段階的クラスター化による効率的描線法†

雄山 真弓††

2次元平面または3次元空間に複数個の線分が組み合わされて構成される作図を考える．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．本論文は，クラスターを用いた描線軌跡の最適化の方法を提案し，この方法を用いて計算機実験を行った結果，高性能な近似解が求められたことを述べる．さらに，クラスター化を行うための種々のクラスター分類法について比較検討し，重複を許すクラスター化法を段階的に行う方法が，作図データの構造によらず，高性能な近似解を効率的に得られたことを述べる．従来，2次元でのデータを扱う方法については種々の研究が行われてきたが，本論文の方法は，2次元データと共に，3次元データについても扱えること，線分データと点データが複合した作図データについても扱えるのが特長である．クラスター化によって作図データをとらえる方法は，人間がものを見るときにパターンとしてとらえる動作と類似している．デジタルな処理を得意とするコンピュータ上で，これをシミュレートするにはクラスター化は有効な方法といえる．パターン認識に優れた能力を持つ人間も，複数の線分の描線順序を最適化する能力は，線分の数が増えると，対応できなくなる．したがって描線順序の最適化は，コンピュータによる処理によって初めて可能となる問題であることがわかった．

1. はじめに

2つの端点を持つ直線分や連続する直線分の組合せである線分が，2次元平面または3次元空間に複数個組み合わされた構造をもつとき，すべての線分を1回通り，最短で連続する軌跡を考える．これは，一般にはTraveling Salesman問題と呼ばれ，端点の数の増加により組合せの数が急増し，厳密解は求められず，それに代る高性能な近似解を求めるためのヒューリステックな方法が必要となる．

その方法としては，既にグラフ構造を用いて求める方法[1),2)]や，最短の端点を順次結合していく方法[3)]が発表されている．前者は線分間の連結状態をすべて与えた2次元における処理においては，かなり良い結果を与えている．後者は連続する線分をあらかじめ連結し，端点の数を減少後，端点間の最短距離を求める方法であり，連結できる線分が少ない場合には，問題がある．

筆者は，線分の端点をクラスター化し，最適な描線軌跡を求める方法を提案し，これによって効率的に高性能な近似解が得られたことを報告する．この方法は，各クラスター内に2端点が含まれる線分についてのみ総あたり法で最短結合を行い，線分を連結する．連結していない端点をさらにクラスター化し，同様に連結する．以上を端点の数が少なくなるまで段階的に繰り返し，最後にクラスター同士の接続順序を総あたり法で最短結合し，最短描線順序を求めるものである．

与えるデータは，各線分の2端点の座標値のみであり，グラフ構造を用いる方法のように，線分間の交差点の情報をあらかじめ与えておいたり，計算で求めたりする必要はない．3次元データや，点データについても扱えるのが特長である．

本論文では，クラスター化の手法として，非階層的な分類法である重複を許すクラスター化法，分割最適化法とモード探索型方法の3つの方法を比較検討する．この中で筆者が提案する方法は，重複を許すクラスター化法を段階的に行い，作図データに適用したもので，他の方法に比べ高性能な近似解を効率的に求めることができた．計算は，HITAC-280Dを用いて行った．

2. 作図データ構成と最適化のための整備

2.1 線分データ構成と連結組合せ数

2次元平面または3次元空間における図形Gは，2端点をもつ直線分の組合せで構成されると考えることができる．Gがk本の直線分からなる場合は，

$$g_1\, g_2\, g_3 \cdots g_i \cdots g_k \subset G$$

で示される．g_iは直線分であり，その2端点の座標値をp_{i1}, p_{i2}とする．端点数の合計は，線分数の2倍となる．

各直線分をすべて1回通り，最短の描線順序で一筆

† Optimization of Drawing Sequence by Consecutive Clustering Method by MAYUMI OYAMA (Information Processing Research Center, Kwansei Gakuin University).

†† 関西学院大学情報処理研究センター

書きするには，k 本の直線分の方向と順序を考慮して各直線分間の移動距離の和が最短になるものを見つければよい．

具体的には，2直線分 g_i, g_j についてその結合を考えると，4通りの組合せがある．ここで，$d(p_{i1}, p_{j1})$ は2端点 p_{i1}, p_{j1} 間の距離を表す．

$$d_{ij1}=d(p_{i1}, p_{j1}),\ d_{ij2}=d(p_{i1}, p_{j2})$$
$$d_{ij3}=d(p_{i2}, p_{j1}),\ d_{ij4}=d(p_{i2}, p_{j2})$$

このうち最短の組合せ D_{ij} は，

$$D_{ij}=\min(d_{ij1}, d_{ij2}, d_{ij3}, d_{ij4}) \quad (1)$$

k 本の線分のうち，最短なもの R_1 は

$$R_1=\min[D_{ij}] \quad (2)$$
$$(i=1, \cdots, k,\ j=i, \cdots, k,\ i\neq j)$$

次に残る $k-1$ 本の線分の中から同様に

$$R_2=\min[D_{ij}] \quad (3)$$
$$(i=1, \cdots, k-1,\ j=i, \cdots, k-1,\ i\neq j)$$

残る線分の数が $k-t$ 本の場合は

$$R_{t+1}=\min[D_{ij}] \quad (4)$$
$$(i=1, \cdots, k-t,\ j=i, \cdots, k-t,\ i\neq j)$$

ここで $k-t\geqq 1$ であり，t は，$1\leqq t\leqq k-1$ である．最短直線分の和 σ を求めると(5)式で示される．

$$\sigma=\sum_{t=1}^{k-1} R_t=R_1+R_2+\cdots+R_{k-1} \quad (5)$$

σ は，k 本の線分の最短の端点を順次結んだものであり，線分をすべて結んだときの線分間移動距離の合計を示す．この操作により1つの軌跡が得られるが，最短軌跡とはならない．このため最短軌跡を求めるには線分の結合の順序を考慮しなければならない．つまり，σ を求める式で R_2 は R_1 に，R_3 は R_1 と R_2 の影響を受けるため最短の端点をもつ直線のみを結合していくだけでは，高性能な近似解は求められない．したがって，すべての組合せ順序について調べる必要がある．図形 G 全体では，その組合せ回数 M は(6)式で与えられる．

$$M=2^{k-1}\cdot k! \quad (6)$$

M 回の組合せを調べれば，最短の描線軌跡が求められるはずであるが，線分数 k が多くなると莫大な計算量になりコンピュータによる処理でも容易に求めることは難しい．そこで，これらの組合せ回数を減らし高性能な近似解が求まる方法を探す必要がある．

2.2 連結組合せ数の減少化

組合せの回数を減らすために以下に述べる方法を行った．作図の形によらず構成される線分が他の線分の端点と重なる端点を持つ場合は，連続する1つの線分として扱い，端点数を減らすことができる．これは，端点の距離比較回数の減少となる．

k 本の直線が，幾つかの部分集合を作るとして，この部分集合 ζ_i が p 本の線分から構成されるとすると

$$g_{i1}\, g_{i2}\, g_{i3} \cdots g_{ip} \subset \zeta_i$$

となる．ζ_i が s 本あるとすると

$$\zeta_1\, \zeta_2\, \zeta_3 \cdots \zeta_s \subset G$$

で端点の数は $2s$ となり最短比較の組合せ回数 m は(7)式となる．

$$m=2^{s-1}\cdot s! \qquad (s\leqq k) \quad (7)$$

連結する線分が多いほど，$s<k$ となり，最短比較組合せは大幅に減少する．したがって，$m<M$ となる．比較組合せ数の比を(8)式に示す．

$$M/m=2^{k-s}\cdot(k!/s!) \quad (8)$$

作図データの入力にあたっては，以上に述べた方法で線分の連結状態を調べて最短軌跡のために必要な端点の数を再編成しておく処理が必要である．

2.3 端点間距離計算と最適化率

端点間の距離は，処理の効率を考えて，各座標値差の絶対値の最大値，最大値ノルムを用いる．最大値ノルム L_∞ を(9)式に示す．

$$L_\infty=\max(|x_{i1}-x_{j1}|, |x_{i2}-x_{j2}|, |x_{i3}-x_{j3}|) \quad (9)$$

軌跡が決まった後に行う実距離の計算については，処理時間短縮のため(10)式のユークリッド距離 L_2 を用いるのがよい．

$$L_2=d_{ij}=\{\sum|x_{ik}-x_{jk}|^3\}^{1/3} \quad (10)$$

最適化を行った結果，その評価法として最適化率 P を用いた．P の計算は，最適解を求めて行うのが良いが，多端点の場合，最適解を求めることは莫大な計算時間を要する．したがって入力データ軌跡のうち，線分間距離の合計 σ_{IN} と，最適化処理を施すことによって求まる線分間距離の合計 σ_{OUT} を用いて処理前と，処理後の比率を求め(11)式に示す P を最適化率として用いた．

$$P=\{(\sigma_{IN}-\sigma_{OUT})/\sigma_{IN}\}\cdot 100 \quad (11)$$

P は，最適解を用いる場合よりも小さい値となるが相対的な比較には，十分評価しうる値である．

3. クラスター化による描線の最適化

3.1 クラスター化を用いる意味

作図の描線最適化は，既に述べたように単純に最短な端点を結んで行くだけでは高性能な近似解を求めることはできない．このため作図自体を巨視的な区分

処理としてとらえる方法が必要である．クラスター化は，この要件を満たし，さらに，作図の特長がつかめること，端点の組合せの数が大幅に少なくなり処理効率が上がること，作図データの与えかたに制限がなく，2次元平面に限らず3次元空間の作図へも適用できること，などの長所を有している．

クラスター化手法は，種々開発されているが，本系に適用する場合，どのクラスター化手法が最も適合するかを処理効率を中心に検討する必要がある．クラスター化手法について検討する前に，処理効率を左右するクラスター化による端点比較数の減少について検討する．

3.2 クラスター化による端点比較数の減少

n 個の端点を q 個のクラスターに分けると，クラスター内とクラスター間での最短距離を求めるための端点間の比較回数，S_W, S_B は次のようになる．ここで，各クラスターはそれぞれ a_j 個 $(j=1,2,\cdots,q)$ の端点からなるとし，各線分の両端点は，同一のクラスターに属するとし，各クラスターは，それぞれ b_j 本 $(j=1,2,\cdots,q)$ の線分からなるとすると，(12)式の関係が成り立つ．

$$n=\sum_{j=1}^{q} a_j=2\sum_{j=1}^{q} b_j \quad (1\leqq b_j\leqq n/2-q+1) \tag{12}$$

クラスター内に b_j $(\geqq 2)$ 本の線分があるとすると，その中の2本が結合する線分間組合せの数 Γ は，(13)式で与えられる．

$$\Gamma=\sum k=(b_j-1)+(b_j-2)+\cdots+1 \tag{13}$$

2線分間の組合せは4通りあり，b_j 本の線分のうちいずれかの2本が連結して (b_j-1) 本になるための結合組合せの数は，4Γ となる．b_j が1本になるまで繰り返すと，(14)式が与えられる．

$$4\sum_{j=2}^{b_j}\Gamma=4\sum_{j=2}^{b_j}\sum_{k=1}^{j-1} k \tag{14}$$

クラスターが q 個あると，クラスター内の組合せ総数 S_W は，(15)で与えられる．

$$S_W=4\sum_{i=1}^{q}\sum_{j=2}^{b_j}\sum_{k=1}^{j-1} k \tag{15}$$

つぎに，クラスター間の組合せの数 S_B は，q 個のクラスターを結合していくための(14)式の b_j を q に置き換えた式(16)で与えられる．

$$S_B=4\sum_{i=2}^{q}\sum_{k=1}^{i-1} k \tag{16}$$

全体の組合せ数 S_T は(17)式で与えられる．

$$S_T=S_W+S_B \tag{17}$$

S_W, S_B は，q の値の増加によりそれぞれ減少，増加する．したがって，S_T が最小となる q_{min} の値が存在することがわかる．n の値を与えて S_T が最小となる q_{min} を求められれば，効率的な処理ができる．一般に n 個の端点を持つ作図を，均等な端点をもつ q_{min} 個のクラスターに分けることはできない．しかし，各クラスター内の端点の数は均等でなくても，作図自体を q_{min} 個に近いクラスターに分けられれば，q の値の増大による組合せ数 S_W の減少傾向と S_B の増加傾向を利用して S_T の最小値を求めることができる．

クラスター化手法にもよるが，クラスター数を指定してクラスター化を行うことは可能である[4),5)]．**図1**は $n=150$ のデータについて q 個のクラスター内の端点数をいろいろ変化させてシミュレーションを行った結果である．各クラスターにおける組合せ数の分散は，クラスター数が少ないほど大きく，クラスター数が増加するほど小さくなる．そこで，**図2**に端点数 n とクラスター数 q を S_T が最小になる場合についてシミュレートした結果を示した．(17)式を最小にする n と q の関係は(18)式を用いて計算した．

$$R=\min(S_T)=\min\left[4\left(\sum_{i=1}^{q}\sum_{j=2}^{b_i}\sum_{k=1}^{j-1} k+\sum_{i=2}^{q}\sum_{k=1}^{i-1} k\right)\right] \tag{18}$$

それぞれのクラスターに含まれる端点数 a_j は一定として計算している．したがって，クラスター内の端点数が，比較的に変動しない値を持つ作図データには適用できる．これは，あくまでもクラスターをいくつにしたら計算処理の効率が上がるかを考える目安として用いるもので，すべての作図データに適用できるものではない．図2から 1000 個の端点をもつ作図の場合，およそ 25 のクラスターがあれば，組合せ回数が最小となることがわかった．端点が作図全体に均等な分散をもち，$a_j\fallingdotseq n/q$ となるデータの場合は，図2に示す n と q の関係は極めて有効である．

3.3 クラスター分類における作図データの扱い

一般に作図データは，均等な端点のちらばりを持つものから偏りの大きなものまで多様なデータがある．作図をクラスター化する場合，次の2方法がある．

Ⅰ．**図3**に示すように，端点をそれぞればらばらにしてクラスター化する．

Ⅱ．2端点を1組のデータとしてクラスター化する．

Ⅰの方法は，2つの端点が，同じクラスターに含まれる場合，そのクラスター内で最短結合をすることができるが，含まれない場合は，他方の端点が属するク

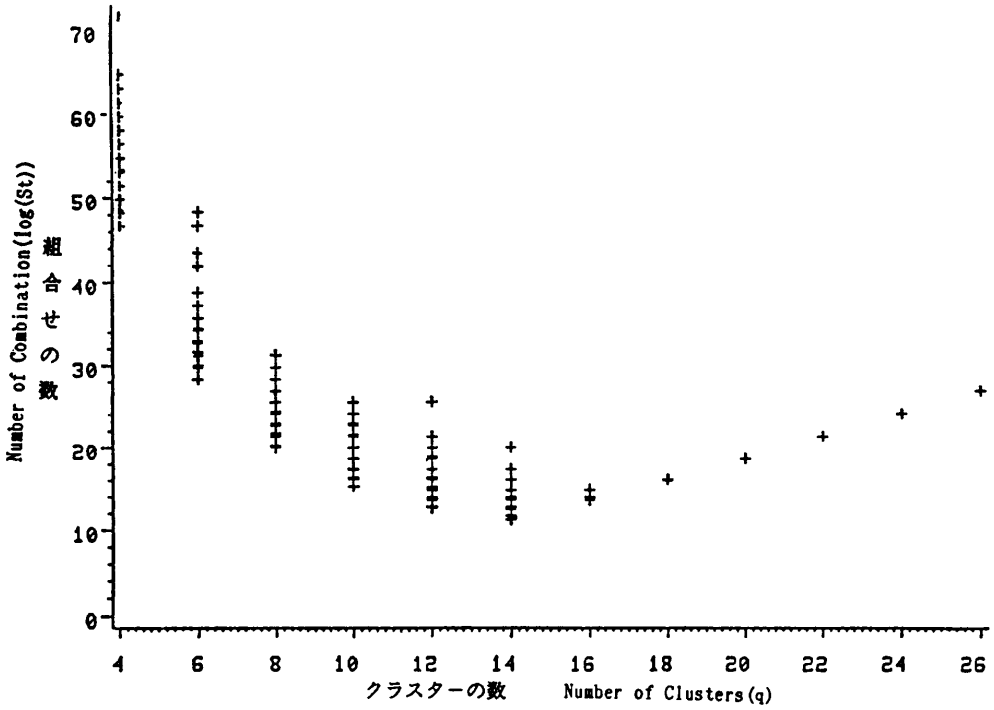


図 1 クラスターの数と組合せの数の関係
クラスター内の端点の数をいろいろ変えてシミュレーション
を行っている（端点の総数＝150）

Fig. 1 Relationship between the number of clusters and combinations (Num. of end points n=150) simulated results using various kinds of cluster.

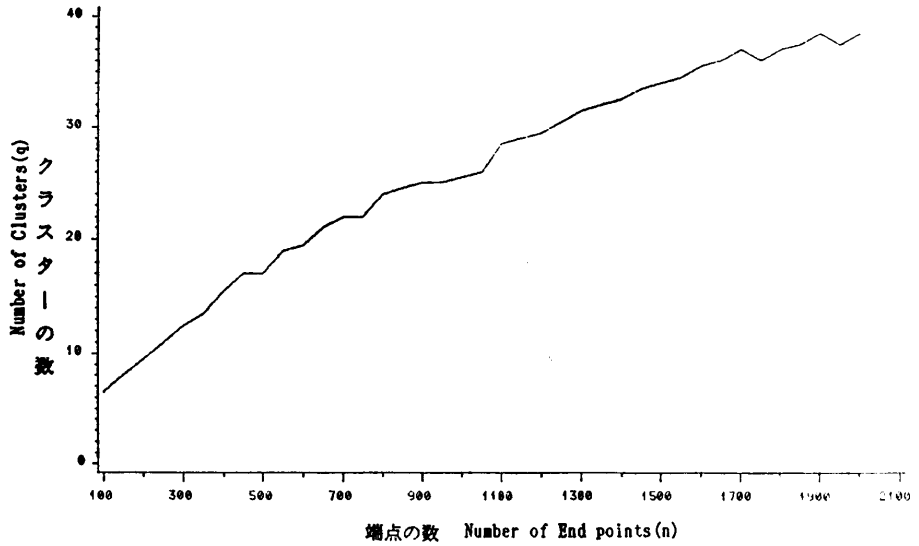


図 2 組合せ数が最小となる端点の数とクラスターの関係
（各クラスター内の端点の数は一定としている）

Fig. 2 Relationship between the number of end points minimizing the combination (the number of end points in each cluster is fixed).

ラスター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る．したがって，1回目のクラスター化においては，結合を行う端点は，両端点が共に同じクラスターにあるものについて結合を行い，残った端点について，2回目のクラスター化を行う，順次端点が2つになるまでクラスター化を行っていく．筆者の提案する方法は，このようにクラスター化を段階的に行っていくものである．その過程を図4に示した．

段階的クラスター化は，Iの方法を用いる．Iは，線分の2端点の分散の大小に関係なく，あらゆる作図データに適用できる．例えば，2端点の座標値が同じ点データでも，2端点の距離が長い線分データにも適用できる．

IIの方法は，線分のクラスター化になり，端点同士の距離が短い線分からなる作図データに対しては有効である．しかし，ほとんどの作図データは，IIでクラスターに分けることは難しく，以後はIの方法について検討した．

4. クラスター分類法の比較

クラスター分類法を段階的に行う場合，どの分類法を用いるのが効率的によいかを検討した[4),5)]．作図データの構造から非階層的な手法が有効であり，その中でも，重複を許すクラスター化法を作図データに適用した段階的クラスター化と，クラスターの数をあらかじめ指定して行う分割最適化法，およびモード探索型方法の3つの手法について検討を行った．

4.1 段階的クラスター化法を用いた場合

この方法は，作図が複数の線分から構成されるため，各2端点が異なるクラスターに所属することを許す重複のあるクラスタリング法として考えてい

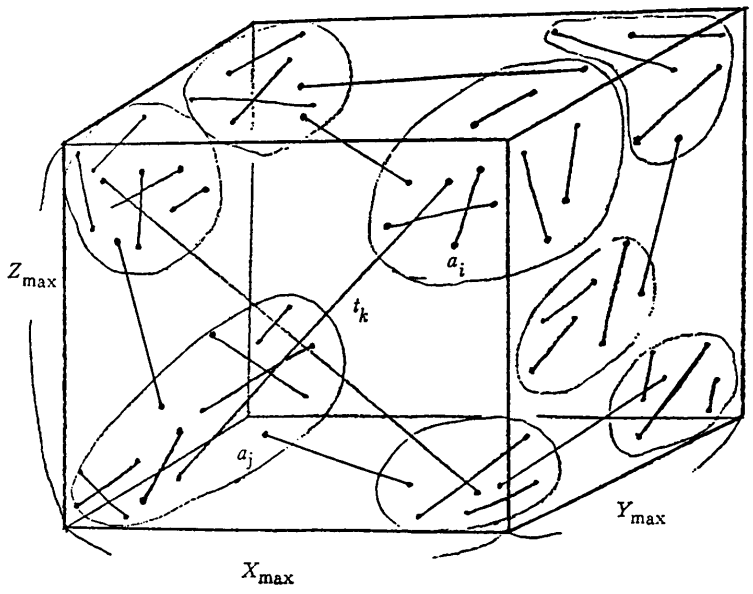
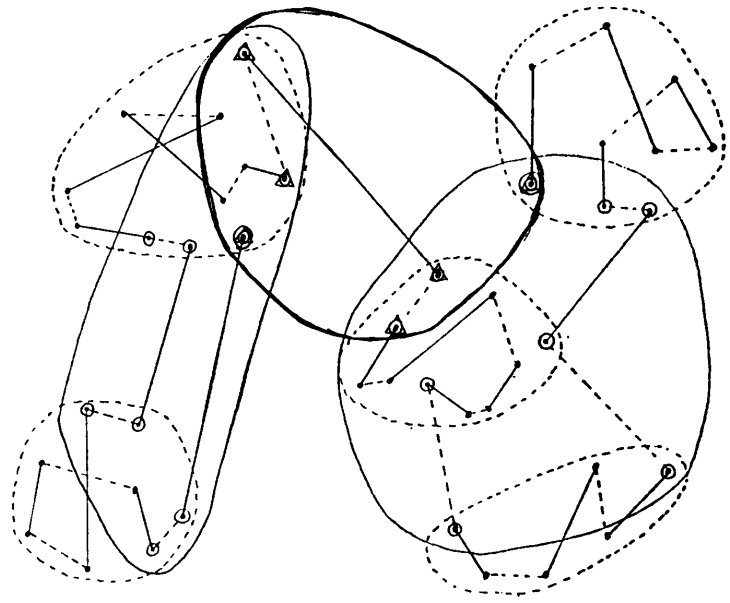


図3 3次元空間における線分データのクラスター化
Fig. 3 Cluster structure in three-dimensions.



• 第1段階クラスターで結合される端点
◎ 第2段階クラスターで結合される端点
△ 第3段階クラスターで結合される端点
◎ 最終に残る2端点

第1段階クラスター
第2段階クラスター
第3段階クラスター

図4 段階的クラスター化の例
3段階のクラスター化で端点が順次結合していく状況を示している
Fig. 4 Example of gradual clusterization in two-dimensional plane.

くものである．重複を許すクラスター化法は，一般には組合せ的要素が強いので大量のデータを処理するのに向かないとされているが，ここでは新しいアプローチを行い，大量データ処理を可能とした．すなわちグループのとらえ方として領域内を同じ大きさのセルで区切っておき，各セルの中に存在する端点を調べてクラスター化を行う．

以下，3次元空間に線分データがある場合の手順を示す．

1) 3次元空間に n 個の端点があり，その1端点 p_i の座標値を $X=(x_i, y_i, z_i)$ $(i=1, 2, \cdots, n)$ とする時

$$x_{max}=\max\{x_i\} \quad (i=1, 2, \cdots, n)$$
$$y_{max}=\max\{y_i\} \quad (i=1, 2, \cdots, n)$$
$$z_{max}=\max\{z_i\} \quad (i=1, 2, \cdots, n)$$

を求める．ここで，$x_i \geqq 0$, $y_i \geqq 0$, $z_i \geqq 0$

2) 次に，x_{max}, y_{max}, z_{max} で囲まれた3次元空間を1辺として h の立方体のセルで分ける．p_i は h^3 の立方体のセルのいずれかに所属する．

p_i が，どのセルに属するかを示す関数として，$c_i(\alpha, \beta, \gamma)$ を与える．α, β, γ は，3次元空間におけるセルの番号を示す．セルの長さ h は，(19)式で与える．

$$h=(x_{max} \cdot y_{max} \cdot z_{max}/\lambda \cdot n)^{1/3} \tag{19}$$

ここで λ は，パラメータ

3) n 個の端点をクラスターに分離させるのに次の手順で行う．

p_i が k 番目のクラスターに属し，p_j の属性 $c_j(\alpha', \beta', \gamma')$ が次の3つの条件を満たすならば，c_j は c_i と同じクラスターに属すとする．

条件 $\alpha-1 \leqq \alpha' \leqq \alpha+1$
$\beta-1 \leqq \beta' \leqq \beta+1$
$\gamma-1 \leqq \gamma' \leqq \gamma+1$

条件を満たさない場合は，p_j は異なるクラスターに帰属すると見なす．

4) 次に，n 個の端点が q 個のクラスターに分かれたとする．k 番目のクラスターの中に，n_k 個の端点があるとすると，n_k 個のうち，クラスター内で pair をなす端点 n_{pk} 個（線分の数は，$n_{pk}/2$）と，pair をなさない端点 n_{lk} 個とがあるとする．k 番目のクラスターについて端点数の関係は，(20)式となる．(20)式から(21)が得られる．

$$n_k=n_{pk}+n_{lk} \tag{20}$$
$$\sum n_k=\sum n_{pj}+\sum n_{lj}=n \tag{21}$$

クラスターごとに pair をなす端点の数 N_p, pair をなさない端点の数を N_l とすると，

$$N_p=(n_{p1}, n_{p2}, \cdots, n_{pq}) \tag{22}$$
$$N_l=(n_{l1}, n_{l2}, \cdots, n_{lq}) \tag{23}$$

N_l について，各クラスターごとの関係を行列 η であらわすと(24)式となる．ここで η は対称行列．

$$\eta=[c_{ij}] \tag{24}$$
$$(i=1, 2, \cdots, q \quad j=1, 2, \cdots, q)$$

次に，各クラスター内で pair をなす線分を総あたりで調べ最短結合する．結合後に残る端点の数を N_p', N_l' は，(25)，(26)式で与えられる．

$$N_p'=(2, 2, \cdots, 2) \tag{25}$$
$$N_l'=(n_{l1}, n_{l2}, \cdots, n_{lq}) \tag{26}$$

(25)，(26)式から，残る端点 n' は(27)式となる．

$$n'=2q+\sum n_{lj} \tag{27}$$

n' を1)の n とし，x_{max}, y_{max}, z_{max} を新たに求め，繰り返す．n' が減少し，n' 個を総あたりで距離比較を行っても効率的に影響のないところまで行う．つまり，クラスター間結合の総あたりで十分行える値 $n'=\varepsilon$ になるまで，何回も1)から繰り返す．

ここで，ε は，クラスター間組合せを開始する端点数に相当し，小さいほど良いが，経験的に 10〜20 を用いて行うのが計算効率を考えるとよいことがわかった．

クラスター化を行うのに要する時間は，$\lambda \cdot n$ に比例する．セルの大きさ h がクラスターの探索に影響するのは当然である．h が非常に小さければ，それぞれのセルに含まれる端点は少なくなる．また h が大きすぎるとクラスターの分離が起こらなくなる場合がある．

セルの探索は，一度調べたセルは省くことができる．探索の順序は，端のセルから始めてそのまわりのセルについて端点の含まれるセルを結合していく方法をとる．セル探索法は他にもあるが，この方法の利点は，探索の手間が最も少なくてすむこと，同心円的なクラスター探索を行うものでないため作図内に存在する種々の形のクラスターをとらえることができることにある．

h を左右するパラメータ λ について，種々の値を用いて計算を行い最適化率との関係を調べた結果を図5に示す．多くの作図データについて計算を行ったが，ここでは，5種類の3次元データについて，λ を2以上の値で変化させて計算した結果であるが，いずれも最適化率が安定した値を示した．

処理効率を左右する端点の組合せ数については，クラスター内の組合せ回数は，クラスター数を多くして各クラスター内に含まれる端点数を少なくすれば，小

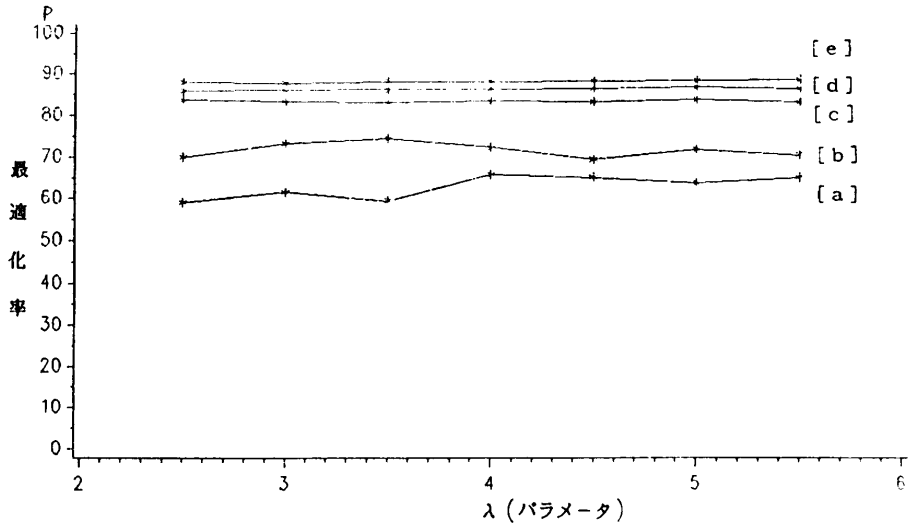


図 5 段階的クラスター化を行った場合のパラメータ λ と最適化率の関係
端点の数 [a]=50 [b]=100 [c]=500 [d]=1000 [e]=1500 3次元データ

Fig. 5 Relationship between parameter (λ) and the rate of optimization by gradual clusterization.
Number of end points [a]=50 [b]=100 [c]=500 [d]=1000 [e]=1500 3-D.

さくなる．したがって，クラスターを何回行うかの段階数と，各段階におけるクラスター内組合せ回数の合計が問題になる．λ の値を2以上にしておくと，数回の段階化で端点数が 10 個以内に減少することが種々の作図データを用いて行った計算機実験の結果からわかった．したがって，λ は $n/2 \geqq \lambda \geqq 2$ の範囲で用いるのがよい．

4.2 分割最適化法を用いた場合

この算法の代表的手法として，k-means 法がある[6]．これは，クラスター内平方和を最小化する典型的方法である．始めから q 個のクラスターがあるとして計算を開始し，反復計算によってクラスター基準の最適化をはかる方法である．

T は全データから得られた総平方和・積和行列，B はクラスター間平方和・積和行列，W はクラスター内平方和・積和行列とすると，(28)式の関係がある[7]．

$$W = T - B \tag{28}$$

T の値は，データが与えられると確定するので，B と W の関係をどう定めるかにかかっている．一般に，$T = W + B$ のトレースまたは行列式を作ってそれらの組合せにより基準を設定することが多い．

トレース基準 $tr(W)$ を用いて

$$r^2 = 1 - tr(W)/tr(T) \tag{29}$$

とおくと，クラスター内平方和 $tr(W)$ の最小化は，r^2 を大きくすることである．また，これを求める方法として山登り法による最適化や，平方和をクラスター内平方ユークリッド距離の総和と置き換えこの基準を2次計画法により最適化するなど種々の方法がある．解は，一般に局所最適解となるので1回の処理では分類結果の解釈が難しい．しかし作図データの場合は，2次元平面上にクラスター化の状況を表示して調べることによって，最適な結果を求めることができる．計算は，SAS[8] のプロシジャー FASTCLUS を用いてクラスターの数 q を指定して行った．FASTCLUS でクラスター分けできた結果を，最短結合を行うプログラムで受けて処理を行った．

図 6 は，端点数 50 と 500 の3次元データを用いた最適化率と q の関係を示した．クラスター基準の最適化を計る反復計算は行っていない．図6から，クラスター数 q を変化させても最適化率 P は一定の値を示していることがわかる．

この方法は，最初に設定した q の数でクラスターに分け処理するため，クラスター内とクラスター間の最短組合せの総数は，q の数によって大幅に変化する．FASTCLUS を実行するのに要する時間は，一般に次のとおりである[8]．

n：端点の総数
p：次元の数（2次元では2，3次元では3）
q：クラスターの数
w：初期核置き換えの数
v：反復計算の回数

とすると，必要とする時間は，$npqv$ に比例する．初期

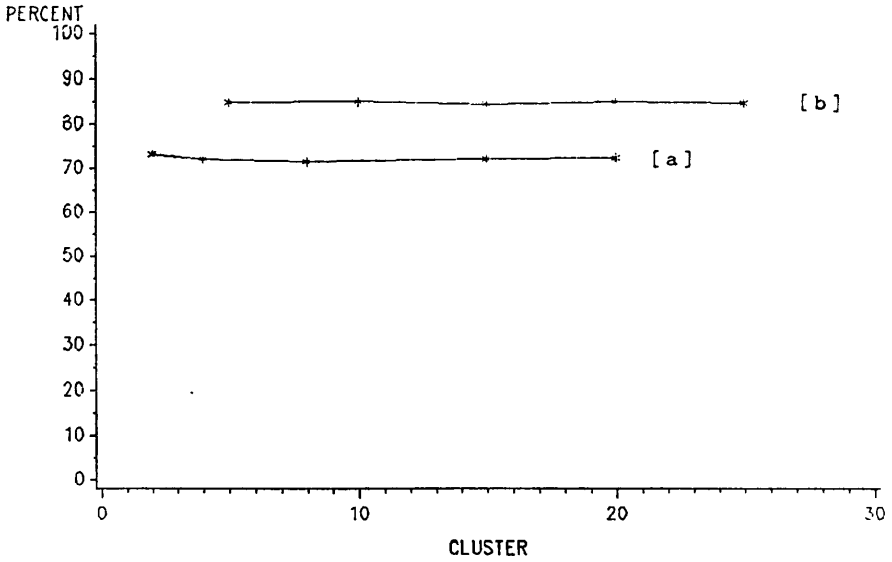


図 6 分割最適化法によるクラスターの数 q と最適化率の関係
端点の数 [a]=50 [b]=500 のデータ，共に3次元データ
Fig. 6 Relationship between the number of clusters (q) and the rate of optimization.
[a]: Number of end points=50, [b]:=500, three-dimensions

核選択に要する時間 t は，(30)式で与えられる．

$$t=npq+(w+q)pq^2/2 \tag{30}$$

核の置き換えを行わないならば，t は(31)式で示す範囲の値をとる．

$$pq^2 \leqq t \leqq npq \tag{31}$$

(30)式からクラスターの数 q が小さいほどクラスターを決定するための FASTCLUS の実行時間は少なくなる．したがって，図2の関係と，FASTCLUS の実行時間を合わせて q の値を決定するのが良いことがわかった．

4.3 モード探索型方法を用いた場合

このアルゴリズムは，ある端点についてその近傍をとり，その領域内に含まれる端点数を数えて1つのグループとしグループ内の端点の数をグループの密度とする．これをすべてのグループについて求めてその分布状態をみて密度の濃淡に分ける．濃い部分を仮のクラスターとして残し淡い部分を最配置してクラスターの所属を決める方法である[4]．

作図データについて，この方法を用いてクラスター化することは，1) 端点をある領域でグループ化し，端点の数をそのグループ密度とする．2) その他の領域との密度の差を比較してクラスターを求めることを行うこと．を意味する．作図データの構造から最初に決める領域の設定が悪い場合は，密度分布の濃淡部を分離するまで何度も繰り返し計算を行わなければならない．したがって計算が複雑となる．また，仮に，分布がクラスターの分離をうまく表示できたとしても，領域の設定の方法が，ある点を中心に同心円を描く方法で端点を探していくため，クラスターが円形に近い形を持つものでない場合は，クラスター分類がうまくいかない場合が生じるなど，いくつかの問題点がある．

その1つは，クラスターの半径の定め方である．また，密度値は，できるだけ滑らかに変化することが望ましいが，作図データの場合は，かなり大量のデータでも滑らかにならないのが現状である．さらに，密度値が同じ値のものがないほうがよいが，実際にはそうはならない．

既に述べたように，ここで扱う作図データ自身が以上の問題点を解決できる性質のものでないことから，モード探索型方法は，目的とする作図データのクラスター化を行うメリットがないという結論を得た．

5. 各方法による結果と検討

5.1 段階的クラスター化による結果

表1は，乱数を使って座標値を求めた $n=1000$ の3次元データを用いて段階的クラスター化を行った結果である．パラメータ λ は，2から5を用いて計算を行った．パラメータ λ の値と，最適化率 P, CPU 時間，軌跡が決定されるまでの段階の数とそれぞれのクラス

表 1 段階的クラスター化を用いた描線順序最適化の結果
Table 1 Computed results of optimization of drawing sequence by gradual clusterization method.

パラメータ λ	最適化率 %	CPU 時間 sec/100	クラスターの段階数と各クラスターの数
2	86.53	3.35	2(100/14)
3	86.25	3.09	2(135/18)
4	86.63	3.19	2(171/25)
5	86.91	3.14	3(210/36/11)

〔端点の数＝1000 の 3 次元データ〕
使用計算機：HITAC-M 280 D

表 2 分割最適化法を用いた描線順序最適化の結果
Table 2 Computed results of optimization of drawing sequence by k-means method.

クラスターの数 q	最適化率 %	CPU 時間 sec/100	FASTCLUS 処理時間
2	88.07	80.01	7
5	87.76	23.63	7
8	87.61	13.12	7
12	87.73	9.04	11
15	87.54	7.30	12
18	87.44	6.15	14
20	87.54	5.81	16
25	87.31	4.85	16
30	87.38	4.32	17
35	87.23	3.96	20
40	87.68	3.64	24
50	87.33	3.41	24
100	87.32	3.34	42
150	87.60	4.39	57

〔端点の数＝1000 の 3 次元データ〕
使用計算機：HITAC-M 280 D

ター数を示している．

P は，いずれも 86% 以上の高い値を示している．さらに，CPU 時間は，5.2 節で述べる分割最適化法に比べて，いずれも 3.4(sec/100) 以下で効率が良いことを示している．**図 7 は**，3 次元の最適化結果を x-y, y-z, z-x の各平面図に分けて示している．左側は，入力データの作図の実軌跡を描くために空間移動した軌跡，右側は，最適化を行った後の同様な空間移動の軌跡を示す．最適化率は，86.91% であり，かなり最適化された結果を図から読み取ることができた．

5.2 分割最適化法による結果

図 8 は，5.1 節と同じデータを，k-means 法を用いて，クラスターの数 q=5 (出力結果 [A]) と 15 (出力結果[B]) の 2 通りについてクラスター分類を行った結果である．[A] [B] 共に SAS の FASTCLUS プロシジャーを用いて各端点がどのクラスターに属するかを決定している．

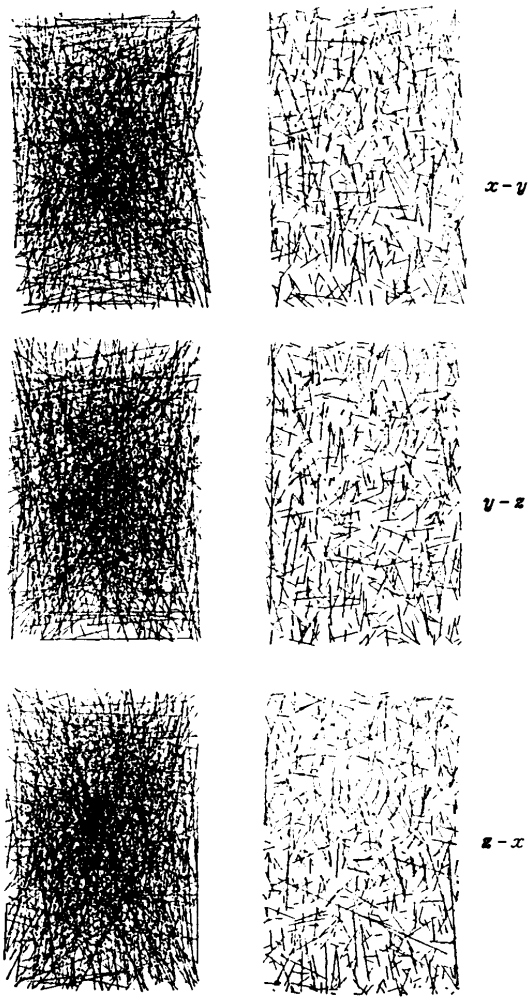


図 7 段階的クラスター化による 3 次元データの最適化前と最適化後の比較
x-y, x-z, z-x 平面図
[n=1000, 最適化率=86.91, λ=5, CPU 時間=3.14 (sec/100)]
Fig. 7 Comparison of three dimensional data optimized and not optimized n=1000, P=86.91, λ=5, CPU-time=3.14 (sec/100)

FASTCLUS プロシジャーの結果を入力データとしてクラスター内とクラスター間での最短結合を行うプログラムで処理した結果が**表 2** である．q=2 から 150 まで，クラスター数を変化させて，最適化率 P と CPU 時間の関係を求めた．FASTCLUS の処理時間は，アプリケーションプログラムのため，クラスター化のみについての CPU 時間を取れなかったが，相対的な処理時間の変化を表 2 に示す．クラスター数の増加で，P はわずかではあるが減少する傾向が見られるが，比

CLUSTER SUMMARY

CLUSTER NUMBER	FREQUENCY	RMS STD DEVIATION	MAXIMUM DISTANCE FROM SEED TO OBSERVATION	NEAREST CLUSTER	CENTROID DISTANCE
1	216	181.9	665.0	3	529.1
2	156	171.5	593.3	5	568.7
3	236	189.4	616.0	1	529.1
4	239	191.8	588.3	1	557.2
5	153	164.5	546.2	3	535.5

STATISTICS FOR VARIABLES

VARIABLE	TOTAL STD	WITHIN STD	R-SQUARED	RSQ/(1-RSQ)
SX	286.83135	159.40611	0.69238	2.25077
SY	284.47044	183.79831	0.58422	1.40510
SZ	291.54170	200.59354	0.52849	1.12085
OVER-ALL	287.62952	182.05301	0.60099	1.50619

FASTCLUS PROCEDURE

PSEUDO F STATISTIC = 374.66

APPROXIMATE EXPECTED OVER-ALL R-SQUARED = 0.66071

CUBIC CLUSTERING CRITERION = -10.307

[A]

CLUSTER SUMMARY

CLUSTER NUMBER	FREQUENCY	RMS STD DEVIATION	MAXIMUM DISTANCE FROM SEED TO OBSERVATION	NEAREST CLUSTER	CENTROID DISTANCE
1	40	105.7	303.9	13	364.6
2	67	123.5	420.5	10	350.1
3	54	117.4	356.3	6	381.5
4	66	116.6	295.7	11	372.4
5	66	114.1	348.4	6	363.2
6	104	129.8	389.6	5	363.2
7	67	123.1	345.1	8	348.9
8	63	116.8	336.6	7	348.9
9	50	131.5	367.6	12	360.1
10	59	113.8	396.0	2	350.1
11	83	124.6	365.0	4	372.4
12	78	123.5	348.0	9	360.1
13	57	116.3	347.1	14	320.9
14	77	120.5	330.9	13	320.9
15	69	119.7	326.9	13	351.0

FASTCLUS PROCEDURE

STATISTICS FOR VARIABLES

VARIABLE	TOTAL STD	WITHIN STD	R-SQUARED	RSQ/(1-RSQ)
SX	286.83135	125.65705	0.81077	4.28456
SY	284.47044	117.14480	0.83280	4.98078
SZ	291.54170	119.32503	0.83483	5.05435
OVER-ALL	287.62952	120.76294	0.82619	4.75346

PSEUDO F STATISTIC = 334.44

APPROXIMATE EXPECTED OVER-ALL R-SQUARED = 0.84027

CUBIC CLUSTERING CRITERION = -4.026

[B]

図 8 FASTCLUS によるクラスター化の結果
端点の数 n=1000, クラスター数 q[A]=5 [B]=15

Fig. 8 Results of clusterization using FASTCLUS.
Num. of end points n=1000, Num. of clusters q[A]=5 [B]=15

較的に安定した値となっている．一方，CPU 時間は，最短結合に掛かる時間は，クラスター数の増加で，大きく減少しているが，FASTCLUS の処理に掛かる時間が増加している．FASTCLUS の処理時間が相対的時間であり，最小の処理時間となる q の値は，表2の結果からは正確に述べられない．しかし，図2の組合せ数が最小となる端点とクラスターの関係を示すシミュレーション結果から，端点数 n が 1000 の場合は，クラスター数 q は 25 であることがわかっている．これは，表2の結果ともほぼ一致する．

5.3 各方法の比較と検討

5.1 節，5.2 節の結果からいずれの方法も最適化率は高い値を示している．処理効率については，段階的クラスター化が優れていることがわかった．

処理の手順は，分割最適化法による場合は，FASTCLUS と最短結合の処理と2段階で行うため，複雑であるが，段階的クラスター化については，一連の処理で行うことができる．

処理に必要なパラメータの指定は，分割最適化法は，クラスター数を与えて計算を行う．クラスター数の与え方によっては，最短結合のためのクラスター内，クラスター間組合せ数の和が大きくなる場合が生ずる．段階的クラスター化による方法は，構成する線分の端点間距離の分散の傾向をみて，λ の値をダイナミックに変化できる．したがって，どのような作図データにも適用できる．さらに，効率的にも優れてい

る.

以上のことから，筆者は，段階的クラスター化による方法が優れている結論を得た.

6. むすび

作図データをクラスター化するため種々のクラスター分類法について比較検討してきたが，筆者の重複を許すクラスター化法を段階的に行うクラスター化の方法が，1) 高性能な近似解が得られること，2) 処理効率処理手順の点で優れていること，3) データ構造によらず適用できることを明らかにした.

既に行ってきた描線の最適化は，線分を結ぶための最短の軌跡を求めることに終始してきた．実際には，線分と点，点のみからなるデータにも応用できる．今後は，線分から線分への移動に制限のある場合など，2次元，3次元データについて種々の条件のもとで軌跡の最適化を行う必要がある.

描線の最適化は，線分を3次元空間に拡張することによって2次元平面におけるよりも，より多くの問題とその応用が考えられる．例えば，シーケンスロボットに最適な腕の軌跡を教えておく場合や，数値制御が行える NC ロボットの NC データの作成には，正にそれを必要とすると言える.

謝辞 貴重なご意見をいただいた恩師である東北大学名誉教授，日本事務器株式会社会長の田中信行先生，統計数理研究所の水野欽司教授および拓殖大学情報工学科の竹谷誠助教授，上智大学機械工学科の浅野孝夫助教授に深く感謝いたします.

参考文献

1) Iri, M., Murota, K. and Matsui, S.: An Approximate Solution for the Problem of Optimizing the Plotter Pen Movement, *Lecture Notes in Control and Information Sciences 38* (Proceeding of the 10th IFIP Conference on System Modeling and Optimization, New York, 1981), Springer-Verlag, Berlin (1982).
2) Iri, M., Murota, K. and Matsui, S.: Linear-Time Heuristics for the Minimum-Weight Perfect Matching on a Plane with an Application to the Plotter Algorithm, Research Memorandum RMI 81-07, University of Tokyo (1981).
3) Leipälä, T. and Nevalainen, O.: A Plotter Sequencing System, *Comput. J.*, Vol. 22, pp. 313-316 (1979).
4) 大隅 昇：クラスター分析はどう使われるか，数理科学，No. 190, April (1979).
5) MacQueen, J. B.: Some Methods for Classification and Analysis of Multi-variate Observations, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, pp. 281-297 (1967).
6) Anderberg, M. R.: *Cluster Analysis for Applications*, Academic Press, New York, San Francisco, London (1973).
7) Warren, S. Sarle: Cubic Clustering Criterion, SAS Technical Report A-108 SAS Institute Inc. (1983).
8) SAS USER'S GUIDE: Statistics 1982 Edition, SAS Institute Inc. (1982).

(昭和62年3月31日受付)
(昭和62年9月 9 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.