

## 解説

## 3. DTP の要素技術—技術動向の現状を知るために—



## 3.4 フォント関連技術の現状と課題†

上原 徹 三竹

## 1. はじめに

フォントとは、従来、共通のデザイン方針に基づく特定サイズの文字パターン一式を指す用語である<sup>10)</sup>。日本では、明朝体、角ゴシック体などのデザイン上の分類である書体に個別の特徴を加えた文字パターンのセットを固有のフォント名称で呼ぶ。基本デザインが共通で太さの異なる細明朝体、太明朝体など同類の書体のフォントをまとめてフォント・ファミリと言う。太さのほか基準枠および後述の元素の変化によるファミリ構成が考えられる<sup>10)</sup>。サイズごとに文字パターンが用意されるドット文字では上のフォントの定義がそのままあてはまるが、サイズ変形が可能なベクトル文字のような最近の技術の下では、サイズに言及しないでフォントの語を用いる<sup>30)</sup>。

テキストの可視化装置では、紙面や画面に文字パターンをドットの行列で表現することが多い。この表現と可視化装置で保持する文字形状表現とが同一である必要はないが、従来、保持方式もドット行列によるものが多く、これをドット行列方式、その文字パターンをドット文字と呼ぶ。本稿では、この可視化フェーズでの保持方式とフォントの設計開発フェーズでの保持方式とを区別して考える。前者では、可視化性能、メモリ量、可視化品質が重要であり、後者では、出力装置に依存しないデザインとしての品質、編集時の操作性、開発工数、表現形式の変換性などが重要である。

ドット文字はサイズ変形での品質劣化が著しいので、サイズごとに作成し保持する必要がある、メモリ量と開発工数の増大につながる。プリンタの解像度の向上によりこの問題は今後さらに大きくなる。また、図面内の文字パターンなどで要求される、任意角度の

回転におけるドット文字の品質の劣化には、対策が困難である。以上により、フォントに対する最近のニーズに対処するには、新しい文字保持方式が必要となる。

これに代えて、文字形状の輪郭線を保持しサイズ変換や回転などの変換を可能とするアウトライン文字方式が、まず英文字を対象として実用化された<sup>11),19)</sup>。これによると、多種サイズのフォントが、サイズ間のデザインを統一しつつ少ないメモリ量で実現できる。アウトライン文字の通常の作成手順では、原字をスキャナなどで取り込み、輪郭線の追跡後、対話型編集により文字形状を完成する。可視化フェーズでドット文字を用いる場合も、設計開発フェーズでアウトライン文字を作成した後で目的のサイズのドット文字に変換するならば、デザインを統一しつつ開発工数が低減できる。

ところで、最近では、基本フォントに太さや端辺形状の変形を加えたフォント・ファミリの構成や、まったく新規のデザインのフォントへの要求がある。これに代えるため、従来どおりデザイナーが紙上に原字を完成することから始めるならば、その工数は非常に大きい。このような条件に対応し得るフォント保持方式としてストロークの骨格に基づく方式がある<sup>9),12),13)</sup>。この方式では文字形状をストローク情報の系列で表現する。ストロークは筆画の構成単位で、一画に一または複数のストロークが対応する。この方式の一例では、フォントのストロークの形状分類を与えるエレメント種別とストロークの骨格点の位置と太さなどの形状パラメタの3種の情報でストローク形状を表現する<sup>40)</sup>。エレメント種別をもたず、骨格線と形状パラメタによる方式も考えられるが、いずれにせよ、文字パターンの生成時に、サイズに限らず太さや端辺形状の統一的な変更を可能とすることができる。

アウトライン文字方式と骨格に基づく方式とは、文字パターンを図形として表現するという意味でベクト

† Current Technology and Problems in Computer Font by Tetsuzou UEHARA (Central Research Laboratory, Hitachi, Ltd.).

\*\* (株)日立製作所中央研究所

ル文字方式に分類し、前者を輪郭ベクトル方式、後者を骨格ベクトル方式と呼ぶことにする。

本稿では、多様な書体やサイズの漢字を含む文字セットの高品質フォントに対するニーズを念頭に、まず、フォントとその基礎技術に対する要件を考える。次に文字フォント表現形式の現状を概観し比較する。これに基づき、特に日本語フォントで重要であるフォント設計開発システムの形態と機能を考える。最後にフォント技術の今後の課題を述べる。

## 2. フォントと関連技術の要件

文字の可視化品質には、その保持パターンそのもののほかに、装置の可視化方式が関係する。同一のドット文字を用いても、可視化装置の種別（ドット・プリンタか、レーザ・プリンタかなど）や方式（レーザ・プリンタでトナーを感光部に付着させるか非感光部に付着させるかなど）により、線が太りやすいかやせやすいかなど、可視化特性が分かれる<sup>28)</sup>。このことは、可視化装置の種別や方式によって保持パターンを変更すべきことを示唆するが、可視化装置に依存した、いわば実装フォントの検討は割愛する。一つの可視化システムで、複数種の可視化装置を接続したり、不特定の可視化装置を接続したりする場合でも、共通のデザインのフォントを各装置で可視化したい。つまり、特定のシステムに固有で装置に共通のフォント（ここでは特定フォントと呼ぶ）が、実装フォントを一般化した、可視化フェーズのフォント保持単位として必要である。

しかし、さらに基本的な重要性をもつのはフォントの設計開発方式の検討である。開発工数の大きい日本語フォントについて、多様な高品質フォントのニーズに応えるには、特定の可視化システム対応でなく、複数の可視化システムに適用可能なフォントの形態とその設計開発法が必要となる。以下に、今後のフォント設計開発フェーズにおける文字方式で重要と考えられる要件をあげる。

(1) **原字パターンへの忠実性**：原字が存在する場合、それに従った文字パターンを作成できること。

(2) **可視化状態の文字パターンに対する制御可能性**：可視化パターンに対してデザイナーの意志を反映しつつ、文字パターンの編集操作が可能であること。

(3) **サイズ変換と回転の品質**：拡大縮小および回転の品質が高いこと。

(4) **デザインの統一性**：フォント内の文字間のデ

ザインの統一が実現できること。

(5) **デザインの統一的な変更の可能性**：フォントのデザインの特徴を統一的に変更し、フォント・ファミリの構成を支援できること。

(6) **フォント開発効率の向上**：各種可視化フォント（特定フォント）全体の開発工数が低減されること。

(7) **可視化フォントへの変換の容易性**：可視化フォント（特定フォント）への変換機能の作成が容易であり、変換作業が簡単であること。

## 3. 文字パターンの表現形式

文字パターンの表現法については、1970年ごろから研究がなされている。当時の研究の重点はフォント・メモリ量の圧縮にあったが、可視化性能はもちろん品質にも程度の差はあれ注意が向けられていたことは現在と変わらない。1980年ごろまでの研究例は英字については文献(6)、8)、漢字については文献(7)に表現法の分類の下に概観されている。現在の表現方式のアイデアの多くはここに現れている。英字では、文字輪郭による表現法<sup>29)</sup>、輪郭をスプライン曲線で表現する方法、ステムやアームなどストロークの部分パターンとその形状変化を与えるパラメータと配置情報による構造的表現法など、漢字では、太さをもたない線ストローク法、部首などの部分パターンを保持して配置位置を指定する合成法<sup>31)</sup>、ストロークの種別を指定することにより明朝体ふうの端辺形状を生成する方法<sup>5)</sup>などが紹介されている。以下では、最近の状況を中心として、文字パターンの表現形式を分類し比較する。

### 3.1 文字パターン表現形式の分類

多種書体のフォントを多種サイズについて用意するには、字種の多い日本語の場合、特に大きな開発量を要し、それに対処できる文字パターンの表現生成方式の技術が要求される。以下、既存の表現方式を分類して示す<sup>43)</sup>とともに、主な例を図-1に示す。

(1) **ドット行列方式(ビットマップ方式)**：文字パターンをドットの集合で表現する。コンピュータによる文字パターンの可視化のために広く用いられている。拡大縮小品質が低い。

(2) **線ストローク方式**：文字パターンを太さをもたない線ストローク情報で表現する<sup>3)</sup>。プロッタやグラフィック・ディスプレイ装置で用いられる。一つのストロークを2本の曲線で表現して品質を上げる方法もある<sup>26)</sup>。

(3) **輪郭ベクトル方式**: 文字パターンの輪郭形状を図形として保持する。輪郭図形の作成単位により、次の二つに分ける。

(3.1) **ストローク輪郭方式**: 文字パターンを構成するおのおのストロークの輪郭線を与える。骨格ベクトル方式の文字生成の中間段階で生成される<sup>40)</sup>。また、ストローク単位の輪郭図形をあらかじめ保持し、変形を加えて配置する方法も考えられる<sup>31)</sup>。

(3.2) **文字輪郭方式 (袋文字, アウトライン文字方式)**: 文字パターン全体を一図形とみて輪郭線を与える<sup>29)</sup>。その考えは古く<sup>2)</sup>、かな文字などでの実験的試みもあったが<sup>15)</sup>、DTP (Desk Top Publishing) システムとしてページ記述言語とともにプリンタに実装されて一挙に実用化した<sup>19)</sup>。ベクトル文字という用語でこれを指すことも多い。

(4) **骨格ベクトル (インライン・ベクトル<sup>26)</sup>) 方式**: 文字ストロークの骨格情報と輪郭形状を生成するための属性情報とを保持し、これらによって文字パターンを生成する。次の方式に分類できる。

(4.1) **エレメント方式**: フォント内の文字パターンを特徴づけるストローク内の基本形状を文字デザインの分野でエレメントと呼ぶ<sup>18)</sup>。エレメントの組み合わせでフォント内のすべての文字パターンが作られる。エレメント方式では、文字ストロークあるいはそ

の部分の形状を類別し、その種類と骨格情報とから輪郭形状を生成する。エレメントの適用部分により、さらに次のように分類できる。

● **ストロークエレメント方式**: この方式の漢字への適用の基本的な考え方として1978年の提案がある<sup>9)</sup>。最近では、エレメントの種別とエレメントに依存した骨格情報と形状パラメータをストロークごとに与える方法が、デザインを統一した漢字フォントの表現法として試みられている<sup>9)・13)・40)</sup>。骨格情報はストロークの中心線を与えるものとは限らず、輪郭の生成に必要な点列である。さらに、毛筆の筆触の形状を設定し、筆画の分類と骨格情報と筆圧情報で毛筆書体を生成する方法が提案されている<sup>12)</sup>。

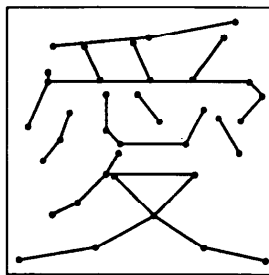
● **始末端エレメント方式**: ストロークの始末端形状を分類し、その始末端種別とストロークの骨格線とでストローク形状を表す方法が毛筆文字について提案されている<sup>21)</sup>。

(4.2) **非エレメント方式**: 上記のエレメントの概念を含まないインライン・ベクトル方式である。普通、中心線と形状情報を保持し、それにより輪郭を生成する。この方式を平仮名生成に用いた報告<sup>17)</sup>、漢字に適用した例<sup>30)</sup>がある。

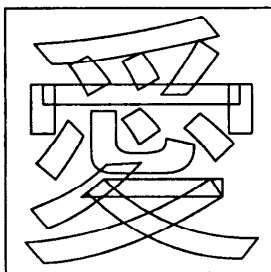
あらかじめ保持した部分パターンを配置して文字パターンを構成する**部分パターン合成法**の考え方<sup>1)</sup>は、



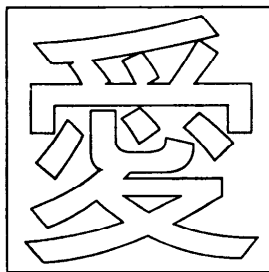
(a) ドット行列方式



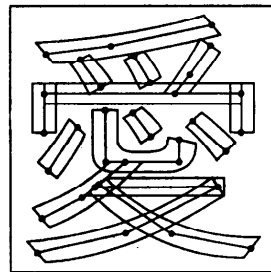
(b) 線ストローク方式



(c) 輪郭ベクトル方式  
ストローク輪郭



(d) 輪郭ベクトル方式  
文字輪郭



(e) 骨格ベクトル方式  
ストロークエレメント

図-1 文字パターン表現の例

注)

- (a) は80ドットパターンをメッシュとともに示す。
- (b) の黒点は線分の端点を示す。
- (c) 4) 5) は同一デザインの文字パターンを示す。
- (d) の黒点は骨格点を示す。形状パラメータは略した。

上記の表現方式によらず適用可能である。骨格ベクトル方式で漢字の部首による部分パターン合成法を採用した例がある<sup>9)</sup>。輪郭ベクトル方式の中にもエレメントを変形しつつ利用する例がある<sup>31)</sup>。

また別の見方として、文字パターン情報が文字形状そのものを直接に表現するもの（ここでは**直接表現法**と呼ぶ）と、文字パターンを生成するための属性や条件を記述するもの（**間接表現法**、**属性記述法**）とに分類できる<sup>26)</sup>。ただし、属性記述法でもその属性記述の程度はさまざまである。フォントの設計開発システムでは直接・間接の両性格をもつことが望ましい。ドット文字はそのデータ圧縮表現を含めて直接表現と言える。英字のアウトライン文字フォントについては、可視化時の品質を考慮してベースラインなどの水平線位置などを保持していることにより、属性記述法に分類されるものがある<sup>23), 26), 39)</sup>が、単純に輪郭曲線のみを表現するものはその品質を問わず直接表現法と言える。骨格ベクトル方式は全体として属性記述法に分類できるが、非エレメント方式、始終端エレメント方式、ストロークエレメント方式の順で後のものほど属性記述度が高くなっている。

METAFONT のような文字パターン記述言語<sup>20)</sup>は、それ自身は文字パターンの表現法を規定するもの

でなく、それは作成するプログラムに依存する。

線ストローク文字とベクトル文字の表現曲線種別として、直線、スプライン曲線 (spline curve) やベジェ曲線 (Bézier curve)<sup>45)</sup>、円などが用いられる<sup>24), 25), 29)</sup>。

### 3.2 文字パターン表現形式の比較

ドット行列、輪郭ベクトル(文字輪郭)、骨格ベクトル(ストロークエレメント)の各方式について、表-1に特徴・用途・課題などを記し比較した。ドット行列方式は、出力性能には問題がない。また、文字パターンの最小単位であるドット単位でデザイナーが直接関与できるため、当該サイズに限れば高品質が期待できる。問題は変形品質が低いことである。輪郭ベクトル方式は、原字に忠実なパターンが作成でき、サイズの変形が容易であるが、端辺形状などのデザイン上の変形は困難である。骨格ベクトル方式は、原字に忠実な形状を与えるには相当の検討と輪郭パターンの生成機能の開発を要するが、一度実現するとデザイン上の変形をフォント内で統一的に実現できる。ただし、生成時間は最も大きく、骨格情報から輪郭情報を生成する処理が輪郭ベクトル文字による生成処理に加わる。輪郭ベクトル文字からはドット行列文字が、骨格ベクトル文字からは輪郭文字とドット行列文字が生成できる。

表-1 文字フォント表現方式の特徴

	項 目	A. ドット行列方式 ドット行列パターン	B. 輪郭ベクトル方式 文字輪郭線パターン	C. 骨格ベクトル方式 ストロークの種別、位置、形状
1	方式概略	2値ドット行列による文字パターン表現	文字パターン図形の輪郭線パターン	文字ストローク単位の種別、位置、形状特性情報
2	研究開発状況	ワードプロセッサ製品 プリンタ製品	DTP システム製品	研究レベル
3	1) 原字への忠実性	◎忠 実	◎忠 実	△ほぼ忠実
	2) 可視化性能	◎ドットの配置のみ	△図形処理と塗りつぶし	△図形処理と塗りつぶし
	3) 生成機能の開発工数	◎開発済み	○容 易	△書体ごとに検討
	4) フォントメモリ量	×サイズごとに保持	○サイズにほぼ独立	○サイズにほぼ独立
	5) フォント開発工数	×サイズごとに開発	○サイズにほぼ独立	○サイズにほぼ独立
	6) 拡大品質	×斜線部劣化	○相似的拡大可	◎相似的拡大と太さ制御可
	7) デザイン変形	×不 可	△困 難	○太さ、端辺形状制御など可
	8) デザインの統一性	×人 手 依 存	△異サイズ間統一可	◎異サイズ間・文字間統一可
	9) 他の形式への変換性	×不 可	○ドット文字への変換可	◎ドット・輪郭文字への変換可
4	方式の課題	な し	高速出力、太さ変形	対象の拡大、変形種別の拡大
5	主な用途	ワードプロセッサ 低価格プリンタ	高価格プリンタ DTP システム フォント設計開発ツール	フォント設計開発ツール 大型文字列可視化システム

## 4. フォントの設計開発

### 4.1 トップダウン・アプローチ

文字フォントを新たに開発するには、字体と書体の両面について、フォントを統一するデザイン方針を定めなければならない。用語の使い方は一定していないが、ここでは、**字体**<sup>41,37)</sup>とは、跳ねるか留めるか、付けるか離すかなど、字をその字と認識させる骨組み(字体で識別される字を**字種**という。たとえば、教育漢字は881の字種よりなる)を言い、一方、**書体**は、線の太さ(の変化)、端辺の形状などのデザイン上の形に関する。字種はISOのフォント規格案<sup>39)</sup>におけるグリフ(Glyph)に、書体は英字のいわゆるタイプフェイス(Typeface)に近い概念である。各文字の書体を統一して一つのフォントが作られる。同じフォントの文字パターンは、サイズの相違によらず共通のデザインを表現する。

フォントの設計法として**トップダウン設計**、**ボトムアップ設計**および両者の折衷があげられる<sup>26)</sup>。高解像度で品質良くデザインされたフォントに基づいて低解像度用フォントを開発するのが**トップダウン設計**である。逆に、低解像度用の既存フォント(たとえば24ドットのドットフォント)を利用して、高解像度用のフォント(たとえば56ドット)を開発するのが**ボトムアップ設計**である。トップダウン設計が望ましい方法であり、既開発の高解像度用フォントさえあればトップダウン設計で済むという見方は、一部当たっているが、問題はそれほど簡単ではない。元来、漢字を32ドットのような低解像度で表現するには、品質以前に字体の表現に無理があり(ゴシック体では40ドットでも十分でない)、デザイナーによるヒューリスティックな設計過程を経て簡略化した字体を実現しているのが実態である<sup>37)</sup>。これをトップダウン設計で自動的に生成するには無理がある。さらに、高解像度から低解像度のビットマップ・パターンに変換する際、前者の解像度が後者の整数倍でないかぎり2値化の誤差が避けられない。ボトムアップ設計では、低解像度のデザイン上の制約が高解像度に引き継がれてしまう問題があり、折衷法は低解像度と高解像度の両者を勘案してフォントのデザインを行うもので、いずれも、高品質フォントの設計法として適当でない。

そこで、以下では、前章で述べたフォント表現の属性記述性がデザイン方針を反映し変換生成能力の高さを表すという考えから、上記の解像度に関する**トップ**

**ダウン設計**を、属性記述度に関する**トップダウン・アプローチ**として再検討する。そのためコンピュータ・メモリ上に字母となるフォントを作成するものとし、そのパターンの形状条件などの属性、変換手続き、編集操作などについて考える。その前に、フォント設計開発の手順と字母フォントを含む開発過程での各種のフォントの関係をみておく。

### 4.2 フォントの設計開発の手順

上記のトップダウン・アプローチを含む今後のフォント設計開発の手順の一案を**図-2**に示す。その中でフォントに関して以下の用語を用いる<sup>43)</sup>。

(1) **原字**: 一式の文字セットの形状パターンについて文字サイズ、線の太さなどを正規化したもの。紙上に表現した**紙上原字**、コンピュータ・メモリ上に表現した**コンピュータ原字**がある(従来、原字と言う場合、上記の正規化以前のものを指すことも多い)。

(2) **コンピュータ字母**: 表現形式、文字サイズ、線の太さなどの変換処理を施すことによって、複数種のコンピュータ・フォントを作成するため、コンピュータ・メモリに保持した一式の文字セットの形状情報。

(3) **コンピュータ・フォント**: コンピュータによ

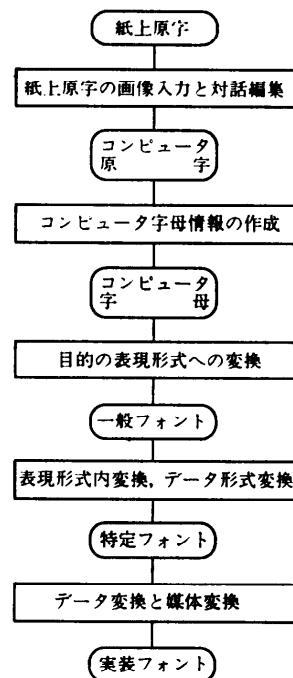


図-2 フォント開発の手順

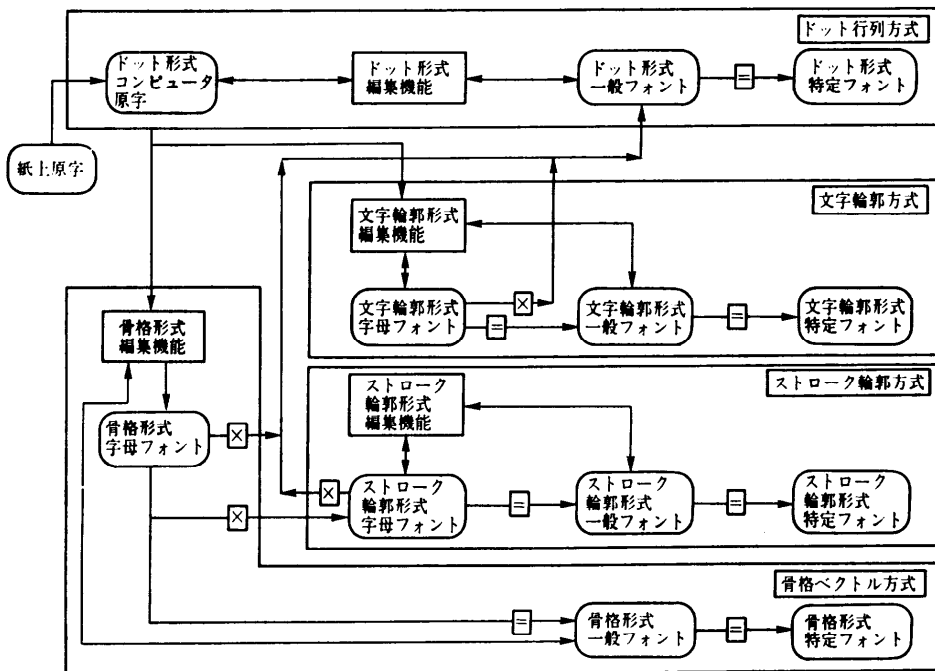
る出力用の文字形状パターンおよび文字サイズなどの配置用の付加情報のセットをメモリ上に保持したもので、以下のフォントに分けられる。

- 一般フォント：特定の表現形式で表されているが、個々の可視化システム固有のデータ形式とは独立な形式で保持されたコンピュータ・フォント。
- 特定フォント：特定の可視化システムに依存したデータ形式で保持されたコンピュータ・フォント。
- 実装フォント：特定の装置に依存したデータ形式で保持されたコンピュータ・フォント。

図-2 では、まず、紙上原字があればそれを画像入力し、ノイズ除去や正規化を経てコンピュータ原字とし、これに基づいて適当な表現形式への変換と属性情報の付加を行ってコンピュータ字母を作成する。これをもとに所要の変形処理を行ったうえで、要求される表現形式に変換し一般フォントを作成する。さらにデータ形式（曲線表現法、圧縮法など）の変換を施して特定フォントを作成する。実装フォントは特定フォントから作られる。この流れの中で、コンピュータ字母の作成法とその形状属性の記述性、および、一般フォントへの変換性が、設計開発フェーズの重要な要素である。

図-3 は、特定フォントとしてドット、輪郭、骨格の各形式を考え、それらを図-2 の手順で開発するフォント設計開発過程で必要な機能と相互関係を例示したものである。本図は、異なる役割や形式のフォントが相互の変換機能をとおして生成される流れを含む。この流れの中で、骨格ベクトル文字は、それが文字の構造や文字形状などの属性記述度が高い形式であるために上流に位置し、直接表現のドット行列文字は最も下流にある。輪郭ベクトル文字はこれらの中間に位置するが、アウトライン文字よりもストローク輪郭文字のほうが、ストローク情報を保持している点でより上流にある。

コンピュータ字母の表現形式は、フォントの含む文字パターンの種類（英字、漢字、記号など）、書体、字母としての利用の頻度、将来にわたる開発保守工数などを含めた検討によって決定される。広く利用されファミリーを構成するようなフォントは、可能なかぎり上流の方式でコンピュータ字母として開発し保持することが望ましい。しかし、骨格ベクトル方式で任意の原字パターンに忠実な形状を与えるのは容易でない。輪郭ベクトル方式では自由なパターンの反映が可能であり、記号や特殊な書体でも、紙上原字さえ用意され



注. [=]: 同一表現形式内の変換 [X]: 異なる表現形式間の変換

図-3 フォント設計開発システム

ば、他の文字と同様の操作でコンピュータ字母の開発が可能である。この方法は字母としての利用頻度の少ない場合でも有効である。ただし現状では、特に漢字について形状属性の記述度が十分でなく、太さや端辺形状の変形方法の検討段階である<sup>24), 29)</sup>。

このようなことから、字母フォントを複数の種類の表現形式で用意し、おのおのの特徴を活かしてフォント設計開発に利用できることが望ましい。

#### 4.3 文字パターン生成における形状属性の実現法

フォント情報に基づいて、その文字パターンを生成したり、他のフォント表現形式に変換したりする場合には、元のフォントのデザイン方針に従う必要がある。そのためには、文字パターンそのものの情報に加えて、フォントおよび各文字パターンのデザイン上の特性や条件が利用できる必要がある。これらがフォント情報の一部に含まれる場合と、文字パターン生成手続き中に組み込まれる場合がある。

このフォントのデザイン上の特性はエレメントで表現されるが、コンピュータでのフォントの設計開発では、これをより具体的な表現にすることが望ましい。エレメントによる骨格ベクトル方式では、このような表現が容易である。図-4のように具体的な端辺角度の条件をもつ10種余のエレメントにより、角ゴシック体の漢字を試作した例がある<sup>4)</sup>。

フォント設計開発においては、このようなエレメントとその基本形状の上に、さらに詳細なフォントの特性として、線の太さの種類とその選択条件、縦線と横線の太さの比率、文字パターンの輪郭矩形の位置とサイズ、曲線ストロークのたわみの度合いなどを規定する<sup>4), 26)</sup>。これらに類する指標をパラメータとして文字パターン生成を試みた例がある<sup>30), 31), 33), 40), 41)</sup>。書体の特性を組み込んだ生成プログラムによって、共通の骨格情報から複数の書体の漢字パターンを生成しメモリ量の低減を実現した報告もある<sup>27)</sup>。この場合、個々の文字パターンに対する制御可能性は低下してしまうが、骨格ベクトル方式の形状属性の記述性を示す一例である。

複数のストロークの接続部に変形規則を設けることが骨格ベクトル方式では必要である<sup>40)</sup>が、ストローク輪郭ベクトル方式でも、あらかじめ用意したストローク輪郭間の接続部を変形する例が報告されている<sup>31)</sup>。

#### 4.4 文字パターン編集機能

ある表現方式の文字パターンおよびその形状特性を入力し、確認し、修正するためのツールが文字パター

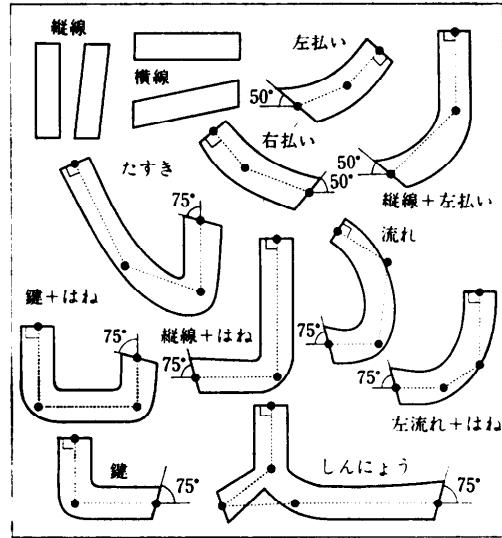


図-4 漢字エレメントのデザイン・ルールの例

ン編集機能である。ディスプレイ画面を用いて文字パターンを表示しつつ対話型で操作するのが普通である。原字があればこれを画面に表示して参照できる機能が望まれる。編集対象の文字パターンを種々のサイズで他の文字と並べて表示する機能があれば効果的である。以下、図-3の表現形式対応の編集機能の特徴を述べる。

##### (1) ドット行列文字の編集

ビットマップの各ドットを編集者が操作して直接表現パターンを得る。このパターンは、操作の便宜上、各ドットを広がりをもった矩形で表示しているため、実際の可視化イメージを与える表示が別途必要である。操作性のために、複数パターンの合成、他のパターンのコピー、部分パターンの拡大縮小などのビットマップ操作のほか、線分や曲線上のドットの制御、矩形や円内のドットの反転などの図形的機能も使われることがある<sup>26)</sup>。

##### (2) 輪郭ベクトル文字の編集

原字を画面に表示し、その輪郭を反映する曲線を作成することによって、原字に忠実な直接表現パターンを得る<sup>11)</sup>。ビットマップ形式の原字パターンから、輪郭線の自動追跡によってアウトラインを生成し、編集機能を用いて完成すると開発効率が上がる<sup>26), 34)</sup>。この方式に特有な編集操作機能としては輪郭線の取り扱いに関するものがある。それらの詳細は、輪郭曲線の表現法に依存する。2本の曲線セグメントの接続部を、頂点とするか、接線を一致させるか、さらに滑ら

かにするかななどの選択が可能な操作機能が考えられる<sup>26)</sup>。なお、文字輪郭、ストローク輪郭文字の編集機能の多くは共通である。

### (3) 骨格ベクトル文字の編集

画面に表示した原字の上に骨格点を指定し、作成中の文字の輪郭や塗りつぶしパターンを原字と比較しながら属性パラメタなどを指定する<sup>14), 40)</sup>。操作者の指定する情報が、文字パターンそのものより、それを生成するための基本パターンと属性や条件であることが特徴である。ストローク・エレメント方式における編集の例<sup>14), 40)</sup>では、操作者が指定する文字パターンらしきものはストロークの骨格点列のみであり、エレメント種別と太さなどの形状パラメタは数値やメニュー項目からの選択として指定する。この形状パラメタには個々の文字のストロークに指定される局所的パラメタと、その省略時に用いられるエレメント種別対応の全体的パラメタとがあり、後者はフォントのデザインを統一する形状を与え、個々の文字の操作者は変更しない。ここでの原字の主な役割はストロークの位置を指定する基準であり、完成した原字は必ずしも必要でない。

## 4.5 文字パターン変換機能

図-3に示したように、フォントの開発では、コンピュータ字母や一般フォントなど、異なるフォント形態間の文字パターンの変換が必要である。変換結果そのままでは利用できず、変換先の形式での編集操作を必要とすることもあるが、その場合でも、デザインの統一性や開発工数の点から変換処理の意義は大きい。

### (1) 同一表現形式内での文字パターンの変換

ドット行列文字でも、サイズ間のデザインの統一のため、基本パターンを拡大縮小し修正して別サイズの文字パターンを作成する方法が採られる。

輪郭ベクトルフォント内では、コンピュータ字母から可視化装置の方式を反映する一般フォントの生成に際して、サイズ(精度)や輪郭曲線種別の変換が必要となる。後者の例としては、スプライン曲線や円弧からベジェ曲線へ、あるいは、曲線列から線分列へなどの変換がある。ただし、これらの間の双方向の変換は必ずしも可能でないことに留意する必要がある。輪郭ベクトル文字のストロークの太さの変更には特別の配慮を要するが、文字パターンの図形単位に変更をもたらさない条件の下で、線幅の異なる2種のパターン間の内外挿処理で別の線幅の輪郭ベクトル文字を生成する試みがある<sup>24)</sup>。

骨格ベクトル方式でも、太さの異なるパターンの骨格と端辺角度や太さパラメタを設定し、これらの内外挿で種々の太さのパターンを生成した例がある<sup>35)</sup>。ここではストローク輪郭を生成した後にアウトライン文字に変換するため、上記のアウトライン文字における太さ範囲の制約はない。骨格ベクトル方式ではその形状パラメタの変更方法が重要であり、この例のように2点を設定して他を内外挿する方法のほか、なんらかのルールでこれを表現することもある。

### (2) 異なる表現形式間での文字パターンの変換

拡大縮小品質の良いベクトル文字の字母から各種サイズのドット・フォントを変換して開発する効果は大きい。この際、ドット行列形式への2値化変換処理の誤差により線幅や線間の不揃い、カスレ、ツブレなどの問題が生じ得る。そのため、アウトライン文字からの変換アルゴリズムに工夫した例があるが<sup>28)</sup>、根本的な解決は表現属性の記述<sup>23), 39)</sup>による必要があろう。

次に、骨格ベクトル文字をコンピュータ字母として用いた場合、その形状パラメタの設定によって得られる種々の変形結果を輪郭ベクトル文字やドット文字に変換して特定フォントを作成できれば、フォント開発の効率を向上できる。骨格ベクトル文字からドット文字パターンへの変換は容易であり、ストローク輪郭文字パターンはその途中で作られる。骨格ベクトル文字から輪郭形状を線分列として表現するアウトライン文字への変換例がある<sup>32)</sup>。先に述べたアウトライン文字表現間の線幅の内外挿方式では除外している範囲の線幅の変更でも、通常どおり太さパラメタを指定してストローク輪郭を生成した後にアウトライン文字に変換することが可能である<sup>40)</sup>。JIS第1水準<sup>10)</sup>の全体2965字について384ドット角の骨格情報から太さ変形済みの256ドット角のアウトライン文字が、大型機により1時間の待ち時間で生成されている<sup>32)</sup>。

## 5. フォントに関する今後の課題

### (1) フォントの形状属性とフォント設計開発

フォントの設計開発および保守のためには、フォントを統一する形状特性や個々の文字パターンの形状特性を定量的に把握することが必要である。英字においては、ベースラインやx-height(小文字の基準の高さ)などラインシステムの考え方やウェイト(x-heightに占める垂直線の太さの割合)、コントラスト(小文字の水平線の太さに対する垂直線の太さの割合)などの概念が明確である<sup>26)</sup>。日本語の文字についても、黒



地面積比, 文字輪郭面積, 外接矩形, 縦横線幅比率など種々の指標が文字デザインその他の分野で検討されており<sup>4), 16), 36)</sup>, フォント開発の実際の場合では個別の工夫とともにこれらに類する指標が用いられていると考えられる。しかし, 実際のフォントについて定量的データを取得し, 一般性のある有効な指標を求め, 字母のもつべき属性を考え, これをフォントの設計開発および保守に活かすようなシステムティックな技術は研究段階にあると言える。線の太さを文字サイズ<sup>40)</sup>や文字の線長合計<sup>30)</sup>との関係で制御するなど, 定量的指標に基づくフォント・デザインの制御の試みが報告されている<sup>33)</sup>。なお, デザイン形状の属性のみでなく, エレメントの分布のような字体に関する属性データ<sup>36)</sup>の把握も開発保守の支援に役立つ。

### (2) 文字列配置のためのフォント属性

文書処理システムにおいて, フォント情報は可視化機能と割付機能から共通の情報として参照される。この場合のフォント情報とは, フォント内の字種間に共通の形状属性, 各字種の形状属性, および, 個々の文字パターン本体を含む。文字パターン本体は可視化処理のみに関係するが, 他の情報は割付処理でも利用される。割付処理はこれらの属性によって文字列や行の配置を決定し, 可視化処理も同じ属性情報によって決定した位置に文字パターン列を可視化する必要がある。英字の横幅は文字ごとに異なり, 文字列としての配置のためのピッチ情報が形状属性として保持される。日本語の文字では, 字種に共通に文字枠が保持されるのが普通である。しかし, 日本語 DTP では, 漢字仮名英数字の混在する文字列の配置が必要であり, 日本語の文字と英数字および記号との間のピッチや水平ラインの関係を適切に設定する必要がある。

### (3) フォント属性記述度に関するボトムアップ変換

トップダウン変換による形状属性の記述度の低いフォントの開発の有効性を前章で述べた。この変換元である形状属性の記述度の高い字母フォント自体は, その表現形式に依存した編集機能によって作成されるが, その品質と作成効率の向上手段として, 属性記述度に関するボトムアップ変換が試みられている。これは既存の直接表現パターンを利用してその属性記述を抽出する機能を含み, この機能を本来の対話型編集機能に組み込むものが多い。この中でドット行列形式のコンピュータ原字から輪郭を抽出してアウトライン・パターンを得ることは一般化しており, 編集機能とは

独立に行われることも多い<sup>34), 42)</sup>。ドット行列文字パターンからそのストロークの骨格線(中心線)と太さなどの形状属性を取り出す試みがある<sup>22), 30)</sup>。線ストローク形式の文字パターンの直線表現をベジェ曲線表現に変換した例がある<sup>42)</sup>。このボトムアップ・アプローチについては, 適用範囲の拡大と属性記述の自動化率の向上を含めて, 今後, より深く研究されるべき課題が多い。

### (4) フォントの品質の評価

これまでフォントの開発工数について言及したが, 実際には, 文字間のデザインの統一を含むフォントの品質と独立に開発工数を比較しても意味がない。しかし, 現状では, フォントの品質自体を比較評価するための客観的な方法が確立していない。経験と美的感覚に優れたデザイナーにとっても, 6000字を越える字種のデザインのバランスを判断することは容易でない。上記のフォント形状属性や文字列配置属性の検討, および, 種々の実測データの取得と管理検索などの機能と, それらを組み込んだデザイナーのためのツールシステムが, 品質評価の支援の基礎として必要であり, これが評価の客観化にも導くものと考えられる。

### (5) その他の課題

その他の検討課題について以下に列挙する<sup>26)</sup>。

- **デザイナーの知識から工学的知識へ:** デザイナーの経験と知識をコンピュータで利用可能な形に整理する。

- **デザイナーのフォント設計ツール:** 新規のデザインを生み出す過程での支援ツールである。

- **人間工学的配慮:** たとえば, ユーザが長時間見る必要のあるディスプレイ装置上の可視化フォントは, 一般の品質の観点でなく, 見やすいフォント, 疲れないデザインなどの観点から検討する必要がある。

- **文書の品質とフォント:** 文書の種類や用途とフォントおよびその使用法(書体, サイズ, 文字ピッチ, 行ピッチなど)との関係を明らかにすること。

## 6. おわりに

字種の多い漢字を含む日本語フォントの多様化高品質化の傾向を念頭に, フォント技術の現状と課題を記した。可視化と設計開発の両フェーズでのフォントの観点を区別し, 特に後者の観点からフォント技術の要件をあげた。フォント表現方式の研究開発例を分類し, それらの特徴と問題点を述べた。この中で文字パターン情報における属性記述性に注目した。今後の日本語フォントの設計開発システムのモデルを示し, 各

種フォントの役割と、文字パターンの編集変換などの諸機能について考え、属性記述性の高いフォントを字母とするトップダウン・アプローチの将来性を述べた。

フォントの形状属性の定量化とフォント設計開発への適用、および、文字列配置のための属性については、英字の分野では研究開発が進んでいるが、日本語フォントについては今後検討すべき要素が多い。その他の問題とともに、フォント技術の今後の課題として述べた。

最後に、本稿で用いた術語の中には、その意味と用法が確定していないものもあることに注意したい。本稿で新規に定義した用語も多い。フォント技術を広い視野でみた上で、概念と用語を整理することも、今後の課題の一つである。

謝辞 当社のデザイン研究所の古川主任デザイナーと中村直喜氏からは、文字パターンの表現例を提供し、実際のフォント設計開発状況について御教示いただいた。中央研究所の下位憲司氏と国西元英氏からは、原稿に関して意見をいただき、文字パターンの表現例の作成に協力していただいた。以上の各位に感謝します。

### 参 考 文 献

- 1) 坂井、長尾、寺井：部分パターンによる漢字の合成、情報処理、Vol. 10, No. 5, pp. 285-293 (1969).
- 2) Frank, A. J.: Parametric Font and Image Definition and Generation, FJCC, pp. 135-144 (1971).
- 3) Hershey, A. V.: A Computer System for Scientific Typography, CG&Im Proc., 1, pp. 373-385 (1972).
- 4) 佐藤：漢字・上, 下一文字のデザインシリーズ 5, 6—, 丸善 (1973, 1976).
- 5) 吹抜：漢字パタンのデータ圧縮の一方式—Modified Stroke 方式—, S 53 信学会総全, 1109 (1978).
- 6) Coueignoux, P. and Guedj, R.: Computer Generation of Colored Planar Patterns of TV-Like Rasters, Proc. IEEE, Vol. 68, No. 7, pp. 909-922 (1980).
- 7) Nagao, M.: Data Compression of Chinese Character Patterns, Proc. IEEE, Vol. 68, No. 7, pp. 818-829 (1980).
- 8) Coueignoux, P.: Character Generation by Computer, CG & Im Proc., 16, pp. 240-269 (1981).
- 9) Hobby, J. and Guoan, G.: A Chinese Meta-Font, ICTP '83, pp. 62-67 (1983).
- 10) JIS C 6226 (現在, X-0208) 情報交換用漢字符号系, 日本規格協会 (1983).
- 11) Flowers, J.: Digital Type Manufacture: An Interactive Approach, Computer, May, pp. 40-48 (1984).
- 12) 張, 真田, 手塚: 漢字楷書毛筆字体の計算機による生成, 信学論, Vol. J67-D, No. 5, pp. 599-606 (1984).
- 13) 菊池, 大山, 高橋: 字体のパラメトリック基本エレメント貼付方式による高品質漢字フォント生成方式, 第 29 回情報処理学会全国大会 3J-7, pp. 1435-1436 (1984).
- 14) 国西, 菊池, 大山, 伊藤, 上原, 桑原: 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 3J-8, pp. 1437-1438 (1984).
- 15) 大山, 菊池, 高橋: 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 3J-6, pp. 1433-1434 (1984).
- 16) 吉村: 手書き文字の個人特性, 大澤編「文字の科学」法政大学出版局, pp. 93-147 (1985).
- 17) 坂元, 高木: 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).
- 18) 大町: レタリングエッセンス, 日本文芸社 (1985).
- 19) Adobe Systems: POSTSCRIPT Reference Manual, Addison-Wesley (1986).
- 20) Knuth, D. E.: The TeX Book, Addison-Wesley (1986).
- 21) 梶田, 中村, 島, 戸倉: 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 3T-7, pp. 1665-1666 (1986).
- 22) 中村, 梶田, 島, 戸倉: 毛筆書体の字形パラメータ抽出アルゴリズム, 第 32 回情報処理学会全国大会 3T-8, pp. 1667-1668 (1986).
- 23) Karow, P.: Digital Formats for Typefaces, URW Verlag (1987).
- 24) 山崎, 入山: 高精細デジタル文字図形の変倍の一方法, 信学論, Vol. J70-D, No. 2, pp. 387-397 (1987).
- 25) 日本語デスクトップ・パブリッシング, 日経エレクトロニクス, No. 435, pp. 128-141 (1987).
- 26) Rubinstein, R.: Digital Typography, Addison-Wesley (1988).
- 27) 東: 日本語フォントクリエイター, 卓上出版シンポジウム報告集, pp. 91-96 (1988).
- 28) 戸張: 高品位フォントのプリントクオリティ, 卓上出版シンポジウム報告集, pp. 119-125 (1988).
- 29) 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).
- 30) 荻原, 中沢, 油田, 中島: ゴシック体超大形日本語文字の自動生成, 信学論, Vol. J72-D-II, No. 7, pp. 1048-1055 (1989).
- 31) 陳, 小沢: 多様な明朝体文字の規則的な生成,

- 信学論, Vol. J72-D-II, No. 9, pp. 1423-1431 (1989).
- 32) 下位, 上原: 骨格ベクトル文字からアウトライン文字への変換方式, 情報処理学会論文誌, Vol. 30, No. 9, pp. 1111-1118 (1989).
- 33) 内尾, 樋口, 北橋, 真田, 手塚: 毛筆楷書文字の感覚的大きさの一正規化手法, 信学論, Vol. J72-D-II, No. 10, pp. 1650-1656 (1989).
- 34) 西川, 直井: 輪郭表現による高品質文字パターン生成方式, 信学論, Vol. J72-D-II, No. 12, pp. 2023-2031 (1989).
- 35) 荻原, 中沢, 油田, 中島: 内外挿を用いたゴシック体日本語超大形文字の自動生成, 信学論, Vol. J72-D-II, No. 9, pp. 1388-1396 (1989).
- 36) 上原, 鍵政, 国西, 下位: 骨格ベクトル方式による漢字フォントの形状特性, 信学論, Vol. J72-D-II, No. 11, pp. 1807-1815 (1989).
- 37) 田嶋: コンピュータと漢字, 佐藤編「漢字と国語問題」明治書院, pp. 229-257 (1989).
- 38) ISO/IEC JTC1/SC 18/WG 8: Information Processing-Font Information Interchange Part 1: Architecture, ISO/IEC/DIS 9541-1 (1990).
- 39) ISO/IEC JTC 1/SC 18/WG 8: Proposed Working Draft For 9541-3, WG 8-N 1038 (1990).
- 40) 上原, 国西, 下位, 鍵政, 菊池: ストローク種別に基づく漢字形状生成方式, 情報処理学会論文誌, Vol. 31, No. 2, pp. 209-218 (1990).
- 41) 国西, 根本, 上原, 下位, 鍵政: 骨格ベクトル方式による仮名英数字の生成, 第40回情報処理学会全国大会 7P-5, pp. 623-624 (1990).
- 42) 斉藤, 穂坂: 拡張2次有理 Bézier 曲線による高品位文字フォントの生成とその特徴, 情報処理学会論文誌, Vol. 31, No. 4, pp. 562-570 (1990).
- 43) 上原, 下位, 国西, 鍵政: 骨格ベクトル方式による漢字パターンの生成(1)方式の特徴とフォント開発への適用性, 第38回情報処理学会全国大会 1K-6, pp. 658-659 (1989).
- 44) 国西, 坂本, 多田, 上原, 下位, 鍵政: 同上(2)輪郭形状ルールに従った漢字ストロークの生成, 第38回情報処理学会全国大会 1K-7, pp. 660-661 (1989).
- 45) 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).

(平成2年6月12日受付)