comput. J.*, Vol. 22, pp. 313-316 (1979).
- 5) Anderberg, M. R.: *Cluster Analysis for Applications*, Academic Press, London (1973).
- 6) 雄山真弓: 段階的クラスタ化による効率的描線法, 情報処理学会論文誌, Vol. 28, No. 11, pp. 1135-1145 (1987).
- 7) Sarle, W. S.: Cubic Clustering Criterion, SAS Technical Report A-108, SAS Institute Inc. (1983).
- 8) 伊理正夫, 野崎昭弘, 野下浩平: 計算の効率化とその限界, 入門「現代数学 13」, p. 6, 日本評論社 (1980).
- 9) 伊理正夫: 計算幾何学と地理情報処理, bit 別冊, p. 114 (1986).
- 10) Imai, H.: OPLLOT User's Manual, Department of Mathematical Engineering and Instrumentation Physics, Faculty of Engineering, University of Tokyo (1982).

(昭和61年12月16日受付)

(昭和63年10月7日採録)



雄山 真弓 (正会員)

昭和16年生。昭和38年東北大学理学部化学科卒業。昭和42年関西学院大学理学部実験助手、昭和51年同大学情報処理研究センター講師、昭和55年同助教授、現在に至る。作図処理の最適化、データ解析、教育のためのソフトウェアの開発、私大間ネットワークなどの研究、情報処理教育に従事。IEEE学会、応用統計学会、CAI学会各会員。私立大学情報処理教育等連絡協議会ネットワーク研究分科会委員。