

ストローク種別に基づく漢字形状生成方式†

上原 徹 三^{††} 国西 元 英^{††} 下位 憲 司^{††}
 鍵 政 秀 子^{††} 菊 池 純 男^{††}

ストローク種別に基づく文字形状表現の一方式が、太さ等の変形の可能な漢字フォントの表現として有効であることを、ゴシック体漢字を例として示す。漢字形状を、それを構成するストロークに関する情報の系列で表現する。このストローク情報とは、文字セットで定まるストローク種別と、ストロークの骨格点座標列と、太さ等の形状パラメタとの組合せである。形状パラメタとしては、ストローク個別の局所的形状パラメタのほかに、ストローク種別対応の全域的形状パラメタを設け、前者の指定を後者より優先する。本方式を用いて指定の大きさの文字パターンを得るには、まず関連のストローク情報を指定サイズに変形した後、その文字を構成する各々のストロークの情報からその輪郭情報を生成し、これらを合成するとよい。ストローク種別は、ゴシック体漢字の場合、16種を設けた。本方式によると、骨格に関するストローク情報からその輪郭形状を生成する際、ストロークの太さや端辺の角度等の輪郭形状に関する種々の制御が可能となる。さらに、上記の2種の形状パラメタによって、文字のストロークごとに個性のある形状を表現しうると共に、フォント全体の統一性を維持できるという特徴を持つ。

1. はじめに

1.1 背景と目的

コンピュータでの文字出力・表示（可視化）の通常の方式は、文字形状をドット行列として表現するものである。文字形状の可視化方式とメモリ上の保持方式とが同一方式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。この方式をドット行列方式、その文字パターンをドット文字と呼ぶ。

ドット文字は拡大・縮小による品質の劣化が著しいので、最近の文字サイズ・書体の多様化や文字形状の高品質化の要求に応えるには、サイズごとに作成し保持する必要がある。これは字種の多い漢字の場合、メモリ量と作成工数の増大という問題を持つ。さらに、文字形状の高品質化のためプリンタの解像度が向上し同サイズの文字の出力にも高ドット数が必要になってきた。

このような情勢から、ドット行列方式に代わる新しい文字保持方式が必要とされるようになってきている。

線情報の系列を保持するXYプロッタ等のストローク文字では、サイズの変形が可能であるが、太さ情報を持たないため品質面で不十分である。

肉厚を持つ図形情報として文字形状を表現する方式が、まず、英文字で実用化された。文字形状の輪郭線

を図形として保持し、サイズ・傾き等の変換後、必要なら輪郭内をぬりつぶすアウトライン文字方式である。この場合も、可視化段階ではドット行列を用いることが多いが、図形としての変形を可視化前に実現できることに意義がある。紙の上の模範文字をスキャナ等でコンピュータ内に取り込み、輪郭線の追跡後、対話修正によりアウトライン文字形状を完成できる。文字形状の作成工数は、特定サイズのドット文字のそれと同程度である。各種サイズを用意すべきドット文字と比べてメモリ量が少ないこともこの方式の利点である。

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。このとき、文字サイズとストロークの太さとの倍率を異なるものとする制御を可能にしたい。

(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変形書体を作りたい。

アウトライン方式での変形法も考えられようが、本報告では、このような変形に適した方式として、ストローク種別による漢字形状の構造的保持・生成方式を提案する。本方式では、漢字を構成するストロークの形状を与えるストローク情報の系列で漢字1文字の形状を表現する。ストローク情報を、当該フォントの漢字パターンの基本ストロークの分類を与えるストローク種別と、ストロークの骨格点の位置と、太さ等の形状パラメタの3種の情報で表現する。これにより、拡大・縮小等の変形に限らず、ストロークの太さの制御

† A Stroke Type-Based Representation Method of Kanji Font by TETSUZOU UEHARA, MOTOHIDE KOKUNISHI, KENJI SHIMOI, HIDEKO KAGIMASA and SUMIO KIKUCHI (Central Research Laboratory, Hitachi, Ltd.).

†† (株)日立製作所中央研究所

やストロークの端辺形状の変更が可能である。

ドット文字と対比して、アウトライン文字方式と本方式とはベクトル文字方式に分類できる。前者を輪郭ベクトル方式(文字)、後者を骨格ベクトル方式(文字)と呼んで区別する。

本方式の形状パラメタは個々のストロークに対する局所的な指定のほか、ストローク種別対応に全域的に指定された値の利用を可能とする。局所的指定は全域的指定に優先して適用され、局所的指定がないストローク情報に対しては、(そのストロークのストローク種別に対する)全域的指定が適用される。これにより、フォント内のデザインを統一しつつ、個々の文字の個性を盛り込んだフォント・デザインが可能となる。

本稿では、ストローク種別に基づく骨格ベクトル文字方式による漢字形状の生成法と、ゴシック体漢字への適用例について述べる。また、ストロークの太さの変更、端辺の角度や形状の変化等の本方式の特徴について記す。16種のストローク種別によって、JIS規格の漢字約6,000字のゴシック体漢字パターンを設計し、対話型編集システムによって試作した。この設計と試作の方法と結果について述べる。

1.2 研究の経過

アウトライン方式の考え方は古く¹⁾、まず、英字について実用化された^{2),3)}。平仮名についての実験的研究もなされていた⁴⁾。この方式が広く採用される契機は、ページ記述言語の解釈・実行機能を内蔵した出力装置が現れ、それが文字出力に本方式を採用したことであろう⁵⁾。日本でも、アウトライン方式の漢字フォントの実用化の動きが活発になってきた。この方式での太さや端辺形状の変形方法についての検討も進められている⁶⁾。

骨格ベクトル方式による漢字形状生成方式としては、METAFONT⁷⁾の応用例⁸⁾が発表されている。文献8)では、まず、14種のストローク種別を用いて約140種の部首パターンを作成しておき、これらを配置して各文字パターンを生成する。この方式で宋朝体の漢字128文字を試作し、パラメタの変更で長い宋朝体の生成が可能であることを示している。

本報告の方式は文献8)と同類だが、本方式ではストローク種別から直接に漢字パターンを生成し部首の階層を置かない。構造としての部首パターンは多くの漢字に現れるが、部首の形状には文字ごとに相当な変化が見られる。したがって、部首の形状をまず与え、

次にこれらを配置して文字パターンを作成するという方法は、フォントのメモリ量の削減効果は大きい、文字形状の自由度を制限する可能性が強いと考えた。

本方式に沿った操作法の対話型編集システムによる漢字フォント開発の実用性を検討することも、本研究の目的の一つである。当初、明朝体の漢字による原理実験を行い⁹⁾、対話型編集機能の作成¹⁰⁾の後、ゴシック体の漢字約6,000字(JIS第1,2水準¹¹⁾)の試作を行った。

漢字や仮名に骨格ベクトル方式に類する考えを適用した例が日本にも数例ある¹²⁾⁻¹⁴⁾。これらの中には、ストロークの中心線と太さ情報を持ち、ストロークの種別を持たないものも含まれる。

2. ストローク種別に基づく骨格ベクトル文字方式

2.1 文字方式の基本的考え方

ストローク種別に基づく骨格ベクトル文字方式は、文字セットに対して定めた少数の基本ストローク(エレメントともいう)の組合せで文字形状を表現するというレタリングの分野の考えにヒントを得た¹⁵⁾。このエレメントの分類を参考に、コンピュータによる漢字パターンの保持・生成・編集の効率等を考慮してストロークのタイプのセットを設定した。このストロークのタイプの組合せによって生成した形状で上記のエレメントが表現できればよい。このようなストロークのタイプ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」などがある。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。

次に、本方式の漢字形状生成法を具体的に説明する。

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・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。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、各ストローク

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、形状パラメタとを与える。これを、当該文字を構成するストロークの各々に対して行う。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る形状パラメタ（局所的形状パラメタ）によって指定する。一方、フォント内の文字デザインの統一の面から、ストローク種別対応にも形状パラメタ（全域的形状パラメタ）を設定できるようにする。そして、個々の文字のストロークについて、局所的形状パラメタ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そのストロークの属するストローク種別に対する全域的形状パラメタの値を有効とする。

結局、1つのフォントに対する文字表現は、以下の階層構造の情報より成る。

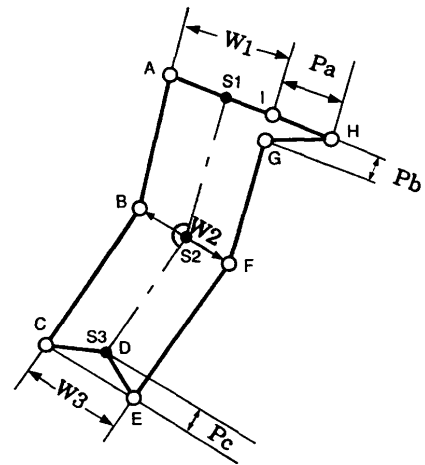
- 1) ストローク種別ごとの全域的形状パラメタ（フォントについて1セット）
- 2) 文字を構成するストローク情報の系列（文字対応）
 - 2.1) ストローク情報（ストローク対応）
 - 2.1.1) ストローク種別
 - 2.1.2) ストローク骨格点座標列
 - 2.1.3) 局所的形状パラメタ

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、また、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様な変更が可能である。

次に、本方式におけるストロークの輪郭形状の生成法について述べる。

2.2 ストローク輪郭の生成

本方式による文字形状の生成法の基本は、上記のストローク情報から、そのストロークの輪郭図形を生成することである。例を用いてこれを説明する。図1は、ストローク種別「左払い」のゴシック体ストロークについて、そのストローク情報と輪郭生成法との関係を示したものである（簡単のためやや単純化した）。図で黒丸印が骨格点（S1, S2, S3）である。ストローク種別と骨格点列（の位置）と局所的形状パラメタ（W1, W2, W3, Pa, Pb, Pc）とが、本ストロークのストローク情報であるものとする。このストローク情報からストロークの輪郭図形を求めるために、まず、輪郭特徴点の列を求める。輪郭特徴点には次の2



- 1) ストローク種別: 「左払い」
- 2) 骨格点: ●
- 3) 形状情報: W1, W2, W3, Pa, Pb, Pc

図1 ストローク情報
Fig. 1 Stroke information.

種がある。それらの間を適当な曲線（直線、円弧等）で補間することによって輪郭曲線が得られるような輪郭線上の点列の場合と、必ずしも輪郭曲線上の点ではないがこれらの点列の制御によって輪郭曲線（ベジェ曲線等）が生成されるような点列の場合とである。簡単のため、いずれの場合でも、特徴点の補間によって輪郭曲線が生成されると言うことがある。特徴点間を補間する曲線の線種は、あらかじめ、当該ストローク種別の形状設計時に、ストローク種別内の部分ごとに定められている。

次にストローク輪郭曲線の生成法の概略を説明する。

図1に示したストローク種別「左払い」は、「行」という字の第1, 2画のように、上方S1から始まり、S2を経て左方向に曲がりつつ下方に進み、S3で終わるストロークである。ストロークの上端には、Pa, Pbの2種のパラメタを用意して変形に備えている。ストロークの下端においては、パラメタPcによって終端形状を制御する。パラメタW1, W2, W3は、上端、中間、下端の太さを指定する。以上から、図のように輪郭特徴点A, B, C, D (=S3), E, F, G, H, Iを得る。3点A, B, CとE, F, Gとをそれぞれ後述の「擬似スプライン曲線」で補間し、3点C, D, EとG, H, Iをそれぞれベジェ曲線で補間し、I, A間を直線で補間する。

以上より、図1に指定された「左払い」ストロークの輪郭図形が生成できる（その生成例は図6を参照）。

2.3 ストロークの接合部の整形と文字形状の合成

文字を構成する各ストロークの輪郭を生成して貼り合わせると、基本的には、その文字全体の輪郭形状が生成できる。しかし、ストローク同士のつながりの部分では、単なるストローク輪郭の貼合わせでなく、接合部分に何らかの変形が要求される場合がある。これをストロークの接合部の整形と呼ぶ。

このような接合部整形の例を試作後のゴシック体漢字について示したのが図2である。図2には「左払い」ストロークと他のストロークとの接合についての幾つかの形態を例示した。本図で、ストロークAは接合のない本来の「左払い」の形である。これに対して、ストロークB、C、Dの各々は、他のストロークと何らかの接合のある例であり、本来のストロークの形状に対してそれぞれ接合部の整形が施されている。

文字パターンの生成方式としては、これら整形の条件を検出する手段と、それに応じた整形を可能とする手段とを用意する必要がある。そこで、整形処理が要求される可能性のある接合の条件を種々の漢字のストロークについて調べた。

この結果、次のような接合条件を考慮すべきであることが判明した⁹⁾。ここで、セグメントとは、同一ストローク内の2つの骨格点を筆の動きに沿って順に結ぶ線分のことである。また、ストロークの起筆部(HEAD)とは、ストロークの第1番目の骨格点(最初のセグメントの第1骨格点)、ストロークの終筆部(TAIL)とは、ストロークの最後の骨格点(最後のセ



- A: 接合なし。
- B: 「横線」の終筆部から始まる。
- C: 終筆部が「右払い」に重なる。
- D: 上の「横線」の途中から始まり
下の「横線」の途中で終わる。

図2 「左払い」ストロークの接合例

Fig. 2 Examples of interstroke joint and link.

グメントの第2骨格点)を指すものとする。

●JOINT 接続

当該ストロークの起筆部または終筆部が、他のストロークの起筆部または終筆部と一致している場合である。この条件が当該ストロークの起筆部で成立する場合を JOINT-HEAD 接続、終筆部で成立する場合を JOINT-TAIL 接続と呼ぶ。当該ストロークの起筆部を上にしたとき接続相手のストロークが右にあるか左にあるかによって整形法が異なることが考えられるので、これらをさらに分類して、JOINT-HEAD-RIGHT/LEFT (JHR/JHL と略す) と JOINT-TAIL-RIGHT/LEFT (JTR/JTL と略す) という4種類の JOINT 機能を考える。

●LINK 接続

当該ストロークの起筆部または終筆部が、他のストロークの起筆部・終筆部以外のセグメント上の点を通る場合である。前者を LINK-HEAD (LH と略す) 接続、後者を LINK-TAIL (LT と略す) 接続と呼ぶ。

図2の「左払い」ストロークの接合例にこの分類を当てはめると、ストロークBは JHL 接続、ストロークCは LT 接続、ストロークDは、上に位置する「横線」に対して LH 接続、下に位置する「横線」に対して LT 接続である。

このようなストロークの接合条件はストロークの骨格位置によって確定するが、どのような条件についてどのような整形を行うのか(あるいは、行わないのか)ということは、デザインの方針に属する問題である。後述のゴシック体漢字の試作においては、単純で読みやすいデザインを目指したため、接合部の形状が複雑にならないような整形を施すこととした。図2の生成例はそのような整形の結果の形状を示している。

以上のような接合部整形を考慮したストローク形状、および、それに基づく漢字1文字の形状の生成手順は次のようになる。

まず、各ストロークの輪郭形状の生成処理の先頭で上記の接合条件を判定する。この接合条件のチェックと成立した条件の分類とは、当該ストロークの起筆部と終筆部が他のストロークの起筆部、終筆部、またはその他のセグメント上の点と一致するかどうか等の判定によって簡単に行うことができる。何らかの接合条件が成立すると、その成立条件および当該ストロークと相手ストロークのストローク種別に応じて、当該ストロークに適用されるべき形状パラメタを変更する。

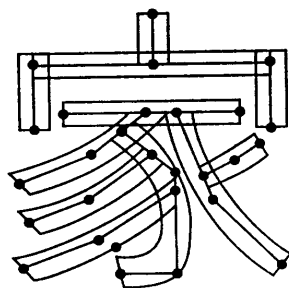


図 3 ストローク輪郭の生成例
Fig. 3 Generated character stroke pattern.

整形の有無によらず、このあと、当該ストロークのストローク種別と骨格点の位置と形状パラメタにより、すでに述べた通常のストローク生成処理を行う。このように生成した各々のストロークの形状を文字枠内の指定の位置に貼り合わせることによって、当該文字の全体の形状が完成する。

以上の方法により生成した「家」という文字の角ゴシック体パターンについて、図 3 に骨格情報と生成後の輪郭形状とを重ねて示した。なお、図 2 の例にあげた LT 接続の「左払い」の終端部の整形結果の輪郭曲線と、接続相手ストロークの骨格セグメントとの関係が本図に示されている。

3. ゴシック体漢字への適用

3.1 ストローク種別

文字フォントを設計するには、まず、そのデザイン方針を定める必要がある。その基本はエレメントの設計である。ゴシック体漢字フォントについて、文字デザイナーの協力によりエレメントの概略を定め、それを効率的に実現するためのストローク種別の設定とその形状の設計を行った。

文字フォントの作成において、オペレータが骨格情報を指定する際の効率には、ストローク種別の個数、ストローク種別の明快さ、1文字の形状を表現するためのストロークの個数等が影響する。妥当なストローク種別とその形状設計は最初から定められるものではなく、実際に文字パターンの作成作業に適用し、その際の作業効率や作成結果の文字形状の品質等を参考にして幾度かの改訂を重ねることによって、完成に近づいていくものといえる。

本方式の場合、約 3,000 字 (JIS 第 1 水準) の作成実験を経た段階で、ゴシック体 (より詳細には、角ゴシック体) の漢字の 16 種のストローク種別とその骨格点および形状パラメタの種類を一応確定した。この

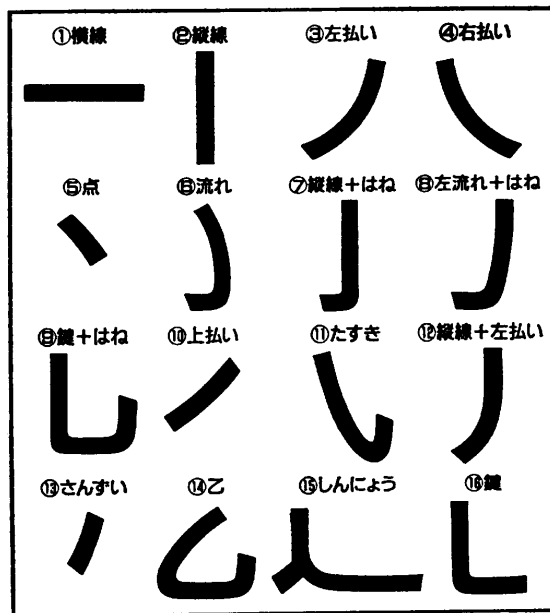


図 4 ストローク種別
Fig. 4 Stroke types.

ストローク種別と、それらに属するストロークの基本的な形状を図 4 に示す。また、表 1 には、ストローク種別を構成する骨格点の数を示すと共に、各ストローク種別が用いられている漢字のストロークを例示した。この表から、「家」という字は、「横線」2本、「縦線」3本、「左払い」4本、「右払い」、「流れ」各1本のストローク群より成っており、文字全体を構成する骨格点数は合計 29 個である。

ストロークの設計においては、ストローク種別と骨格点位置に加えて形状パラメタを設定する必要がある。これには、ストローク種別ごとのストロークの形状の取りうるバリエーションを考え、そのための輪郭特徴点を想定し、それらを生成するための形状パラメタを設定するという設計過程をとる。現在の実現例では、ストローク種別「横線」の輪郭特徴点数は 10 個であり、図 1 で簡略化した例を示したストローク種別「左払い」の実際の輪郭特徴点数は 11 点である。また、形状パラメタの個数は、前者で 7 個、後方で 12 個であり、その役割は、ストロークの太さ、端辺の角度、ストロークの始終端形状等を制御することである。

ストロークの設計は、基本的なデザインを共通とするある程度の書体のバリエーションに対処しうることを考慮して行った。つまり、ここで設定したストローク仕様を用いて、(同一の書体グループに属する) 複数の書体のフォントが生成できる。ある書体のフォン

表 1 ストローク種別とその用途
Table 1 Stroke types and their applications.

No.	ストローク種別	骨格点	荒	家	羽	池	式	込	乙
1	横線	2	1,5	3,5	1,5	4	1,3,5	1,5	1
2	縦線	2	2,3,4,8	1,2,4		6	4		
3	左払い	3		6,8,9,10				2	
4	右払い	3		11				3	
5	点	2			3,7	1,2	6	4	
6	流れ	4		7					
7	縦線+はね	4			2,6				
8	左流れ+はね	4				5			
9	鍵+はね	4	9			7			
10	上払い	3			4,8				
11	たすき	4					2		
12	縦線+左払い	3	7						
13	さんずい	3				3			
14	乙	4							2
15	しんじょう	5						6	
16	鍵	3	6						

注) 第4フィールド以下は、第1行に記す文字を構成する各ストロークが属するストローク種別を示す。数字は、当該文字を構成するストロークを筆順に並べたときの番号である。

トでは全く用いられない形状パラメタが、他の書体のフォントでは重要な意味を持つことがありうる。

基本的なデザインを異にするような別書体に対しては、ストローク種別のセットを改めて設計する必要がある。このとき、ある書体では同じストローク種別に属していた2つのストロークが、別の書体では異なるストローク種別に属することがありうる。さらに、同一の書体に対してもストローク種別のセットの設定法には任意性がある。重要なことは、設定したストローク種別を組み合わせ、対象の書体のエレメントが効率的に生成できることである。

3.2 ストロークの輪郭曲線

本方式における書体対応のストロークの設計は、ストローク種別の設定、その形状を制御する骨格点と形状パラメタの設定、これら骨格情報に基づく輪郭特徴点列の生成法と、輪郭特徴点列の補間法の決定という一連の設計項目を含む。本項では、この中の最後の項目に関して、輪郭特徴点の補間曲線について述べる。

さきに、「左払い」ストロークについて、ストローク情報からの輪郭特徴点の生成法と補間法について紹

介した(2.2節)。このストロークでは、補間曲線として、直線、擬似スプライン曲線およびベジエ曲線を用いていた。ゴシック体漢字のストローク種別全体についての輪郭特徴点の補間曲線としては、このほかに、円弧を用いている。

一般には、対象フォントの書体デザインが異なると、ストローク種別の設計が異なることは、前に述べたとおりである。しかし、このことは、必ずしも輪郭特徴点間の補間曲線の種別にまでは影響しないようである。つまり、対象フォントの書体デザインが異なっても、上記の補間曲線以外の曲線を追加する必要は少ない。むしろ、輪郭ベクトル方式による英文字の実用化例を見ると、3次式ベジエ曲線で広範囲の書体の輪郭を表現している⁵⁾。

ところで、ある程度の表現力を持った曲線の間では、文字パターン編集(入力・修正)を対話型で行う場合の操作性の観点から、曲線の種別の選択の問題が論ぜられることがある。これは、輪郭ベクトル方式の場合、保持パターンを表現する輪郭曲線形状そのものが、編集オペレータの操作対象になるためである。

本方式による文字形状の保持・生成法では、通常、フォント編集オペレータの操作対象は骨格情報であって、輪郭形状はその結果として人手を介在せず生成されるものである。しかし、骨格ベクトル方式をフォント開発に用いるにしても、実際の文書出力システムそのものが保持する文字フォントの形式が骨格ベクトル情報であるとは限らない。アウトライン・フォントを内蔵するシステムや出力装置の普及の現状を見ると、骨格ベクトル情報そのものではなく、それから生成したストローク輪郭情報や文字輪郭情報（アウトライン情報）を実システムに組み込むことが、少なくとも当面は現実的である。

実際出力システムでは、外字作成機能が必須であり、この機能は当該システムの内蔵する文字パターン編集も可能であるのが便利である。このような状況により、骨格ベクトル情報から生成した輪郭パターンについても、それを対象とする編集操作を行う可能性があり、この編集操作の効率の面から輪郭曲線の種別を検討しておく必要がある。また、一方、出力用のフォント・パターンとして利用する以上、出力処理の複雑さや処理性能との関係からも、輪郭曲線の種別の特性を考慮しておく必要がある。

そこで、各ストローク種別の骨格情報から生成される上記4種のストローク輪郭曲線について、これらの観点から若干の検討を行う。まず、4種の輪郭曲線の各々が適用されるストローク種別の例と共に、輪郭曲線を規定する輪郭特徴点を次に示す。

- **直線**は、ゴシック体の漢字では、ほとんどすべてのストロークで用いる。直線は、2点の輪郭特徴点に対して指定される。
- **円弧**は、「縦線+左払い」の左への曲がりのところに用いる。円弧は、弧の両端の2点の輪郭特徴点と中心とで指定される。
- **擬似スプライン曲線**は、「左払い」、「右払い」、「上払い」、「さんずい」等で用いられ、前に述べたように、その上を通る3点に対して指定される。
- **ベジェ曲線**は、すべてのストローク種別で用いられ、頻度も直線よりさらに多い。ベジェ曲線は、3点または4点の輪郭特徴点に対して指定され、そのうち両端の2点を通り、特徴点列の与える3角形または4角形の中に収まる。

各ストロークの輪郭曲線は、その特徴点を指定し、あるいは、修正し、それに応じた輪郭曲線を生成し確認することができるシステムによって編集可能であ

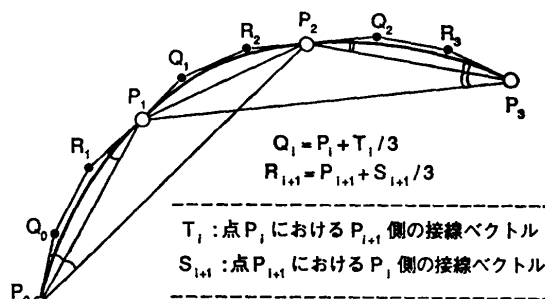


図5 擬似スプライン曲線
Fig. 5 Pseudo-Spline curve.

る。直線、円弧、および、ベジェ曲線については通常用いられる曲線であり、その生成・編集法は明らかである。次に、これまで「擬似スプライン曲線」と称した曲線について説明する。

図5は、 P_0, P_1, P_2 および P_3 を通る擬似スプライン曲線である。点 P_i における P_{i+1} 側の接線ベクトルを T_i と記し、点 P_{i+1} における P_i 側の接線ベクトルを S_{i+1} と記す ($i=0, 1, 2$)。

中間点 P_{j+1} における接線ベクトルの方向は、線分 $P_j P_{j+2}$ に平行である ($j=0, 1$)。接線 T_0 が線分 $P_0 P_1$ となす角は角 $P_1 P_0 P_2$ に等しく、接線 S_3 が線分 $P_3 P_2$ となす角は角 $P_1 P_3 P_2$ に等しいものとする。接線 T_0 と S_1 の大きさは $P_0 P_1$ 、接線 T_1 と S_2 の大きさは $P_1 P_2$ 、接線 T_2 と S_3 の大きさは $P_2 P_3$ に、それぞれ、等しいものとする。

そこで、擬似スプライン曲線とは、2点 P_i および P_{i+1} を通り、両点での接線ベクトルが T_i と S_{i+1} である3次曲線である。これは、4点 P_i, Q_i, R_{i+1} 、および P_{i+1} を制御点とするベジェ曲線である。

ここで、

$$Q_i = P_i + T_i / 3$$

$$R_{i+1} = P_{i+1} + S_{i+1} / 3$$

である。

これにより、本方式で生成するストロークの輪郭曲線は、結局は直線、円弧、ベジェ曲線の3種の曲線となる。これらの曲線に関する対話型の編集操作は通常用いられるものであり、特別の問題はない。

文字パターンを出力する処理から考えても、直線、円弧、ベジェ曲線は一般的なものである。なお、最近、種々の曲線を集約的にベジェ曲線で表現して出力するシステムが見られる⁹⁾。円弧はベジェ曲線で高精度に近似できる（直線もベジェ曲線の特別の場合として表現できる）¹⁰⁾ ので、本方式の輪郭特徴点のすべて

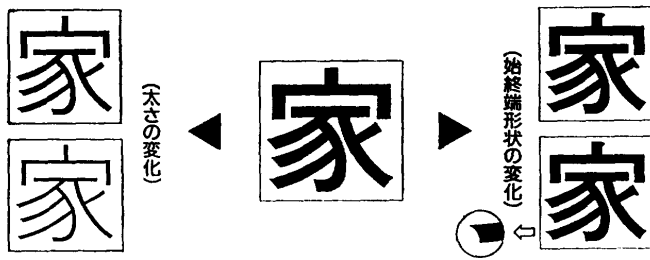


図 6 文字形状の変形例

Fig. 6 Character pattern variations.

の補間曲線をベジェ曲線に置き換えて、このような出力方式に従うことも可能である。

3.3 文字生成結果とその検討

本方式に基づいて、文字パターン編集プログラムを試作した。本プログラムは、模範となるドット文字を画面に表示し、その上をなぞってストローク情報を入力していく対話型編集機能を持つ¹⁰⁾。これによって作成した漢字パターンの例として、図 6 に太さと端辺形状の変形例を示す。JIS 規格約 6,000 字のゴシック体漢字のあるデザイン方針に従って試作したが、その生成例を図 7 に示した。

骨格パターンは 384 ドット角の正方形の中に作成した。これを拡大・縮小した文字パターンを生成するとき、太さの拡大・縮小率をサイズのそれと同じとした場合を 100 とし、それに対する太さの割合を百分率で表して、太さ拡大率と呼ぶ。384 ドットの元のパターン上で適当な太さを設定してあるが、この文字パターンを拡大・縮小したとき、相似形（太さ拡大率 100%）の太さが適当であるといえない。

図 7 の例は、128 ドットの文字パターンを太さ拡大率を一律に 90% として出力したものである（出力装置の解像度は 240 ドット/インチ）。一般に、太さ拡大率は、文字サイズの縮小時は 100% またはそれ以下、拡大時は 100% またはそれ以上が適当である（ただし、理想的な太さ拡大率は、文字ごとにその特性によって決められるべきである）。

高品質の文字フォントを開発するには、その文字フォントの用途に適したデザイン方針を設定すること、および、当該フォントを構成する個々の文字の形状を、それ自身が高品質であると同時に、このデザイン

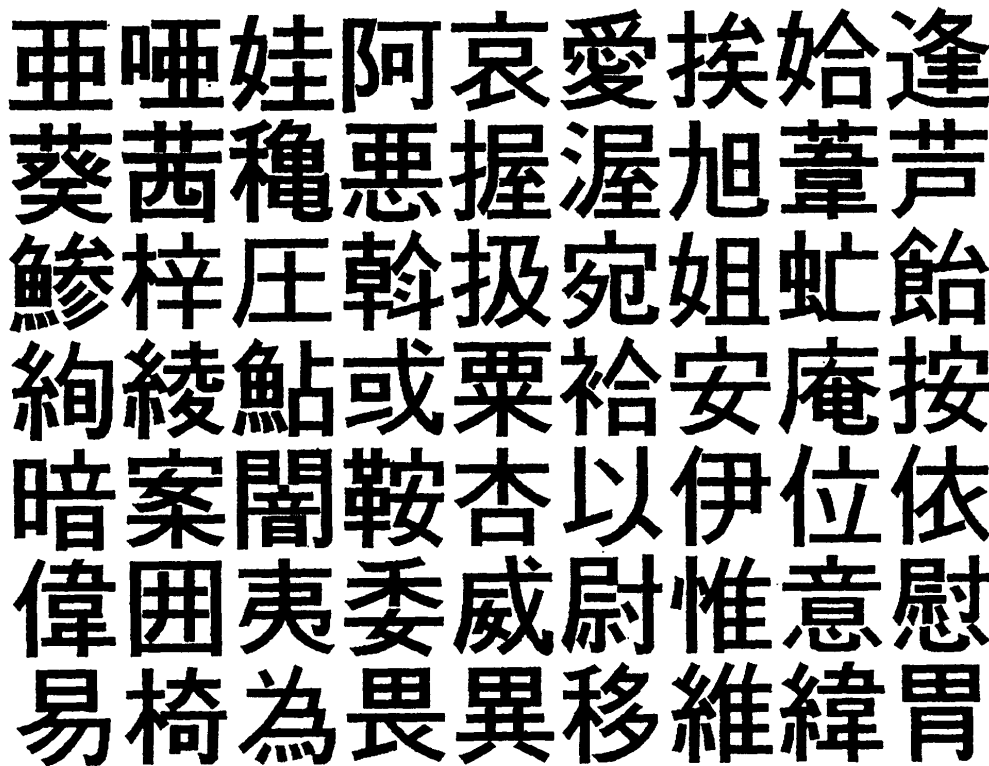


図 7 ゴシック体漢字形状の生成例

Fig. 7 Generated Gothic Kanji patterns.

方針に従うように作成することが重要である。個々の文字形状の品質を向上するため、あるいは、デザイン方針を守るためには、作成した文字パターンを並べて出力し、種々の観点からチェックし修正するという作業のサイクルを必要とする。このサイクルの回数は一度に留まらず、高品質性を追求するほど、多くの繰り返し回数を要する。もちろん、各サイクルの密度もさまざまでありうる。今回の試作では、これまでに、文字相互間のサイズや黒っぽさの調整を主とする粗い密度のサイクルを2回実施した。図の例もこの結果であり、JISコード順の最初の部分である。

この場合の修正作業では、文字パターン作成者がストロークの太さを局所的形状パラメタで指定していたものを削除して、全域的形状パラメタに従うように修正した例が多かった。これは、本方式の機能によってフォント・デザインの統一性を向上する一例である。

しかし、文字パターン相互のサイズのバランスや横書きにおける上下位置のバランス等の調整は、現在の機能では統一的に制御できない。また、個々の文字単独の形状として、でき上がりの悪いものもある。これらについては、個々の文字の個々のストロークの骨格点の位置や形状パラメタを修正する必要がある。

このように、試作文字フォントの品質を向上し、実用的なフォントに仕上げるためには、上記のサイクルをさらに重ねる必要がある。本方式の役割は、定式化されたデザイン上のルールをプログラムに組み込むことによって、上記サイクルの作業効率を向上し、文字デザイナを中心とする人々が高品質フォントを開発するための補助手段を提供することである。このような適用の経験を積み、さらに多くのルールを方式に組み込むことができれば、文字フォント開発の効率の向上にさらに貢献できると思われる。

実際には、個々の文字フォントの開発工数は現実の条件で制約されることが多い。工数の制約を持った文字フォント開発において、本方式のような補助手段の役割は、開発される文字フォントの品質を向上することであるともいえる。

4. おわりに

ストローク種別に基づく骨格ベクトル文字方式の概要とその角ゴシック体漢字への適用法について述べた。本方式では、ストローク情報の系列として1つの漢字パターンを表現する。ストローク情報は、ストローク種別とその骨格点の座標列と形状パラメタとの

組合せである。形状パラメタとしては、個々の文字ストローク対応に指定する局所的形状パラメタと、ストローク種別対応に設定される全域的形状パラメタとの2種を用意し、前者の指定を省略したときに後者を用いる。

ゴシック体漢字についてデザイン方針を設定し、16種のストローク種別を用いて本方式によるフォントを設計した。対話型文字パターン編集システムを作成し、上の設計に従ったJIS漢字約6,000字を試作した。

明朝体の漢字については、以前に、特徴的な形状を持つ漢字を選んで本方式の適用性を検討した⁹⁾。これらの漢字は、本文で述べたのと共通のストローク種別のセットで生成できた（その輪郭形状生成法は異なる）。元来の漢字のストロークの単位を失った草書体のような書体に本方式をそのまま適用することはできないが、ゴシック体や明朝体のようなストロークの単位の明確な書体については本方式は適用可能であり、ストローク種別の多くを共通にできると考えている。

本方式の今後の課題の1つは、仮名文字・英文字への適用性の検討である。現在、本稿に述べた漢字の場合より自由な形状を表現できるようストローク機能を拡張した形で、これらの文字の生成実験を進めている。

高品質の文字フォントの要件は、個々の文字パターンのバランスよい表現と、各文字間のデザイン方針による統一とである。本方式の役割は、高品質の文字フォントをこのような過程で開発する際の効率向上手段の提供である。個々の文字の形状は、各ストロークの骨格点の位置と局所的形状パラメタで制御できる。一方、文字フォントの全体的な統一には、ストローク種別、その太さや端辺形状に利用できるストローク種別対応の全域的パラメタ、その輪郭形状の生成ルールが役立つ。

今後の課題の1つは、文字パターンのサイズや位置関係の統一などを可能とするように、本方式のフォント形状の統一制御機能を拡張することである。

謝辞 本研究の方向づけの段階で、助言とご支援を頂いた当社中央研究所の堀越 彌所長とソフトウェア工場の高橋 栄部長、および文字デザインの知識およびゴシック体漢字のデザイン・ルールの設定等について、ご支援・ご討論頂いた坂本 達氏や多田 護氏ほかのデザイン研究所のデザイナの方々、その他協力を頂いた多くの方々に深く感謝いたします。

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 Typograh, *Sci. Am.*, Vol. 249, No. 2, pp. 106-119 (1983).
- 3) Flowers, J.: Digital Type Manufacture: An Interactive Approach, *IEEE Comput.*, Vol. 17, No. 5, pp. 40-48 (1984).
- 4) 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).
- 5) Adobe Systems: POSTSCRIPT Reference Manual, Addison-Wesley (1986).
- 6) 日本規格協会情報技術標準化センター: 昭和 62 年度高品質フォントの標準化調査研究報告書, 日本規格協会 (1988).
- 7) Knuth, D. E.: The METAFONT Book, Addison-Wesley (1986).
- 8) Hobby, J. D. and Gu, G.: A Chinese Meta-Font, Stanford Univ. STAN-CS-83-947 (1983).
- 9) 菊池ほか: 字体のパラメトリック基本エレメント貼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).
- 10) 国西ほか: ベクトルフォント編集プログラムの機能と方式, 第 29 回情報処理学会全国大会論文集, pp. 1437-1438 (1984).
- 11) JIS C 6226 (現在 X-0208) 情報交換用漢字符号系, 日本規格協会 (1983).
- 12) 坂元ほか: 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).
- 13) 張ほか: 漢字楷書毛筆字体の計算機による生成, 信学技報, PRL 83-24 (1983).
- 14) 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).
- 15) 大町: レタリングエッセンス, 日本文芸社 (1981).
- 16) 山口: コンピュータディスプレイによる形状処理工学(II), 日刊工業新聞社 (1982).

(昭和 63 年 11 月 2 日受付)

(平成 元年 11 月 14 日採録)



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.



国西 元英 (正会員)

1979 年愛媛県立松山工業高校卒業. 同年(株)日立製作所に入社, 中央研究所に勤務. 文書出力, 文字パターン生成編集等の研究に従事. 電子情報通信学会会員.



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.



鍵政 秀子 (正会員)

1979 年津田塾大学文芸学部数学科卒業. 同年(株)日立製作所に入社, 中央研究所に勤務. 仮名漢字変換, 文書割付け, 文字パターン評価等の研究に従事.



菊池 純男 (正会員)

昭和 27 年生. 昭和 53 年東京工業大学理工学研究科電気工学修士課程修了. 同年(株)日立製作所入社. 以後, 中央研究所にて 3 次元形状処理, 画像処理, 文字形状処理の研究開発を経て, 現在, 言語処理系とプログラミング環境の研究に従事. 共著「CAD/CAM 入門」(工業調査会). ACM 会員.