骨格ベクトル文字からアウトライン文字への変換方式†

下位憲司†† 上原徹三††

計算機上で高ドットで高品質の文字を保持する代表的な方法として，文字の輪郭線で表現する輪郭ベクトル文字（アウトライン文字）と，文字を構成するストロークを骨格情報で表現する骨格ベクトル文字がある．この骨格ベクトル文字の生成途中で作られるストロークごとの輪郭線をポリゴン近似したデータから，図形的処理を行うことにより，輪郭ベクトル文字として必要な情報である外輪郭線情報および内輪郭線情報を生成するプログラムを開発した．その方法は，まず外輪郭線情報を，文字や図形認識の分野で用いられている方法で追跡することにより求めた．内輪郭線情報を求める方法は，ストロークごとのポリゴンデータ（座標値）とそれらの交点座標値から，外輪郭線と辺を共有しない最小のポリゴンをすべて検出し，それらのポリゴンがストロークの中に包含されているか否かを判定し，包含されていない場合にそのポリゴンを内輪郭線情報とする方法である．大型計算機（M-680 H）による変換処理時間は，256×256 ドットの JIS 第1水準の漢字の平均値で，66 ms であった．また，骨格ベクトル文字情報からストローク輪郭の生成時間は，平均 33 ms である．文字の太さの変更等が可能である骨格ベクトル文字の特長を利用して，文字を生成し，品質を劣化させることなく，輪郭ベクトル文字に変換することが可能になった．

1. まえがき

計算機上で文字を表現する場合，ドット行列として表現することが一般的である．文字形状の可視化方式とメモリ上の保持方式とは，同一である必要はないが，従来，保持方式もドット行列であることが多かった．そのため，多くの文字を多種類，多サイズ分ドット行列で保持しようとすると，膨大なメモリ容量が必要になる．また，ドット文字は拡大や縮小による品質の劣化が著しいので，最近の文字サイズや書体の多様化や文字形状の高品質化の要求の高まりに応えるためには，サイズごとに作成する必要がある．さらに，文字形状の高品質化のためプリンタの解像度は向上しており，同じサイズの文字の出力にも高ドットが必要になってきた．このような情勢は，ドット行列方式に代わる新しい文字保持方式を要請している．

ドット文字が拡大，縮小等の変形に弱いのは，独立なドットの集まりのみを保持し，構造的な情報を保持しないためである．そこで，文字を図形情報として保持することにより，上記問題を解決しようとする試みが，英文字で実用化された．文字を表現する方法として，文字の外輪郭線および内輪郭線を図形として保持して，必要ならば，出力時にドット行列に展開して文字を生成する輪郭ベクトル文字（アウトライン文字とも言われている）が脚光をあびている[1)~4)]．この方式の有利な点は，文字の拡大，縮小が比較的容易であるため，一つの文字に一種類の情報だけ保持していれば良く，文字を格納するメモリ容量が少なくて済む点である．

ところで，文字に関する最近のニーズの拡大は，次のような要請を含んでいる．

（1） 同一の書体で各種サイズの出力を行いたい．文字デザインの経験則により，文字サイズをある程度以上大きくすると，ストロークの太さを文字サイズ比率以上に太くしたい．

（2） 基本書体のデザイン方針を変えず，太さや端辺形状の異なる新書体を作りたい．

このような要求に応えるため，文字を複数のストロークで構成し，かつ文字サイズに応じて太さの制御が可能な文字生成方式が考えられている．その方式は，文字の骨格点情報とストロークの種類や太さ等の情報を保持する骨格ベクトル文字である[5)~11)]．

この方式の文字の場合も，輪郭ベクトル文字とほぼ同じ容量で文字情報を格納可能である．しかし，文字を生成するのに必要な時間を比較すると，処理の性質上，輪郭ベクトル文字のほうが，骨格ベクトル文字よりかなり高速である．

また，レーザ・ビーム・プリンタを中心とする最近の高解像度の文書出力装置は，ドット文字とは別に輪郭ベクトル文字により，各種サイズの文字や変形文字（斜体や白抜き文字等）をサポートするものが標準になりつつあり，出力性能も向上してきている．そこで，上記（1），（2）で述べた骨格ベクトル文字の特長をいかして文字を生成し，輪郭ベクトル文字に変換で

† Conversion Method from Skeleton Character Pattern to Outline Character Pattern by KENJI SHIMOI and TETSUZOU UEHARA (Central Research Laboratory, Hitachi Ltd.).

†† (株)日立製作所中央研究所

きれば多様なニーズに応じた文字作成が可能となり，非常に有意義である．

本報告は，骨格ベクトル文字からストロークごとの輪郭を表すポリゴンデータを生成し，図形的処理を行うことにより，輪郭ベクトル文字として必要な情報である外輪郭線情報および内輪郭線情報に変換する方式についての報告である．

2. 骨格ベクトル文字生成方式

ストローク種別に基づく骨格ベクトル方式は，文字セットで定まる比較的少数の基本ストローク（エレメントとも言う）の組み合わせで文字形状が表現できるというレタリングの分野の知見にヒントを得ている[12]．この基本ストロークの分類をそのままコンピュータにおける文字パターンの保持に採用できる可能性がある．そうでなくとも，新たに設定したストロークの分類によって生成した形状の組み合わせでエレメントが表現できれば良い．このようなストロークの分類をストローク種別と呼ぶ．漢字のストローク種別としては，縦線，横線，左右の払い等がある．これらの種別ごとのストロークの形状を高品質に生成する方法があれば，文字内の各ストロークの配置の指定によって，文字形状を高品質に生成できるはずである．ストロークの太さや端辺形状については，その特徴を形状パラメータとして指定する．これらの情報によって，文字を構成する各ストロークの厚みを持った輪郭が生成できる．

より具体的には，同じ種別のストロークでも，それが用いられる文字ごとあるいは文字内の部分ごとに，大きさや形状が異なる．本方式では，ストロークの形状の大きさや位置を指定するために，当該ストロークの骨格点の位置を与える．ストロークを構成する骨格点の数は，ストローク種別ごとに定める．当該文字を構成するストロークに適当な順序（筆順に従うのが適当）を与えて番号を付ける．文字形状を作成する矩形の左下隅を原点とする直交座標系を考えて，ストロークに対して，ストローク種別と，ストロークを構成する骨格点座標列と，形状パラメータとを与える．ストロークの線の太さや端辺の角度等の輪郭形状は，個々の文字の，個々のストロークに対して与え得る形状パラメータによって指定する．しかし，一つのフォント内では文字のデザインは統一されるので，個々のストロークごとにその形状を指定するのは冗長である．そこでストローク種別ごとに形状パラメータを指定できるようにする．この2種の形状パラメータのうち，前者を局所的形状パラメータ，後者を全域的形状パラメータという．前者が指定されていれば，これを有効とし，指定されていなければ，後者を有効とする．

この方式によると，文字サイズの拡大縮小およびサイズと独立なストロークの太さ制御が可能である．ストローク形状を文字セット内で統一でき，特定種別のストロークの生成形状の変更により当該ストロークを含む全文字の形状の一様な変更が可能である[11]．そして，もし必要ならば，ストロークごとの輪郭線内を塗りつぶすことにより，ドット文字を生成することが可能である．

3. ベクトル文字変換方式

3.1 変換方式の比較

骨格ベクトル情報からドット文字を生成する途中で生成されるデータには，以下のようなものがある．

（1） ストロークごとの輪郭特徴点の座標群とそれらの点を結ぶ適当な曲線（直線，円弧，ベジェ曲線等）の種類

（2） （1）で得られた情報から，直線近似に変換した時の座標群（ポリゴンデータ）

（3） （2）のデータをドットイメージに変換し，ストローク内を塗りつぶした時の文字全体のドットデータ群

輪郭ベクトル文字に変換する際には，上記三つのデータ群のうち，いずれかのデータを入力として変換することになる．すなわち，次の三つの方法が考えられる．

- 輪郭特徴点とそれらの点を結ぶ曲線から変換する方法（輪郭点変換法）
- ポリゴンデータから変換する方法（ポリゴン変換法）
- ドットデータから変換する方法（ドット変換法）

以上の方法を比較したのが**表1**である．

ドット変換法は，文字や図形認識の分野で広く用いられている方法で，ドットレベルの追跡をすることにより，輪郭線情報を取得する方法である．この方法の特徴は，ドットイメージに変換してから処理するため，輪郭追跡して輪郭ベクトル情報を得るまでの時間は比較的小さいことである．しかし，骨格ベクトル情報からドットイメージに変換するまでの処理時間が大きいため，輪郭ベクトル情報を得るまでのトータルの処理時間が非常にかかる．また，ドットイメージに変

表 1 変換方式の比較
Table 1 Comparison among conversion methods.

方式	処理時間	変換時間	変換アルゴリズムの複雑さ	文字品質	保持すべきデータ量
輪郭点変換法	中	大	大	高	小
ポリゴン変換法	中	中	中	高	中
ドット変換法	大	小	小	低	大

処理時間とは，変換対象データの作成時間＋変換時間を意味する．

換する時の量子化誤差が，輪郭ベクトル情報にそのまま反映されるため，文字としての品質に欠ける場合がある．さらに，保持すべきデータ量の観点から見ると，直線部分を検出することにより，データ量を減少させたとしても，他の方式に比べると非常に大きい．

一方，輪郭点変換法については，ベジェやスプライン曲線同士または曲線と直線の交点を計算した後，その交点とストロークの輪郭特徴点を原形に忠実にベジェやスプライン曲線で再度近似するのは，非常に困難である．また，保持すべきデータ量は最も少なく有利であるが，精度の高い近似曲線を得るためには，変換に要する時間が多くなることが予想される．

以上の理由により，ポリゴン変換法により，骨格ベクトル文字から輪郭ベクトル文字に変換することにした．

3.2 変換方式

例として，“日”という文字を骨格ベクトル文字から輪郭ベクトル文字へ変換する過程を図1に示す．

図1(A)は，“日”という文字を骨格ベクトル文字の基本構成要素であるストロークで表現するには，2本縦線ストロークと3本の横線ストロークより構成されていることを示している．この図において，●印は，ストロークの骨格点，○印は輪郭特徴点を表す．輪郭特徴点とは，曲線種別（直線，スプライン曲線，ベジェ曲線，円，楕円等）に応じて，輪郭線を定義するのに必要な制御点のことである．なお，本例は，すべての輪郭特徴点間を直線で近似する場合の例である．

図1(B)は，ストロークを組み合わせて，“日”という字を表現した図である．▽印および△印は，ストローク同士の交点を示す．このうち▽印は，輪郭特徴点同士または，輪郭特徴点とストロークの交点を表し，△印は，それ以外のストローク同士の交点を表す．

図1(C)が，外輪郭線（実線）と内輪郭線（破線）で構成される輪郭ベクトル文字である．

3.2.1 変換概要

ポリゴン変換法は，各ストロークの輪郭線をポリゴン近似した情報を入力とし，文字全体の外輪郭線情報（ポリゴン座標群）と内輪郭線情報（ポリゴン座標群）を求めることである．その方法を概説すると以下のとおりである．ここで，本報告で扱う骨格ベクトル文字では，ストロークそれ自身では，内輪郭を含まないとする．以下，座標系は直交座標系で，座標値はすべて正数とする．

ステップ1：ストローク同士に交点があるか否かをチェックし，交点がない場合は，独立のストロークとして何も変換しないで，ポリゴン座標群をそのまま外輪郭

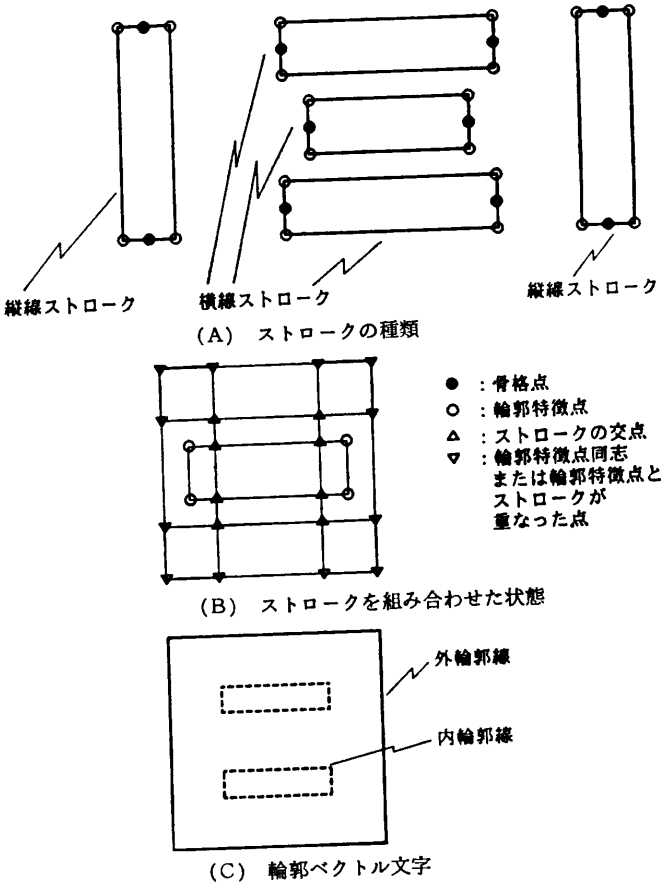


図1 骨格から輪郭への変換過程
Fig. 1 Converting process skeleton into outline.

線情報とする．交点がある場合は，その交点座標を記録し，相互に交点を有するストロークすべてをまとめてグループとし，グループごとに以下のステップ2，3を処理する．

ステップ2：グループ内のすべてのストロークの交点座標とポリゴン座標からなる点列のうち，任意の輪郭特徴点を起点として，一定方向にトレースし，分岐点があった場合，トレース方向を決定するため，各方向の単位ベクトルと基準ベクトル（直前のトレース方向）とのなす角度の大きい方向を選択する．この処理を起点に戻るまで繰り返すことにより，外輪郭線情報を生成する．

ステップ3：交点を起点とする最小の多角形をすべて抽出し，それらの多角形がストロークの内部に含まれているか否かを判定し，含まれていない場合を内輪郭線情報とする．

以上の処理を行うことにより，輪郭ベクトル文字を構成するのに必要な外輪郭線と内輪郭線情報が得られ，骨格ベクトル文字から輪郭ベクトル文字へ変換されたことになる．

3.2.2 外輪郭線情報の取得方法

上記ステップ2で述べた外輪郭線情報を取得する方法について，**図2**を用いて詳細に説明する．この図では，ストローク1（多角形 *DFGJ*）と2（多角形 *ABEI*）が重なりあって複数の交点（点 *C*, *H*: △印）を有する場合であり，この二つのストロークで一つのグループをなしている場合である．

まず，輪郭特徴点（○印）の中の任意の一点（例えば，*X* 座標値が最大で，*Y* 座標値が最小値の点 *A*）を起点として抽出する．なお，起点の決定方法は，輪郭特徴点であればどの点でも良い．点 *A* より上方向（*Y* 座標値の大きい方向）の輪郭に沿ってトレースする．最初の点 *B* は，交点ではないので線分 *AB* が外輪郭線情報の一部として確定する．次に，点 *B* から接続されている点は *C* だけであるので線分 *BC* も外輪郭線情報の一部として確定する．

次に点 *C* からは，三つの方向（点 *D*, *E*, *F* の方向）に分岐している．どの方向に分岐するかを決定する必要がある．外輪郭線をトレースしているのであるから，一番外側の点 *F* に分岐する必要がある．この点 *F* を選択するには，単位ベクトルの外積をとり，その符号を判定することにより可能である．つまり，ベクトル *BC* とベクトル *BD*，ベクトル *BC* とベクトル *BE*，およびベクトル *BC* とベクトル *BF* の各々の外積をとると，その符号は，負，ゼロ，正となる．この符号の意味は，基準ベクトル（この場合はベクトル *BC*）に対して，負の時は左方向，ゼロは同じ方向，正は右方向にあることである．この場合，外輪郭をトレースするのであるから，正の値（右方向）となった

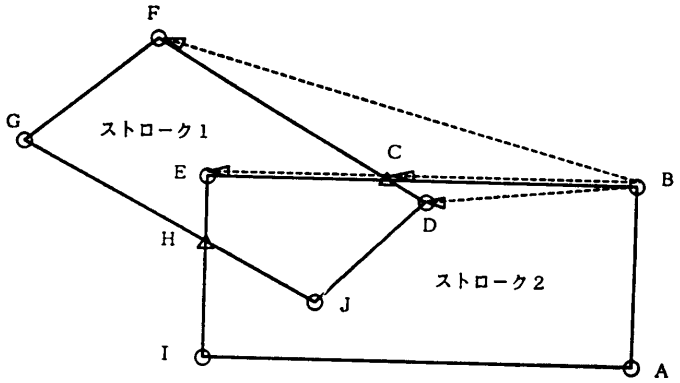


図2 外輪郭線追跡法
Fig. 2 Trace method for outer outline.

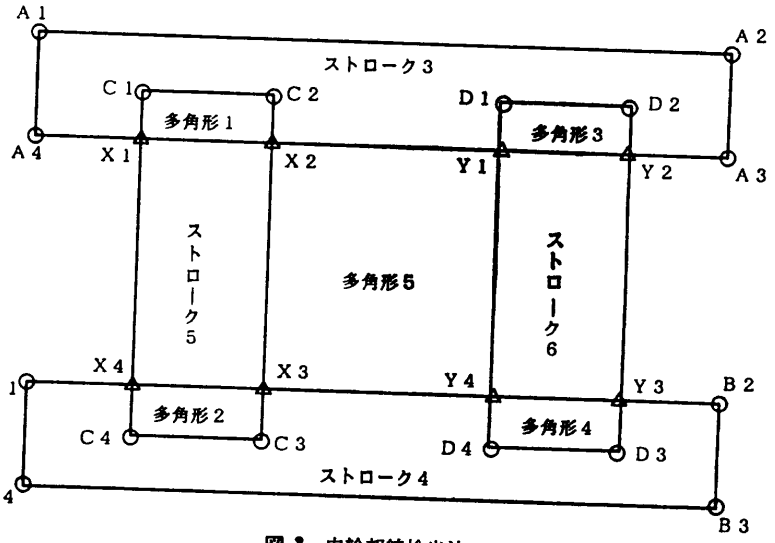


図3 内輪郭線検出法
Fig. 3 Detection method of inner outline.

ベクトル BF が選択され，点 C からは点 F の方向が選択されたことになる．

なお，ベクトルの外積をとった結果，その符号が負とゼロの場合は，ゼロを選択する．また，同一符号のベクトルが複数ある場合は，基準ベクトルと各ベクトルとのなす角度により決定され，右方向の場合は，角度の一番大きなベクトル，左方向の場合は，一番小さなベクトルを選択すれば良い．

以上の操作を起点に戻るまで繰り返して行うことにより，外輪郭線情報が得られる．

3.2.3 内輪郭線情報の取得方法

ステップ3における内輪郭線情報の求め方を図3を用いて説明する．

ストローク同士の交点を起点とした最小の多角形をすべて求め，それらの多角形がいずれのストロークの内側にもない場合，その多角形を内輪郭線情報とするというのがこの方法の基本である．

以下，詳細に説明する．

図3において，外輪郭線情報は，3.2.2 項に示す方法でトレースされており，以下の点列が既に選択されている．

$B3,\ B2,\ Y3,\ Y2,\ A3,\ A2,\ A1,\ A4,$
$X1,\ X4,\ B1,\ B4,\ B3$

まず，これらの点列のうち，交点 ($X1,\ X4,\ Y2,\ Y3$) を起点として最小の多角形を求める方法は，以下のとおりである．

交点 $X1$ からは，4方向 ($A4,\ X4,\ C1,\ X2$) に分岐している．点 $A4$ と $X4$ の方向は，既に外輪郭線としてトレースされているので，除外する．なぜならば，あるポリゴンの任意の一辺が外輪郭線の一部である場合，そのポリゴンは内輪郭とはなり得ないためである．点 $C1$ と $X2$ のどちらの方向を選択するかは，基準をどちらにするかによるが，ここでは右回りにトレースすることにすると，左側の点 $C1$ が選択される．次に点 $C2,\ X2,\ X1$ が選択され，多角形1 ($X1,\ C1,\ C2,\ X2,\ X1$) が求められる．以下同様な方法で，交点 $X4,\ Y2,\ Y3$ を起点として，多角形を求めると以下のようになる．

多角形1：$X1,\ C1,\ C2,\ X2,\ X1$
多角形2：$X4,\ X3,\ C3,\ C4,\ X4$
多角形3：$Y2,\ D2,\ D1,\ Y1,\ Y2$
多角形4：$Y3,\ Y4,\ D4,\ D3,\ Y3$

次に，残された交点 ($X2,\ X3,\ Y1,\ Y4$) を起点として上述の方法で多角形を求めると以下の多角形が求められる．この場合は，どの交点を起点としても同じ多角形となる．

多角形5：$X2,\ Y1,\ Y4,\ X3,\ X2$

なお，ストローク3～6の多角形の輪郭特徴点は以下のとおりである．

ストローク3：$A1,\ A2,\ A3,\ A4,\ A1$
ストローク4：$B1,\ B2,\ B3,\ B4,\ B1$
ストローク5：$C1,\ C2,\ C3,\ C4,\ C1$
ストローク6：$D1,\ D2,\ D3,\ D4,\ D1$

上記の多角形のうち，多角形1は，ストローク3と5に，多角形2は，ストローク4と5に，多角形3は，ストローク3と6に，多角形4は，ストローク4と6にそれぞれ含まれる．結果として，多角形5がどのストロークにも含まれないので，内輪郭線を表す多角形となる．

ここで注意すべきことは，すべての文字が内輪郭線を有するわけではなく，外輪郭線だけの文字もある．

上で述べた多角形1～5とストローク3～6（多角形）が互いに包含関係にあるか否かの一般的なチェック手法は以下に示すとおりである[13]．

判定しようとする二つの多角形AとBのすべての構成線分について，線分同士の交差判定を行う．その後，次の方法によって，AとBの関係を求める．

- 交点が全くなく，一方の多角形の1頂点が他方の多角形に含まれない場合，AとBは互いに独立である．
- 交点が全くなく，一方の多角形Bの1頂点が他方の多角形Aに包含される場合，AがBを包含している．
- 交点が二つ以上存在する場合，AとBは交差している．

しかし，上記方法では，処理時間が大きくなることが予想されるため，本報告で扱っている図形の性質を利用して，短縮できる方法がないかを検討した．多角形同士は接するか辺を共有することはあっても，交点を有する線分はありえない．なぜならば，ストローク同士の交点座標を計算して，最小単位の多角形を定義しているため，交点を有する場合はない．交点を有するとすれば，別の多角形が作らているはずである．

そこで，以下に示す方法で多角形同士の包含関係のチェックが可能であると考えた．

(1) 多角形Aの頂点と多角形Bの頂点の間に以下の(a)～(d)の関係がすべて成立することが，多角形AがBを包含するための必要条件である．

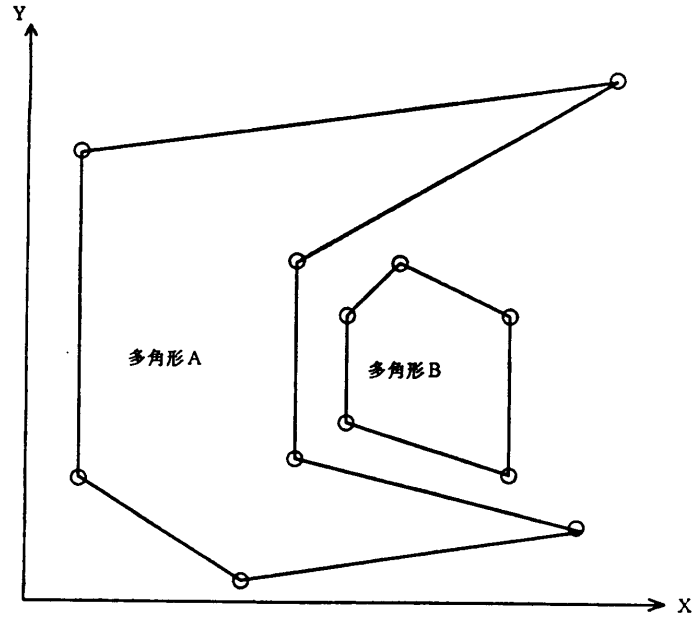


図 4 多角形AがBを包含しない例
Fig. 4 Example that polygon A does not include polygon B.

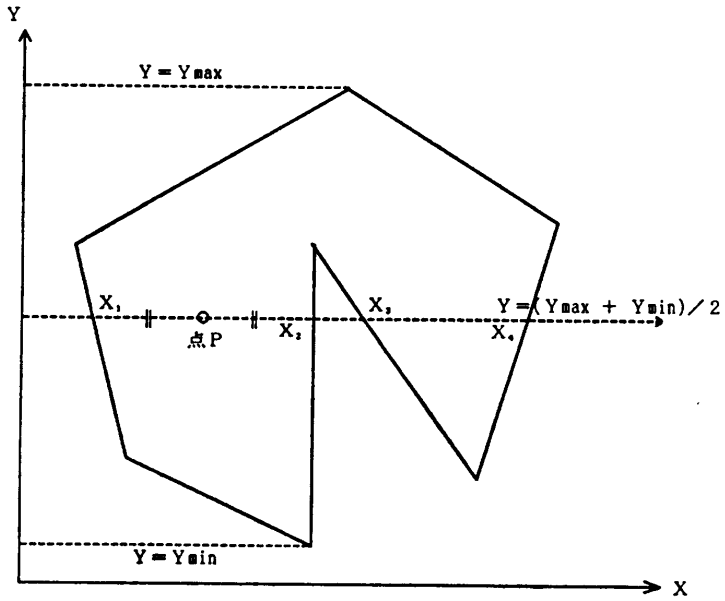


図 5 多角形内の一点を求める方法
Fig. 5 Method of finding point P in polygon.

（a） 最大値 $(XA) \geqq$ 最大値 (XB)
（b） 最大値 $(YA) \geqq$ 最大値 (YB)
（c） 最小値 $(XA) \leqq$ 最小値 (XB)
（d） 最小値 $(YA) \leqq$ 最小値 (YB)

ここで，XA は多角形Aの頂点の x 座標値，XB は多角形Bの頂点の x 座標値，YA は多角形Aの頂点の y 座標値，YB は多角形 B の頂点の y 座標値である．また，座標値は，すべて正数である．

しかし，この条件だけでは不十分で，図4に示すような場合は，上記(a)～(d)の条件は満足しているが，多角形Aが多角形Bを包含していない．そのため，さらに次の条件を十分条件として，チェックするようにした．

（2） 多角形B内の任意の一点を求め，その点が多角形A内にあれば，多角形Aは多角形Bを包含している．

ここで，多角形内の任意の一点を求める方法として，凸多角形の場合は，重心を求めることにより可能であるが，凹多角形の場合は，一般的な方法はないため，以下のようにした（図5参照）．

（a） 多角形の頂点の中で Y 座標の最大値 $(Y\max)$ と最小値 $(Y\min)$ の中間値を求める．

$$Y=(Y\max+Y\min)/2$$

（b） 上式で示される X 軸との平行線と，多角形の交点を求める．

（c） 交点の X 座標値を昇順または降順でソートし，第1番目 (X_1) と第2番目 (X_2) の中点を多角形内の任意の一点 P とする．座標は以下のとおりである．

$$\left[\frac{X_1+X_2}{2},\ \frac{Y\max+Y\min}{2}\right]$$

この点 P が，多角形S内に存在するか否かの一般的チェック方法は，以下に示す二つの方法がある．

（1） 点 P を端点とする任意の方向の半直線と，多角形Sを構成する辺との交点数が，奇数ならば，点 P はSの内部に存在し，偶数なら外部に存在する．

（2） 点 P と多角形Sを構成する頂点 P_i，P_{i+1} とのなす角 $(\angle P_i\ P\ P_{i+1})$ を α_i とする時，$\sum\alpha_i$ が“0”の時，点 P は多角形Sの外部であり，“2π”の時，点 P は多角形Sの内部に存在する．

しかし，(1)に示す方法では，半直線が多角形の頂点を通過する場合は，補正処理が必要となり，本処理全

体が複雑になるため，(2)に示す方法（図6参照）によりチェックすることにした．

以上の二つの条件をチェックすることにより，多角形の包含関係が判別可能なため，内輪郭線情報が得られる．

3.3 変換結果

以上の処理を行うことにより，文字全体の外輪郭線と内輪郭線情報が取得できる．

本変換によって得られた 256×256 ドットの輪郭ベクトル文字のサンプルを図7に示す．図中，実線が外輪郭線で，破線が内輪郭線である．また，上側6文字と下側6文字とでは，文字の太さを変えて生成し，変換した例である．

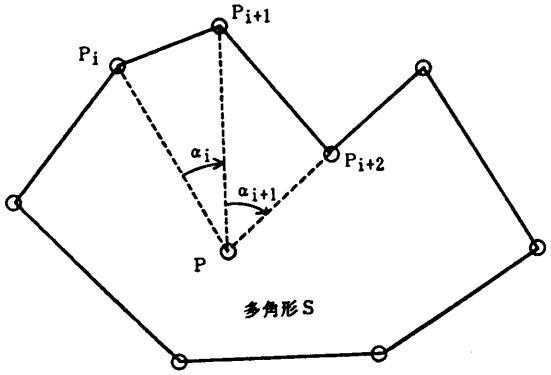


図6 点と多角形の包含関係
Fig. 6 Including relation of point and polygon.

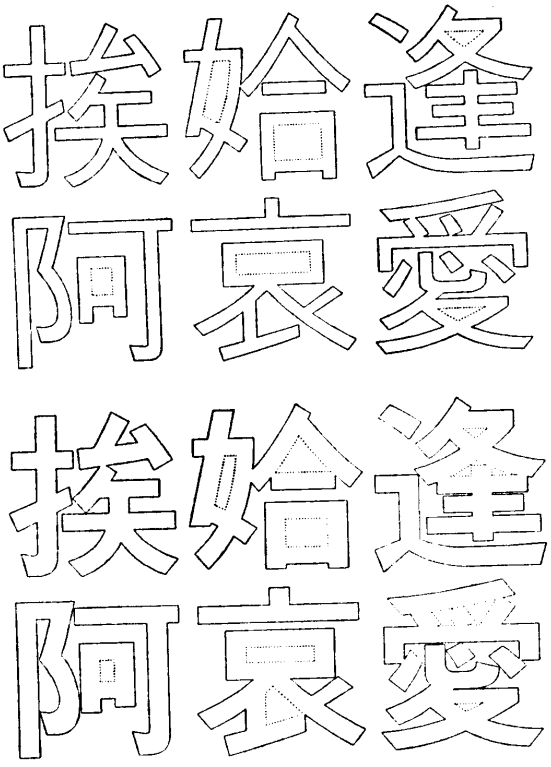


図7 輪郭ベクトル文字のサンプル
Fig. 7 Sample of outline characters.

表2 文字生成・変換性能
Table 2 Performance of character generation and conversion.

文字サイズ	128×128	256×256	512×512	768×768	1024×1024
生成部の処理時間	32.8	33.3	33.6	34.0	34.1
変換部の処理時間	54.4	66.4	85.6	103.4	118.8

処理時間：ms，文字サイズ：ドット

なお，大型計算機（M-680 H）を用いた場合の変換性能は，256×256 ドットの JIS 第1水準の漢字（2965 文字）の平均値で，生成部（骨格点データから，ポリゴンデータを生成するまで）の処理が約 33 ms で，変換部（ポリゴンデータから外輪郭線と内輪郭線情報を生成するまで）の処理が約 66 ms であり，1文字当たり 100 ms（JIS 第1水準全体で約5分）で高品質の文字が生成可能となった．これに対して，アウトライン文字のエディタを用いて人手で文字を作成する場合の工数は，1文字当たり 15 分程度（JIS 第1水準全体で約 100 人日）と言われており，本報告で示したプログラムを使うことにより，文字セット全体の作成工数の大幅な減少が可能である．

また，生成する文字サイズを変えた時の生成部および変換部の処理時間を表2に示す．生成部の処理時間は文字サイズにほとんど依存しないが，変換部の処理時間は，文字サイズが大きくなるに従って増加している．これは，ベジェやスプライン等の曲線で表現するストロークをポリゴン近似する場合，精度を良くするため，生成する文字サイズに応じて頂点数を多くしている．そのため，交点計算等の処理時間が増えるためである．

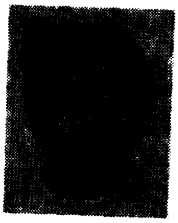
4. おわりに

骨格ベクトル文字情報からストロークごとの輪郭を表すポリゴンデータを生成し，図形的処理を行うことにより，輪郭ベクトル文字として必要な情報である外輪郭線情報および内輪郭線情報を生成することができた．これにより，骨格ベクトル文字の特長である文字の太さ等の変更機能を用いて文字を生成し，品質を劣化させることなく，輪郭ベクトル文字に変換することが可能になった．

参考文献

1) Frank, A. J.: Parametric Font and Image Definition and Generation, *Proc. AFIPS FJCC*, pp. 135-144 (1971).
2) Bigelow, C.: Digital Typography, *Sci. Am.*, Vol. 249, No. 2, pp. 106-119 (1983).
3) 大山ほか：高品質文字フォント生成のための文字輪郭線ベクトル化方式，第 29 回情報処理学会全国大会論文集，pp. 1433-1434 (1984).
4) Adobe Systems Inc.: POSTSCRIPT Reference Manual, Addison-Wesley (1986).
5) Knuth, D. E.: *The Meta-font Book*, Addison-Wesley (1986).
6) Hobby, J. D.: A Chinese Meta-font, Stanford Univ. STAN-CS-83-974 (1983).
7) 高木，坂元：高品質明朝体ひらがなカタカナフォントの計算機による生成，信学論，Vol. J68-D, No. 4, pp. 702-709 (1985).
8) 杉田ほか：マルチフォント漢字合成器，信学会研究会，IE 81-120 (1981).
9) 梶田ほか：毛筆書体の生成アルゴリズム，第 32 回情報処理学会全国大会論文集，pp. 1665-1666 (1986).
10) 菊池ほか：字体のパラメトリック基本エレメント貼り付け方式による高品質文字形状生成方式，第 29 回情報処理学会全国大会論文集，pp. 1435-1436 (1984).
11) 上原ほか：ストローク種別に基づく文字形状生成方式，情報処理学会研究会資料「文書処理とヒューマンインタフェース」，19-2 (1988).
12) 大町：レタリングエッセンス，日本文芸社 (1981).
13) 中前ほか：3次元コンピュータグラフィックス，昭晃堂 (1986).

(平成元年2月 7 日受付)
(平成元年6月 13 日採録)



下位 憲司 (正会員)

昭和 21 年生．昭和 46 年静岡大学大学院工学研究科修士課程修了．同年(株)日立製作所に入社．以来大型コンピュータのハードウェア，光通信システム，光ディスクのアプリケーションおよび文書処理の研究開発に従事．現在(株)日立製作所中央研究所に勤務．

上原 徹三 (正会員)

昭和 19 年生．昭和 44 年京都大学大学院工学研究科修士課程修了（数理工学専攻）．同年(株)日立製作所に入社．文書処理，文字パターン生成等の研究に従事．現在，中央研究所主任研究員．電子情報通信学会，日本ソフトウェア科学会各会員．