

## 報 告

## 計算機による標準文字図形の発生\*

山崎 一生\*\* 飯島 泰蔵\*\*

## 1. まえがき

文字認識方式の研究や、印字文字の品質評価法の研究、さらには光学文字認識用の印字字体の設計などを行なう際に、標準文字をいかにして作成するか、あるいはいかにして計算機に入力するかということが、一つの大きな問題となってくる。手間さえいとわなければ、文字の拡大図を描き、サンプリングしたい格子点間隔の碁盤目状のわくをあてがって、格子点が文字線の黒部分の中にあるか否かによって、あるいは昇目に占める黒部分の面積が 50% を越えるか否かによって「1」、「0」の二値に量子化して、白黒二値の文字図形を得ることができるが、手作業でこのようなことを行なうと、「1」、「0」のどちらに判定するかの基準が不明確となるばかりでなく、作業の段階で誤りをおかす可能性が大きく、所要時間もばう大なものとなる。そこで、このような作業を計算機の中で行なわせることを考察することにしよう。計算機による文字図形の発生については広川、葛城の発表<sup>1)</sup>があるが、その詳細は明らかでない。

ここでは、定規とコンパスとを使用して文字図形を描くことを意識して設計された字体である「ラインプリンタによるかたかな字体」<sup>2)</sup>あるいは OCR-A<sup>3)</sup>、このようなことは考慮しないで、光学文字認識用として美学的な見地から ECMA<sup>\*1</sup> で設計した OCR-B<sup>3)</sup> の計算機による発生方法について述べる。後者の場合には、文字線の中心線の座標が適当な間隔で与えられるので、これらの座標点を直線あるいは円弧で、ほぼなめらかにつないで文字図形を発生していく。

## 2. 文字を描くためのデータ

文字を描くためのデータの与えられ方には、大別すると 2 種類がある。一つは定規とコンパスを使って

文字図形を描くことを意図したもので、文字図形を構成するストローク（文字線）の中心線は直線と円弧とで表現されていて、この中心線の両側に等しく肉づけすることによって、文字図形が描かれる。もう一つは、文字線の中心線の座標がある適当な間隔で与えられる場合で、これらの座標点をなめらかにつないで中心線を描き、この中心線の両側に等しく肉づけして文字図形が描かれる。いずれの場合にも、文字線の端点は文字線幅に等しい大きさの直径の円弧で丸められる。前者の方法で描かれる字体には「ラインプリンタによるかたかな字体 (JEIDA-5)」, ISO<sup>\*2</sup> から勧告され JIS 規格ともなっている OCR-A があり、JIS 原案として現在設計中の OCR 用かたかな字体も、この方法にそったものである。また、後者の方法で描かれる文字には、OCR-A と同時に JIS 化された OCR-B がある。OCR-B には衝撃印字用の一定線幅の文字と、活版印刷用の非定線幅の文字とがあるが、ここでは非定線幅の文字は考えないことにする。

## 2.1 定規・コンパス型

ラインプリンタによるかたかな字体の場合には、Fig. 2.1 と Table 2.1 とに示すように水平・垂直方

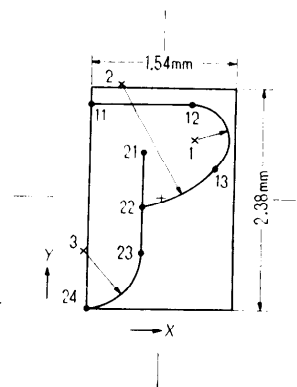


Fig. 2.1 ラインプリンタによるかたかな字体の例  
An example of KATAKANA character.

\* Generation of Standard Character Figure with Computer, by Issei Yamasaki and Taizo Iizima (Iijima Special Section, The Electrotechnical Laboratory)

\*\* 電子技術総合研究所飯島特別研究室

\*1 European Computer Manufacturers Association

\*2 International Organization for Standardization

Table 2.1 ラインプリンタによるかたかな  
字体の数表の例  
An example of table of KATAKANA  
character.

SIGN	X	Y	R
1	(1.09)	(1.82)	0.38
2	(0.31)	(2.40)	1.34
3	(-0.04)	(0.60)	0.60
11	0.00	2.20	
12	1.09	2.20	
13	1.32	1.52	
21	0.56	1.68	
22	0.56	1.08	
23	0.56	0.58	
24	0.00	0.00	

単位: mm

向にそれぞれ X, Y 軸をとって, 直線の場合には二つの端点の座標が, 円弧の場合には二つの端点と曲率中心の座標, および曲率半径が与えられる. Fig. 2.1 において, 一桁の数字は曲率中心を表わし, 二桁の数字は文字線の中心線上の点で, 各線素の端点を表わす.

2.2 座標表示型

OCR-B の場合には水平・垂直方向に, それぞれ X, Y 軸をとって, Fig. 2.2 と Table 2.2 とに示すように, 文字線の中心線の座標が適当な間隔で与えられる. Fig. 2.2 において, 一桁の数字は文字の基準位置を表示するためのもの, 二桁の数字は中心線上の座標を表示するためのもの, 三桁の数字は非定線幅字体の

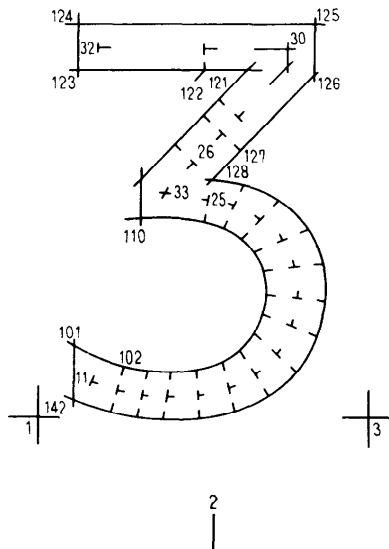


Fig. 2.2 OCR-B の例  
An example of OCR-B character.

Table 2.2 OCR-B の数表の例  
An example of table of OCR-B  
character.

N	数字 3				
	Y1	X12	Y2	X3	Y3
1	2000.	809.	2000.	707.	2000.
2	1338.	2000.	1228.	2000.	1118.
3	2000.	3029.	2000.	3117.	2000.
11	2234.	1179.	2246.	1108.	2312.
12	2181.	1342.	2184.	1286.	2241.
13	2148.	1519.	2146.	1478.	2197.
14	2128.	1679.	2122.	1651.	2170.
15	2142.	1892.	2138.	1883.	2189.
16	2192.	2095.	2197.	2104.	2256.
17	2289.	2259.	2310.	2281.	2385.
18	2429.	2415.	2474.	2451.	2572.
19	2619.	2508.	2696.	2552.	2825.
20	2812.	2541.	2921.	2587.	3083.
21	3026.	2517.	3170.	2562.	3368.
22	3212.	2437.	3388.	2475.	3616.
23	3345.	2290.	3543.	2315.	3794.
24	3432.	2110.	3644.	2120.	3909.
25	3482.	1933.	3702.	1928.	3975.
26	3715.	1836.	3974.	1822.	4286.
27	3912.	2026.	4204.	2028.	4549.
28	4022.	2129.	4332.	2140.	4696.
29	4372.	2462.	4741.	2502.	5163.
30	4486.	2459.	4873.	2498.	5314.
31	4485.	1905.	4872.	1897.	5313.
32	4487.	1189.	4875.	1119.	5316.
33	3524.	1653.	3752.	1624.	4033.

(以下省略)

単位: ミクロン

標準文字線幅の場合の輪郭の座標を表示するためのものである. また, Table 2.2 において, Y1, X12, Y2, X3, Y3 なる記号が表われるが, これは文字の大きさ (サイズ) に 3 段階があるからであって, サイズ I の場合には (X12, Y1) を, サイズ II の場合には (X12, Y2) を, サイズ III の場合には (X3, Y3) を座標値として用いる.

3. 文字線と線素

ラインプリンタによるかたかな字体的場合, 文字を構成する線の「一画」をストロークあるいは文字線と呼び, 文字線を構成する直線あるいは曲線のことを線素と呼ぶことにしよう. たとえば Fig. 2.1 の文字図形「ア」は 11-12-13-22 と 21-22-23-24 なる二つの文字線から構成され, 前者の文字線は 11-12, 12-13, 13-22 の 3 つの線素から構成され, 後者は 21-23, 23-24 なる 2 つの線素から構成される.

OCR-B の場合右まわりの曲線を描くような座標点の集りをストロークあるいは文字線と呼び, 文字線を構成する素片のことを線素と呼ぶことにする. たとえ

ば Fig. 2.2 の文字「3」の場合には、30-31-32-29-30, 33-26-27, 27-28-29, 33-25-24……-12-11 の5つの文字線から構成され、各文字線はそれぞれ 2, 1, 2, 2, 14 個の線素から構成される。

3.1 直線型線素

Fig. 3.1 に示すように  $x, y$  軸をとり、与えられる2つの座標点  $P_1, P_2$  の座標をそれぞれ  $(X_1, Y_1), (X_2, Y_2)$  とし、文字線幅を  $2w$  とすれば、文字線の

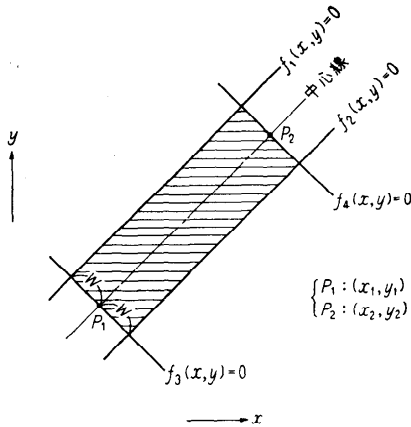


Fig. 3.1 直線型線素  
Straight segment.

黒部分は4つの直線  $f_1, f_2, f_3, f_4$  でかこまれる領域となる。4つの直線の方程式は

$$\begin{cases} f_1(x, y) = (y - Y_1)(X_1 - X_2) - (x - X_1)(Y_1 - Y_2) - w\sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} = 0 \\ f_2(x, y) = (y - Y_1)(X_1 - X_2) - (x - X_1)(Y_1 - Y_2) + w\sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} = 0 \\ f_3(x, y) = (X_1 - X_2)(x - X_1) + (Y_1 - Y_2)(y - Y_1) = 0 \\ f_4(x, y) = (X_1 - X_2)(x - X_2) + (Y_1 - Y_2)(y - Y_2) = 0 \end{cases} \quad (3.1)$$

で表わされ、黒部分は不等式

$$\begin{cases} f_1(x, y) \leq 0 \\ f_2(x, y) \geq 0 \\ f_3(x, y) \leq 0 \\ f_4(x, y) \geq 0 \end{cases} \quad (3.2)$$

が同時に満足される領域である。

3.2 円弧型線素

(1) 曲率中心・半径が与えられる場合

Fig. 3.2 に示すように、与えられた2つの端点  $P_1, P_2$  と曲率中心  $O$  との座標をそれぞれ  $(X_1, Y_1), (X_2, Y_2), (X_0, Y_0)$  とし、曲率半径を  $R$  とすれば、線幅  $2w$  の円弧型線素の黒部分は2本の直線  $g_1, g_2$  と2

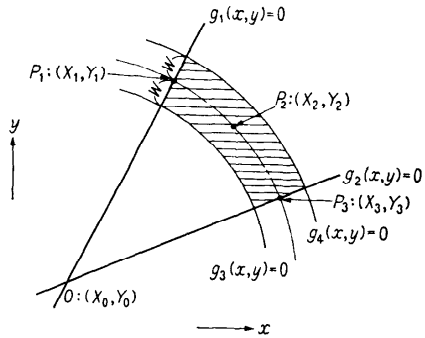


Fig. 3.2 円弧型線素 (1)  
Circular segment (1)

つの円弧  $g_3, g_4$  とでかこまれる領域となる。2つの直線  $g_1, g_2$  と2つの円弧  $g_3, g_4$  との方程式は

$$\begin{cases} g_1(x, y) = (y - Y_1)(X_0 - X_1) - (x - X_1)(Y_0 - Y_1) = 0 \\ g_2(x, y) = (y - Y_2)(X_0 - X_2) - (x - X_2)(Y_0 - Y_2) = 0 \\ g_3(x, y) = (x - X_0)^2 + (y - Y_0)^2 - (R - w)^2 = 0 \\ g_4(x, y) = (x - X_0)^2 + (y - Y_0)^2 - (R + w)^2 = 0 \end{cases} \quad (3.3)$$

で表わされ、黒部分は不等式

$$\begin{cases} g_1(x, y) \geq 0 \\ g_2(x, y) \leq 0 \\ g_3(x, y) \geq 0 \\ g_4(x, y) \leq 0 \end{cases} \quad (3.4)$$

が同時に満足される領域である。円弧の描き方が反時計の場合 (Fig. 3.2 で  $P_1$  と  $P_2$  とを入れ替えたもの) には式 (3.4) の上の2つの式の不等号の向きが逆になる。また、円弧が時計まわりに半円以上描かれる場合には、文字線の黒部分は式 (3.4) の上の2つの不等式の中のどちらかが成立して、しかも、下の2つの不等式が同時に成立する場合として表わされる。

(2) 座標点が3つ与えられる場合

Fig. 3.3 に示すように、与えられた3つの点  $P_1, P_2, P_3$  の座標をそれぞれ  $(X_1, Y_1), (X_2, Y_2), (X_3, Y_3)$  とすると、線幅  $2w$  の円弧型線素の黒部分は上の (1) の場合と同様に、2つの直線  $g_1, g_2$  と2つの円弧  $g_3, g_4$  とでかこまれる領域となる。これらの直線型、円弧型線素は式 (3.3) と全く同じ型式で表わされるが、円弧の曲率中心  $O: (X_0, Y_0)$  と曲率半径  $R$  とが与えられた点  $P_1, P_2, P_3$  の座標で表現される。すなわち

$$\begin{cases} X_0 = \beta / (2\alpha) \\ Y_0 = -\gamma / (2\alpha) \\ R = \sqrt{(4\delta - \beta^2 + \gamma^2) / (2\alpha)} \end{cases} \quad (3.5)$$

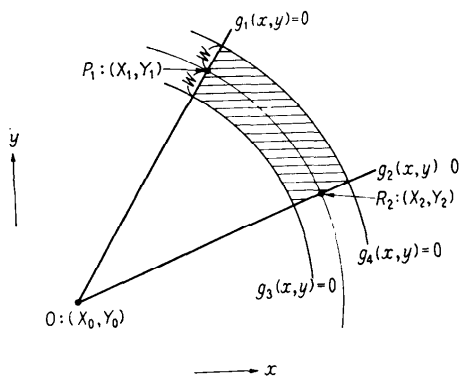


Fig. 3.3 円弧型線素 (2)  
Circular segment (2)

ただし

$$\begin{cases} \alpha = (X_1 - X_2)(Y_2 - Y_3) - (X_2 - X_3)(Y_1 - Y_2) \\ \beta = (Y_2 - Y_3)(X_1^2 + Y_1^2 - X_2^2 - Y_2^2) \\ \quad - (Y_1 - Y_2)(X_2^2 + Y_2^2 - X_3^2 - Y_3^2) \\ \gamma = (X_2 - X_3)(X_1^2 + Y_1^2 - X_2^2 - Y_2^2) \\ \quad - (X_1 - X_2)(X_2^2 + Y_2^2 - X_3^2 - Y_3^2) \\ \delta = (X_1^2 + Y_1^2)(X_2 Y_3 - X_3 Y_2) \\ \quad - X_1(Y_3(X_2^2 + Y_2^2) - Y_2(X_3^2 + Y_3^2)) \end{cases} \quad (3.6)$$

である。文字線の黒部分を表わす不等式は上の (1) の場合と全く同様である。

### 3.3 点型線素

この場合には一点が与えられるから、この点を中心として、文字線幅  $2w$  と等しい直径を有する円を描けば、円の内部が「文字線」の黒部分を表わすことになる。Fig. 3.4 に示すように、与えられる点  $P$  の座標

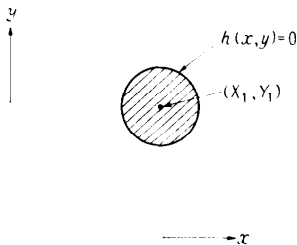


Fig. 3.4 点型線素  
Spot segment.

を  $(X_1, Y_1)$  とすれば、半径  $w$  の円の方程式  $h$  は

$$h(x, y) = (x - X_1)^2 + (y - Y_1)^2 - w^2 = 0 \quad (3.7)$$

となり、文字線の黒部分は不等式

$$h(x, y) \leq 0 \quad (3.8)$$

を満足する領域として与えられる。

## 4. かたかなの発生

Fig. 2.1, Table 2.1 に示すようなデータが与えられた場合、人間はこれらの図と表とを見ながら、文字図形を作図することができる。一方、計算機で自動的に文字を発生するためには、Table 2.1 に与えられる座標、曲率半径などの情報のほかに、ある線素の端点が文字線の端点、屈折点となっているか否か、あるいはつぎの線素となめらかにつながる点か否かなどの情報が必要となってくる。また、円弧型線素の場合には 3.2 で述べたような理由から、円弧が時計まわりか反時計まわりかの情報とともに、半濁点のように半円を越す円弧を描く場合には、これらのことを情報として付加しておくことが必要となる。

Fig. 2.1 において、点 11, (22), 21, 24 はいずれも端点であり、点 12 はなめらかに接続する点であり、点 13, 23 は屈折点である。文字線の端点では文字線幅と等しい直径の円弧で丸める。Fig. 4.1 の作図例を見ればわかるように、屈折点では文字線部分の欠け

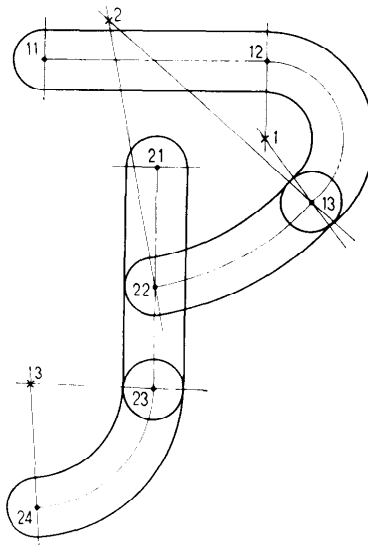


Fig. 4.1 作図例  
An example of drawing.

(Fig. 4.1 の点 13, 23 の近傍の陰影部) が起こるので、屈折点でも文字線幅と等しい直径の円弧で丸めることが必要となってくる。

計算機によって文字図形を発生するには、まず、上のような付属的な情報を伴った一文字分のデータを計算機に入力しておく。つぎに、サンプリング点を逐次

動かしていった、そのサンプリング点が各線素ごとにできる 3. で与えた形式の不等式を満足する点であるか否かを判定し、満足していればサンプリング点の値

を「1」、満足していなければ「0」とするようにして、文字図形の与えられている領域全体をサンプリングすれば、自動的に二値化された文字形を得ることができる。

Fig. 4.2 に計算機によってかなもじを自動的に発生するプログラムの概略の流れ図を示す。Fig. 4.3 に計算機によって発生した標準線幅 (0.35 mm) の文字「ア」を示す。縦横比が 1:1 となるように、水平・垂直方向のサンプリング間隔の比が、ラインプリンタの印字ピッチと改行ピッチとの比に等しくとってある。水平・垂直方向のサンプリング間隔は、それぞれ 0.024 mm, 0.040 mm である。

### 5. OCR-B の発生

Fig. 2.2, Table 2.2 に示したような、OCR-B 文字に関するデータが与えられたとき、人間の場合には図と表とを対照しながら、与えられた座標点をプロットしてゆき、これらの点を適当になめらかにつないで文字線の中心線を描き、つぎにこれらの中心線の両側に等量だけ肉づけし、ストロークの端を線幅と等しい直径を有する円弧で丸めて文字図形を描くことができる。一方、計算機は図面を参照できないので、与えられた各座標点が文字線の中心点の中でいかなる状態にあるのかという情報——たとえば、ストロークの端点になっているとか、途中のなめらかにつながる点であるとかといった情報——が必要となってくるほか、ある座標点とつぎの座標点とを直線でつなぐか否かという判定も必要となってくる。

ここでは、与えられた文字線の中心線上の座標点を直線あるいは円弧でほぼなめらかに順次つないでいって、この中心線に肉づけして文字図形を自動的に発生する方法を考察しよう。

#### 5.1 計算機用の文字図形データ

文字線の中心線に関するデータは、第一線素用に原則として三点の座標値がまず与えられ、これに続く座標点は一点ずつ逐次与えられるものとする。第一線素が点型線素の場合には 1 つの座標点のみが、直線型線素の場合には 2 つの座標点、円弧型線素の場合には 3 つの座標点と与えられる。また、各点の座標は時計まわりの曲線を描くように与えられるものとし、半円以上の円弧を描くような円弧型線素はないものとする(このように約束することによって、円弧型線素を表現する不等式を一組だけにすることができるので、プログラムを簡略化することができる。3. 参照)。

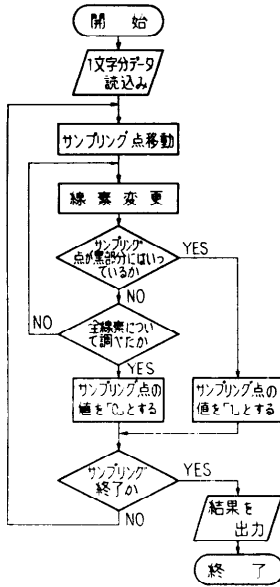


Fig. 4.2 かなの発生の流れ図  
A flow chart of KATAKANA character generation.

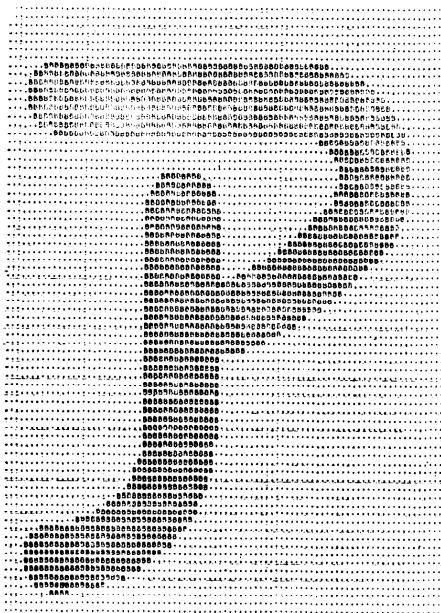


Fig. 4.3 かなの発生例  
An example of KATAKANA character generated.

たとえば, Fig. 2.2, Table 2.2 に示されている文字 3 の場合には, つぎに示すような順序で座標値が計算機に与えられる.

((32, 31, 30), (ES))((30, 29), (ES))((29, 28, 27, 26, 33), (ES))((33, 25, 24), (23), (22), ..., (12), (11), (ES))

ここで (ES) は文字線の終了を表わす目印で, 小括弧内のデータは同時に与えられるものとする. 小括弧内では順序を入れ替えることはできないが, 大括弧を単位として相互に入れ替えても差し支えない.

5.2 直線でつなぐか円弧でつなぐかの判定

第一線素の場合には, 与えられる座標点の数に従って, 3. で述べた点型線素・直線型線素・円弧型線素を表わす不等式によって発生できるが, 第一線素用データに続いてつぎつぎに座標点が与えられる場合には, 与えられたつぎの座標点を直線型線素でつなぐか, 円弧型線素でつなぐかの判定をまず行なう必要がある.

Fig. 5.1 において, 点 A: (XA, YA), B: (XB, YB), C: (XC, YC), D: (XD, YD) を文字線の中

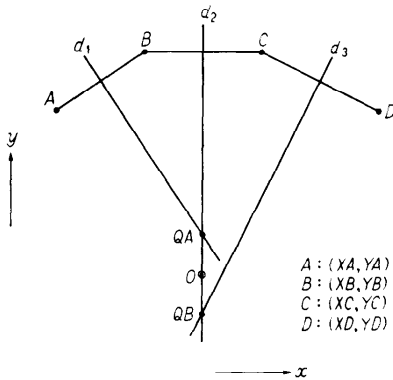


Fig. 5.1 座標点 B, C を通る円弧の曲率中心 O の決め方

Determination of the center O of curvature connecting the points B and C.

心線上の途中の座標点であるとし, 二点 B, C をつなぎたい 2 つの座標点であるとしよう. いま, 線分 AB, BC, CD の垂直二等分線をそれぞれ d1, d2, d3 とし, d1 と d2 との交点を QA: (XQA, YQA), d2 と d3 との交点を QB: (XQB, YQB) とする,

そこで, これら 2 つの交点 QA, QB がともに存在する場合には, 座標点 B, C を円弧型線素でつなぐこととし, 円弧の曲率中心 O を二点 QA, QB の中点に選ぶことにする. 他方, 2 つの交点 QA, QB が存在しない場合(存在しても座標点 B, C からある程度

以上隔っている場合も含む) には, 座標点 B, C を直線型線素で結ぶこととする.

Fig. 2.2, Table 2.2 に示したように与えられる OCR-B の場合, 与えられた座標点を直線あるいは円弧でなめらかにつないで文字図形を発生することは, 一般には不可能であるが, 上に示したような手順で, 円弧型線素の場合の曲率中心を定めることによって, 完全になめらかではないが, ほぼなめらかに各座標点をつないで文字図形を発生することができる. この方法はつなぎたい二点のさらに外側の二点の相対位置関係を考慮しているので, 1 本のストロークを描くときの動きが慣性を有するものであるということも考慮に入れていることになる.

直線・円弧型線素のうちのどちらでつなぐかということ, 上のように 2 つの垂直二等分線が交点を有するか否かで判定することは, 計算機の取り扱いうる数値が有限であることなどから不可能であるので, 実際にはつぎに示すような方法で, この判定を行なっている.

Fig. 5.1 において, 線分 AB, BC, CD のこう配を s1, s2, s3 とすれば

$$\begin{cases} s_1 = (YA - YB) / (XA - XB) & (XA \neq XB) \\ s_2 = (YB - YC) / (XB - XC) & (XB \neq XC) \\ s_3 = (YC - YD) / (XC - XD) & (XC \neq XD) \end{cases} \quad (5.1)$$

となる.

そこで

$$s_1 = s_2 \text{ あるいは } s_2 = s_3 \quad (5.2)$$

である場合には, 2 つの座標点 B, C を直線型線素でつなぎ, そのほかの場合には円弧型線素でつなぐことにすれば, 先に述べた判定方法と等価になる.

円弧型線素でつなぐ場合の曲率中心 O: (XO, YO) と曲率半径 R とは, それぞれ

$$\begin{cases} XO = (XQA + XQB) / 2 \\ YO = (YQA + YQB) / 2 \\ R = \sqrt{(XB - XO)^2 + (YB - YO)^2} \end{cases} \quad (5.3)$$

ただし

$$\begin{cases} XQA = \frac{YBC(XYA - XYB) - YAB(XYB - XYC)}{2(XAB \cdot YBC - XBC \cdot YAB)} \\ YQA = \frac{(-)XBC(XYA - XYB) - XAB(XYB - XYC)}{2(XAB \cdot YBC - XBC \cdot YAB)} \\ XQB = \frac{YCD(XYB - YXC) - YBC(XYC - XYD)}{2(XBC \cdot YCD - XCD \cdot YBC)} \end{cases}$$

$$\begin{cases} YQB \\ = (-) \frac{XCD(XYB - XYC) - XBC(XYC - XYD)}{2(XBC \cdot YCD - XCD \cdot YBC)} \end{cases} \quad (5.4)$$

ここで

$$\begin{cases} XAB = XA - XB, & XBC = XB - XC, \\ XCD = XC - XD \\ YAB = YA - YB, & YBC = YB - YC, \\ YCD = YC - YD \\ XYA = XA^2 + YA^2, & XYB = XB^2 + YB^2, \\ XYC = XC^2 + YC^2 \end{cases}$$

で与えられる。したがって、式 (3.3) において  $X_1 = XB, Y_1 = YB, X_2 = XC, Y_2 = YC$  (すなわち,  $P_1 \rightarrow B, P_2 \rightarrow C$ ) として、さらに上で求めた  $(XO, YO), R$  をそれぞれ曲率中心, 曲率半径としてとれば、円弧型線素の黒部分が式 (3.4) で表現されることとなる。

文字線の最終の座標点のつなぎ方は、後続の点が存在しないので、上のような方法は適用できない。この場合には Fig. 5.1 の座標点  $D$  を最終点とするとき

$$s_2 \approx s_3 \quad (5.6)$$

であるときに直線型線素でつなぐことにし、これ以外の場合には円弧型線素でつなぐことにする。

最終点を円弧型線素でつなぐ場合の曲率中心を、2つの垂直二等分線  $d_2$  と  $d_3$  との交点  $QB$  にとることにすれば、曲率中心  $O$  の座標  $(XO, YO)$  と曲率半径  $R$  とはそれぞれ

$$\begin{cases} XO = XQB \\ YO = YQB \\ R = \sqrt{(XC - XO)^2 + (YC - YO)^2} \end{cases} \quad (5.7)$$

で与えられ、式 (3.3) において  $X_1 = XC, Y_1 = YC, X_2 = XD, Y_2 = YD$  (すなわち  $P_1 \rightarrow C, P_2 \rightarrow D$ ) として、曲率中心  $O: (XO, YO)$  と曲率半径  $R$  とを式 (5.7) のようにとれば、この場合の円弧型線素の黒部分は式 (3.4) で表現される。

### 5.3 文字線の発生

OCR-B を発生する手順も、かたかなを発生する場合の手順とはほぼ同じである。サンプリング点を逐次移動していった、このサンプリング点が線素の黒部分の内部の点であるか否かによって、サンプル点の値を二値に量子化していく。かたかなの発生の場合と違うところは、つきつぎに与えられる座標点を、前後の座標点の位置関係を自動的に考慮しながら、直線あるいは円弧線素でつないでいく機能が備わっていることである。

第一線素は 3. で与えた線素を表現する式そのもの

で表現できる。すなわち、与えられる座標点が1つの場合には、点型線素の式 (3.7), (3.8) で、2つの場合には直線型線素の式 (3.1), (3.2) で、3つの場合には曲率中心と曲率半径とを式 (3.5) にとって円弧型線素の式 (3.3), (3.4) で記述される。ここで、座標点が3つ与えられた場合には、一般には円弧型線素であるが、曲率半径がある程度以上大きい場合には、中間の点を無視して直線型線素とみなすことにしている。

第二線素以後は、直線でつなぐか円弧でつなぐかの判定を上 5.2 で述べたような方法で行なっているから、直線あるいは円弧型線素で記述する。

各座標点では完全になめらかにはつなげておらず、これらの点の近くの文字線部分に欠けが生ずる可能性があるため、各座標点では、これらの点を中心とする直径が文字線幅と等しい円で丸めることとする。

Fig. 5.2 に OCR-B を発生するプログラムの概略の流れ図を示す。Fig. 5.3 に計算機によって発生した標準線幅の文字図形「3」を示す。かたかなの例の場合と同様の事情によってサンプリング間隔の縦横比

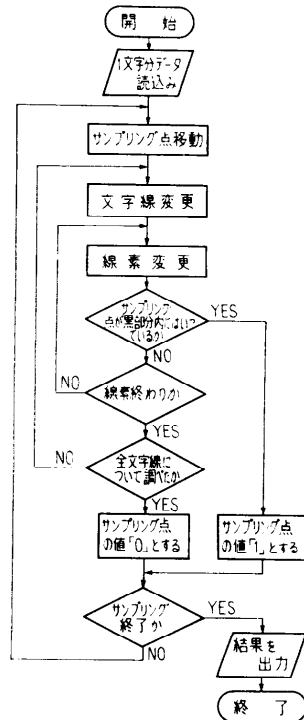


Fig. 5.2 OCR-B 発生の流れ図  
A flow chart of OCR-B character generation.

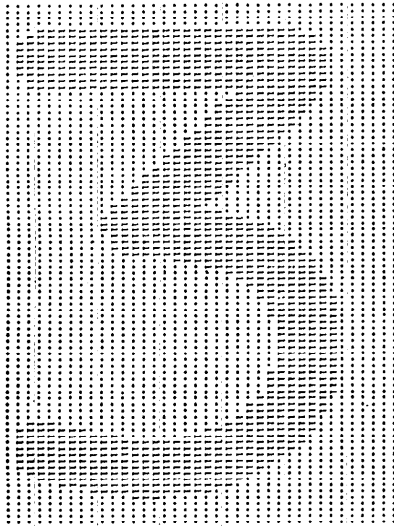


Fig. 5.3 OCR-B の発生例  
An example of OCR-B character generated.

は 5:3 に選んであり、水平方向 0.055 mm, 垂直方向 0.033 mm である。

## 6. むすび

文字図形を構成するストロークの中心線が全部直線と円弧とでできていて、直線の 2 つの端点の座標と、円弧の曲率半径と中心点、端点の座標とが与えられている場合の標準文字図形の発生方法と、定規とコンパスとで文字図形を描くことを意識しないで、文字線の中心線の座標が適当な間隔で与えられる場合に、これらの座標点を直線あるいは円弧でほぼなめらかにつないで標準文字図形を発生する方法とについて述べた。

文字図形 1 個の発生に要する時間は文字によって異なる。かたかなの発生の場合には、文字図形を構成する線素数に比例し、OCR-B の発生の場合には与えられる座標点数にほぼ比例する（これらの所要時間は IBM/360-75 を使用した場合であって、FACOM 230-50 を使用した場合には、 $[\text{所要時間}] = a \times [\text{直線型線素数}] + b \times [\text{円弧型線素数}]$ ; ( $a, b$  は係数) で与えられる。このような相違はコンパイラの違いによるものと思われる。サンプル点数が 38 (横)  $\times$  84 (縦) の場合には、IBM/360-75 を使用すると、OCR-B 文字を発生するのに数秒から 20 秒を要し、FACOM 230-50 を使用すると数十秒から数百秒を要する。

ここで述べた原理的な方法では、サンプル点が、す

べての線素を調べてみてどの線素の黒部分にも含まれていないことを確かめたくて、その点の値が「0」(白)であるという決定をくださるので、「1」(黒)であるという決定をくださる場合よりも「0」であるという決定をくださる場合のほうが時間を要することとなる。そこで、時間短縮のために、文字図形の上下左右の余白の部分は自動的に「0」とするようにしてある。

また、発生時間の短縮の手段としては履歴現象を入れること、すなわち、あるサンプリング点のつぎの点の値は、前の点の値と同じである可能性が大きいということも考慮に入れたり、あるサンプリング点が「黒」であると決定するときを用いた線素の表現式を、つぎのサンプリング点でも引き続き用いることなどが考えられるが、ここでは取り上げなかった。プログラムが複雑になる割には、所期の結果を得がたいのではないかと思われること、数種の線幅の標準文字図形を同時に発生することを考えているが、この場合、文字線幅の大きいほうから発生していけば、時間の短縮が可能であることなどの理由からである。

これまでに述べてきた文字図形の発生方法は、点サンプリングであって面サンプリングではないが、細かくサンプリングしておいて、これをあらくサンプリングし直すことによって、面サンプリングとほぼ等価なサンプリングができる。

実際のプログラムはサンプリング間隔、サンプリング領域を任意に与えて、数種類の任意の線幅の標準文字図形を順次発生できるようになっている。プログラムの詳細な流れ図と FORTRAN によるプログラムのリスト、およびデータのリストの一例、さらには OCR-B の数表の値の不自然な箇所などについては、別の機会に発表する予定である。なお、本報告の一部はユニコンを使用して行なわれた。

## 参考文献

- 1) 広川・葛城: 計算機による文字資料の作成, 昭和 43 年度電通全大 125 (昭和 43 年 10 月)
- 2) 日本電子工業振興協会: ラインプリンタによるかたかな字体規格 (原案), JEIDA-5 (昭和 44 年 2 月)
- 3) ISO: ISO RECOMMENDATION R1073, ALPHANUMERIC CHARACTER SETS FOR OPTICAL CHARACTER RECOGNITION あるいは日本工業標準調査会: 光学式文字認識のための字形 (英数字), JIS C 6250-1970 (昭和 45 年 6 月)

(昭和 45 年 10 月 15 日 受付)