

部分パターンによる漢字の合成*

坂井利之** 長尾 真** 寺井秀一**

Abstract

A method of synthesis of Chinese characters is represented in this paper.

Chinese characters have too complicated structures to be described by a simple algorithm. We have extracted about 250 basic components of Chinese characters.

From these components and with 10 synthesis rules we have succeeded in writing 2000 Chinese characters on the XY-plotter controlled by a digital computer.

In this system we have some subprograms such as enlargement, vertical/horizontal writing control.

1. ま え が き

人間の独占物であった文字が、情報処理技術の進歩によって、計算機にも開放されるようになってきている。電子計算機の大きな特色である大量・正確・高速なデータ処理が、漢字の場合に対してはどのようになるだろうか。まず、文字数ではアルファベット 26 文字に対して現用の漢字は 2,500 種程度あり、数においては著しく不利である。アルファベットを符号化する場合、せいぜい 5 ビットあればよいが、漢字では 12 ビットは必要である。さらに、1つの文字を表わすストローク構成が、アルファベットに比べて複雑である。筆者らが当用漢字 (1,850 文字) について計算したところでは、14.8 ストローク/1 文字という結果が出ている***。

文字の自動読取についても漢字はいろいろと問題がある。英数字の自動読取は、現在実用化の域に達しているが、漢字の自動認識は、数の多さと複雑さのために進まず、いままでに R. Casey, G. Nagy の研究¹⁾がある程度で、実用化はまだまだである。漢字の伝送については、現在主として漢字テレタイプが用いられているが、漢字を頭ごなしにある符号に対応づけてゆくやり方であり、実用的な面からも、また、漢字の本質をあらわすという意味からも不満足である。電子計

算機に漢字を記憶させる場合、あるいは、それを出力して漢字を描く場合、どのような方法をとれば最も能率よく、また漢字固有の特徴をそこなわずに行なえるかということを考えるのはきわめて大切である。これはまた漢字の入力の問題にも通じることである。この問題を論じるには、まず、漢字を生成する機械としての漢字の文法体系を考えるべきである。そして、その体系に則した符号列を発見し、これによって漢字をユニークに指定することができ、しかも、われわれが日常もっている漢字に対する常識に背反しないものであれば、それは漢字の符号化に一応成功したものと考えてよいであろう。

この論文は、以上の考えに立って、漢字の幾何学的構造に注目し、漢字が明確な構造をもっているとの観点²⁾から、それを構成している各要素 (偏、旁など) の位置関係にある約束のもとに記述して、計算機で漢字を合成する方式について述べている^{3), 4), 5)}。位置関係を示すオペレータとして 10 種類、要素としての部分パターンを多くて 250 種類とれば、すべての当用漢字はこの記述方法で記述されることが判明した。さらに、この考え方は、原理的には他のあらゆる漢字、および漢字以外の文字 (ハングル文字など) にも適用できる。われわれの研究室では、合成された漢字は XY レコーダに on line で出力されるようになっている。

2. 漢字の記述方式

ここで漢字をその構造面から 1つの絵文字とみなしその構成要素の幾何学的な配置関係に注目してみる。このような立場に立つと漢字をいくつかの要素に分解

* A Description of Chinese Characters Using Sub-patterns by Toshiyuki Sakai, Makoto Nagao and Hidekazu Terai (Department of Electrical Engineering, Kyoto University)

** 京都大学工学部

*** ここでいう 1 ストロークとは、折線でない 1本の線分のことである。たとえば、「7」の従来の画数は 1画であるが、ここでは 2 ストロークになる。

表1 オペレータとその例

Operators for the composition of Chinese Characters

operator	意味	例	operator	意味	例
H	左右(離)	休	S	包圍	困
V	上下(離)	電	K	冠	字
P	左右(接)	非	L	ニョウ	進
Q	上下(接)	否	J	タスキ	司
I	貫通	中, 丹, 孝	G	タレ	歴

し、これらの要素どうしの結合法則をいくつか規定して漢字を記述することが可能である。漢字は本質的には線図形であり、いくつかの線分の組合せで描くことが可能であるが、それでは結合に関する記述が煩雑になる。漢字を1本の線分までに分解してしまわずに、ある程度まとまった形を残したままの要素に分解する方法をとれば、要素の結合関係はかなり単純明確化される。そこで表1に示すような種類のオペレータをとった。

“H”は「休」のようにいくつかの要素(例では“イ”と“木”)が左右に並んでいる関係を示すオペレータである。“V”は「電」のようにいくつかの要素(“雨”と“電”)が上下に並んでいることを示す。“P”、“Q”はそれぞれ“H”、“V”と同様のオペレータであるが、“H”、“V”は要素が離れて並んでいる関係を表わしているのに対し、“P”、“Q”は接して並んでいる関係を表わしている。“I”はある要素が他の要素を貫通している関係を表わす。この場合「中」のように“1”が“口”を縦方向に貫いているものや、「丹」のように“1”が“口”を横方向に貫いているもの、さらに、「考」のように“/”が斜方向につきぬけているものなどいろいろあるが、貫く要素が変わるだけであり、オペレータ“I”で一括して表現する。“S”は「困」のようにある要素(“口”)が他の要素(“木”)を包圍している関係を表わす。しかし、要素どうしの位置関係が、上に述べた6つのオペレータで一意的に記述できない漢字がある。たとえば、「進」は“1”と“木”が左右に並んでいるとも考えられるし、上下に並んでいるとも考えられる。このような表現の二重性を防ぐために、さらに4種類のオペレータを規定する。それが表1の“K”、“L”、“J”、“G”である。おのおのが要素どうしのどのような位置関係を表わしているかは、表1にあげた例をみればわかるであろう。このようなオペレータによって結合される構成要素(以後、部分パターンと呼ぶ)

として、約250個を当用漢字から経験的に抽出した。その際、次の点に留意した。

(i) 従来の214個の部首をある程度参考にはするが、厳密にはそれにとらわれない。——たとえば、「高」は1つの部首としてあるが、これは“1”、“口”、“冂”を用いてV(1, 口, S(冂, 口))と記述できるので部分パターンとして採用しない。反面「冂」は従来の部首にはないが、よく用いるので部分パターンとして採用する。

(ii) できるだけ多くの漢字に対して融通性があること。——「糸」ヘンをもつ漢字は約60個と多い。それで「糸」を1つの部分パターンにとってもよいがそれよりも“彡”+“小”に分解しておいた方が得策であると考えられる。なぜなら、「幼、幻、幾、機、番、蓄、磁、滋」のように他に“彡”を使う字が多くあり、また、“小”の方は「京、原」などに用いられて使い道が多い。

(iii) 部分パターン自体のストローク数が多くなりすぎないこと。しかし、使用頻度の高いものはより小さな単位に分解することはせずに、少くらしいストローク数がふえてもかまわないから、そのままにしておく。——これは(ii)とのかね合いでむずかしい。たとえば、「言」を偏にもつ字は当用漢字だけで60個ある。これをV(1, 一, 一, 口)と分解してもよいわけであるが、それでは細かくなりすぎるきらいがあるし、1つの完成した漢字としての記述が長くなる。

(iv) 補正あるいは修正のための特別な部分パターンを用意して、でき上りの漢字の品質を上げようようにする。——これは一種のウメ木、あいは無駄要素である。ある部分パターンの大きさを故意に変えたときや位置をずらせたいときに用いる。たとえば、部分パターン“木”は無駄要素“b”といろいろ組み合わせられて図1に示すように変形される。

250個の部分パターンは001から250まで通し番号をつけて磁気テープに記憶させてある。その際、すべての部分パターンが10mm×10mmの枠内に納まるように大きさを正規化しておく。また、おのおのの部分パターンの属性を考慮して、3種類のクラスに分類

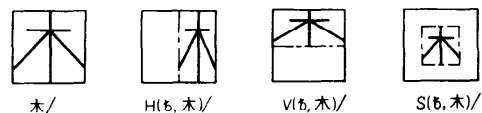


図1 むだ要素(b)と“木”の結合
Combination of a dummy element (b) and “木”

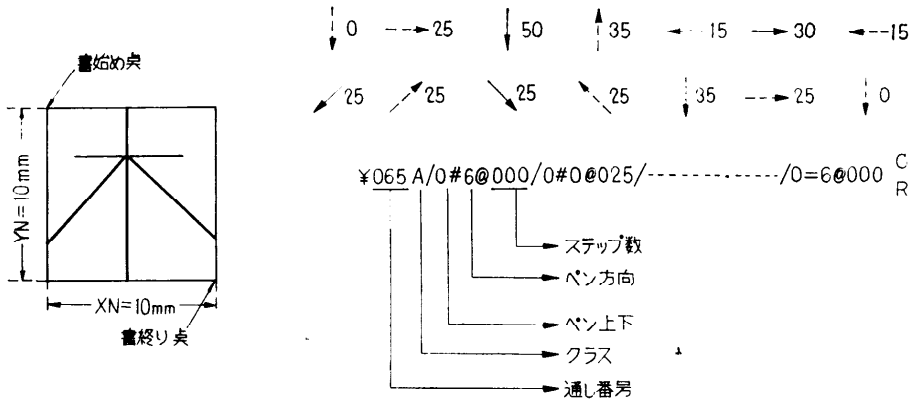


図 2 部分パターンの記述様式
Detailed description of a subpattern

しておく。ここで属性とは部分パターンが縦方向に長い、横方向に長い、あるいは縦横両方向に広がったものであるか、という固有の形状に関する性質をいう。縦横両方向に広がったもののクラスをA、横長のクラスをB、縦長のクラスをCとする。さらに、後の処理のため、すべての部分パターンは枠の左上隅から書き始め、常に右下隅で終わるように統一しておく。部分パターンを表2に示す。部分パターンの記述方法を「木」を例にとって図2に示す。部分パターンはストローク単位で記述されている。図2の矢印はXYレコーダで出力するときの方向を示し、点線はペンがup、実線はペンがdownしていることを表わしている。矢印の後に書かれた数値はそのストロークのステップ数である。出力装置として1ステップ0.2mmで8方向にペンが動くXYレコーダ⁶⁾を用いるので、これにあった記憶方式をとってはいるが、他の出力装置(CRT)にも適用できるように考慮されている。

以上10種類のオペレータと約250個の部分パターンを用いれば、当用漢字をすべて記述することができる。例として「算」についての構造を述べてみる。この字を分解する方法はいろいろあるが、現在の部分パターンの中に含まれていて、しかも、記述が簡潔になるように考えた場合、要素は「ケ」、「目」、「J」、「I」、「一」の5つになる。これらの結合関係をかっこ式で書けば

／算／…V(H(ケ, ケ), 目, I(H(J, I), 一))
となる。いくつかの漢字のかっこ式を図3に示す。このかっこ式に関してレベルという概念を導入する。レベルとはかっこ式中のかっこの深さに対応していて、部分パターンおよびオペレータにつけられた数値であ

- 密 /-----V(K(一, I(心, /)), 山) /
- 好 /-----H(女, 子) /
- 審 /-----V(Q(一, K(一, 一)), 才) /
- 痴 /-----G(一, H(矢, 口)) /
- 致 /-----H(Q(一, 一, 一), 女) /

図 3 いくつかのかっこ式の例
Few examples of the coding of Chinese character

る。「算」の例では「ケ」はレベル2、「目」はレベル1、「I」はレベル1、「J」はレベル3である。

3. 漢字合成の方式

部分パターンどうしがオペレータによって結ばれ、1つの漢字ができていくわけであるが、漢字を合成する場合は、まず入力されたかっこ式を解釈して、1つの漢字を構成しているおのおの部分パターンが実際に漢字内でどのような位置を占有するかを決めてゆく。つまり、でき上がりの漢字がはいる枠を想定し、枠によって囲まれた領域全体をかっこ式に応じて分割し、所用の部分パターンをはめ込んでゆく。領域を分割してゆく方法はそれぞれのオペレータによって違っている。図4に10種類のオペレータに対する分割の要領を示す。分割によってできた小領域(以後、これを部分領域と呼ぶ)の位置は、左上隅と右下隅のX、Y座標で表わされる。ただし、座標の原点はもとの全領域の左上隅にとり、右方向にX軸、下方向にY軸とする。

オペレータ「H」と「P」、「V」と「Q」は互いに似た

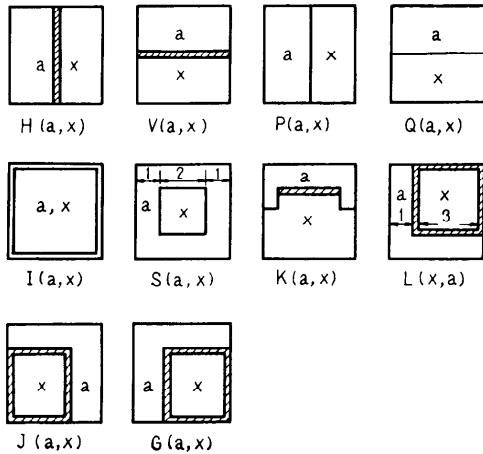


図 4 領域分割のオペレータ

Operators of the domain partition

性質のものであるが、「H」、「V」に関する分割に際しては、要素間の「すきま」を含めて部分領域に分割するのに対し、「P」「Q」においては「すきま」部分をもうけずに分割する。図4の斜線部分がその「すきま」である。

オペレータ「I」は、図形パターンとしては貫通関係をあらわすものであるが、領域分割では、むしろ「重ね合わせ」の機能をもつ。つまり、ある要素が他の要素を貫通しているということは、いまの場合要素どうしを重ね合わせることによって得られる。その際中心線がずれないかという心配があるが、部分パターンを作るときに、あらかじめ中心線を統一しておけば問題はない。結局「I」に関する分割については、貫通する要素と貫通される要素に、同じ X, Y 座標を与えればよいことになる。

包囲を表わすオペレータ「S」は、たとえば、S(口, 木) というように包囲する方を先に、包囲される方を後に書く。包囲される方の部分領域の大きさは、現在図4に示した比率をとっている。また、このオペレータと無駄要素を用いて、他の要素を故意に縮小することができる。たとえば、S(b, x) とすれば要素 x は 1/4 に縮小される。

オペレータ「K」、「L」、「J」、「G」はいずれも「H」と「V」の両属性をそなえている。分割の要領は図より明らかであろう。これらのオペレータに関係した部分パターンは「冂, 凵, 宀, 冫, 廴, 辶, 廌, 廴, 冫, 廴, 冫, 廴」ぐらいである。

領域が各オペレータによって分割されるとき問題になるのは、部分領域どうしの大きさの比率ができ上がりの文字に与える品質の影響である。「恒」という字に

ついて考えてみよう。これがかっこ式で書けば $H(\uparrow, V(-, \text{日}, -))$ である。H で結ばれている「冫」, 「亘」の横幅は通常「冫」より「亘」の方が広い。また、Vで結ばれている「一」, 「日」は、横方向の長さはいわゆるかわらないが、縦方向の長さ(厚みのこと)は明らかに「一」より「日」の方が長い。部分パターンあるいはブロック(いくつかの部分パターンとオペレータがまとまって1つの閉じた部分かっこ式をついているもの)によって、幅や厚みがいろいろ違っている。この事実を無視して均等に部分領域に分割してゆけば、自然性の欠けた文字になってしまう。また、「健」の字で $H(\text{イ}, L(\text{隹}, \text{辶}))$ を均等に分割しゆけば、やはりレベルの深い「隹」ほど小さくなってしまい、字の粗密、大小関係が不均衡になってくる。これを防ぐために、部分パターンおよびブロックに一定の重みをつけ、分割率をかえてゆくことを考える。各文字の形状に応じて、部分パターンあるいはブロックに最適の重みを与えることは困難であるが、いま一応の基準として表3に与えた値を採用する。特に「Q」に関して「一」の重みを0、「P」に関して「冫」の重みを0としたのは、接している限りでは、「一」, 「冫」に厚

表 3 分割の重み
Partition weighting

operator	要 素	重 み
H	L, G, J, P	4
	クラスA, クラスB	2
	クラスC	1
V	K, Q	4
	クラスA, クラスC	2
	クラスB	1
P	L, G, J, H	4
	クラスA, クラスB	2
	クラスC(部分パターン「冫」)	1(0)
Q	K, V	4
	クラスA, クラスC	2
	クラスB(部分パターン「一」)	1(0)

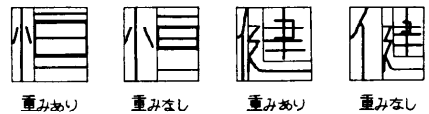


図 5 領域分割に重みをつけたときとつけなかったときの比較例

Comparison of the domain partition with the partition weighting and without it

み、幅がないものと考えたからである。このようにしておけば、トリミングによる悪影響を受けずに都合がよい。図5に重みをつけない場合の分割結果と重みをつけて分割した結果の文字例を示す。ここでいうトリミングとは、部分パターンを記述しているストロークの特定の一部分を除去することをさす。この処理は、トリミングを受けるべきストロークのステップ数を強制的に0にすることによってなされる。トリミングはオペレータ“P”、“Q”の場合に行なわれる。“P”、“Q”に関する分割に際しては、部分領域間に“すきま”を含めずに分割することはすでに述べた。一方、部分パターンは10mm 平方の枠内いっばいに描かれてあるから、このような分割のしかたで作られた部分領域に部分パターンをはめ込むだけで、これらは互いに接し所期の目的が達せられるわけであるが、中には枠内いっばいに書けない部分パターンがある。「畜」-V(Q 十、玄、田)、の「十」がその1つの例である。このような部分パターンについては、図6(a)に示すようにただはめ込んだだけでは「十」と「玄」は接しない。部分パターンを作るとき、あらかじめ書き始めと書き終わりの順序を統一してあるので、いまの例の場合は「十」の書き終わりと「玄」の書き始めのペン up のストロークがトリミングされる。こうして得られた結果を図6(b)に示す。枠いっばいに書かれた部分パターンは、書き始めと書き終わりのペン up のストロークのステップ数は、いずれも最初から0であるから、トリミングを受けても大きさは変わらないが、枠いっばいに書かれていないものについては、ペン up のストローク分の“すきま”がつまるわけであり、それだけ全体の文字は圧縮される。

次に、辞書に登録されている標準部分パターンが、得られた部分領域にうまくはまり込むようにその形を変形する。領域分割の結果、各部分パターン N_i が占める部分領域の大きさがわかる。この部分領域 D_i の横方向の長さを X_{D_i} 縦方向の長さを Y_{D_i} とする。一方、辞書に登録されている標準部分パターン N_i の

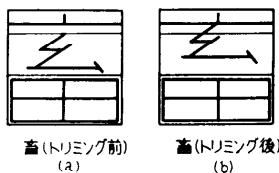


図6 部分パターンの接続

A character with trimming operation (b) and without it (a)

横枠、縦枠の長さを X_{N_i} , Y_{N_i} (現在はそれぞれ 10 mm) とすると

$$F_{X_i} = Y_{D_i} / X_{N_i}$$

$$F_{Y_i} = X_{D_i} / Y_{N_i}$$

$$F_z = \text{Max}(F_{X_i}, F_{Y_i})$$

この F_{X_i} , F_{Y_i} , F_z は標準部分パターン N_i の横方向、縦方向、斜方向の圧縮率(伸張率)になる。標準部分パターンは、すべて図2に示した形式でストロークごとに記述されているから、横方向のストロークのステップ数には F_{X_i} 、縦方向には F_{Y_i} を乗じれば、縦横に関する部分パターンの変形はできたことになる。ここで問題になるのは、斜方向のストロークである。最初部分パターンを作るとき、斜ストロークはこう配 ± 1 で記述してあるが、得られた部分領域の縦横の比率が1に等しくない場合、斜ストロークの勾配もこの値に応じて変化させる必要がある。この処理を行わないと、部分パターンに斜ストロークを含んでいる字では、全体がずれて不自然になる。また、1つの漢字は、普通いくつかの部分パターンが組み合わさってできている。このため、ある部分パターンの書き終わりから、他の部分パターンの書き始めへの移行データを作る必要がある。遷移データは部分領域どうしの距離から求められる。

4. 漢字合成のシステムとアルゴリズム

現在、行なっている漢字合成のシステムを図7に示す。使用計算機は NEAC 2200 モデル 200 で、コア容量は 16 K characters である。この計算機は character machine であり、1 character は 6 bits から成っている。標準部分パターンは磁気テープ (No. 1) に格納してある。プログラム実行の最初に磁気テープ上の部分パターンを磁気ドラムにうつしかえるが、これはコアの容量が小さいために全部コアへ入れることがで

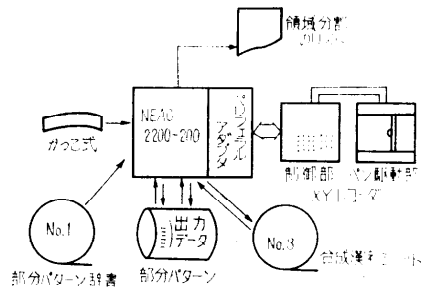


図7 漢字合成システム

Block diagram of the machine for the composition of Chinese character

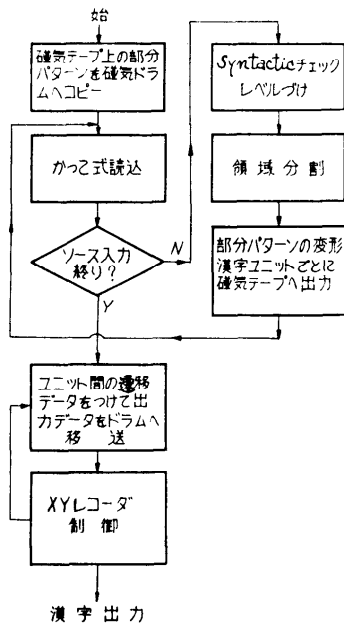


図 8 漢字合成プログラムのフローチャート
Flow chart of the programming system

まず、磁気テープではアクセス時間が遅いためにとった処置であり、本質的な問題ではない。合成された漢字データは1文字ごとにラベルをつけて磁気テープ (No. 3) に記憶されるとともに、XYレコーダ上に描かれる。磁気ドラムがその際の出カバッファとなる。XYレコーダ⁶⁾には16個の機能ボタンがあり種々の制御指令を計算機に送ることができる。現在はこれらのボタンに、出力文字の拡大、縦書き/横書き選択、改行復帰動作などの機能を割り当てている。

合成アルゴリズムのフローチャートを図8に示す。使用プログラム言語はEASYCODERと称するassembly言語であり、プログラムのためのメモリ量は約38K characters、全実行命令数は約2,000ステップである。処理速度は平均1.5秒/1文字であるが、そのうち90%はXYレコーダへの出力データ列を生成するのに費されている。

まず、紙テープにパンチされたかっこ式が、1つの漢字のかっこ式の終わりを示す“/”記号まで読まれると、プログラムはsyntacticチェックのルーチンへジャンプする。ここでは入力かっこ式が正しく記述されているかどうかをチェックする。同時に各オペレータおよび部分パターンにレベル付けを行なう。

syntacticチェックを通りぬけたかっこ式は領域分割ルーチンで解釈され、各部分パターンの位置がきめ

られる。位置のきめ方は3.で述べたとおりである。領域分割はレベルの浅いものから深い方へと逐次行なう。分割の手順を「疎」という文字を例にとって述べる。「疎」のかっこ式は次のとおりである。

／疎／…H(Q(→,止), I(V(一,口,H(/,\)), |))

(i) レベル0に関する分割

レベル0のオペレータはHであるから、図4の要領で20mm×20mmの領域が分割される。この結果Q(→,止)とI(V(一,口,H(/,\)), |)の2つのブロックの部分領域の位置がきまる。いまの場合、両ブロックの重みはともに2であり均等分割である。

(ii) レベル1に関する分割

まず、Q(→,止)の分割が行なわれる。「→」はクラスB、「止」はクラスAの部分パターンであるから、上下に1:2の割合で部分領域に分割される。しかもオペレータがQであるから、トリミングのための情報が書き込まれる。「→」の書き終わりのペンupのストロークと、「止」の書き始めのペンupのストロークがトリミングを受ける。次いでI(V(一,口,H(/,\)), |)の分割がなされる。この段階でブロックV(一,口,H(/,\))と部分パターン「|」の位置が決定する。

(iii) レベル2に関する分割

レベル2を持つものは「→」、「止」、V(一,口,H(/,\)),「|」の4つである。このうち「→,止,|」は部分パターンであり、レベル1に関する分割の段階で位置がきまっているから、もはや、分割はなされない。

V(一,口,H(/,\))が分割される。「一」、「口」、H(/,\)の重みは1:2:2であるから、この比率で上下方向に並んだ部分領域に分割される。

(iv) レベル3に関する分割

レベル3の要素のうち、部分パターンでないものはH(/,\)だけであり、(i)と同様の要領で「/」と「\」が均等分割される。「疎」のかっこ式の最深レベルは4であり、もはや、分割されるべきブロックはない。

以上ですべての分割が終了する。この分割の順番を図9(a)に、その出力文字を図9(b)に示す。このようにして求めた部分領域の大きさに合うよう各部分パターンを変形する。その方法はすでに説明した。この結果得られたストローク列からXYレコーダへの出力データを生成する。部分領域が正方形でない場合、斜ストロークの出力データは、部分領域の縦横の比率

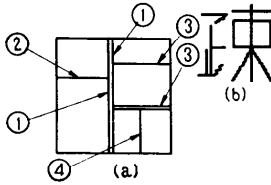


図9 疎の部分領域分割
Domain partition of "疎"

に等しい勾配をもった近似線分で置き換えられる。

5. 検討

漢字ディスプレイ装置に応じて、種々の漢字記憶方式^{7),8),9)}が考えられる。従来発表された漢字記憶方式を分類すると、次のようになる。

(a) ドットパターンに分解する方法⁷⁾。

(b) ストローク列に分解する方法。

(b1) 各ストロークの始点と終点の XY 座標を記憶する⁸⁾。

(b2) 各ストロークの始点の XY 座標とストロークの長さ、方向を記憶する⁹⁾。

(c) 部分パターンに分解する方法^{3),4),5)}。

2,000 文字を対象とした場合、各方式での所要メモリの概略を比較すると表4のようなになる。(a)および(b)の方式では分解能がそのまま内部メモリ量に関係してくる。たとえば、ドットマトリックスで分解能を縦横 1/20 にしようとするれば $20 \times 20 = 400$ ビット/文字必要となってくる。(b)の方法で分解能を 1/16 から 1/32 に上げれば、XY 座標それぞれ 5 ビット必要になり、1 ストロークにつき (b1) では 20 ビット、(b2) では 18 ビットとなる。また、(b2)の方法で方向を 8 方向から 16 方向に細かくすれば、各ストロークごとに 1 ビットふえる。

一方、(c)の方法では、計算によって任意の大きさの漢字を得ることができ、表示される文字の大小、荒

表4 メモリ量比較

Comparison of the necessary memory capacity for various methods

記憶方式	分類項目	メモリ容量
ドットパターン	15×18ドット	540Kビット; 18Kワード
	18×18ドット	648Kビット; 21.6Kワード
ストローク列 (16×16メッシュ)	ストロークの始点と終点の座標	416Kビット; 14Kワード
	ストロークの始点の座標と、長さ、方向	416Kビット; 14Kワード
部分パターンと かっこ式	部分パターン	46Kビット; 125Kワード
	かっこ式	122Kビット; 5.6Kワード

* 1ワード=30ビット

さ、細かさは、かっこ式のメモリ量に何ら影響を及ぼさない。この点が他の方式と大きく異なっている。部分パターンのストロークは 8 方向 (3 ビット) で表現されているが、この方向も部分領域内では、各領域に応じた任意勾配の線分に置き換えられるから、3 ビットで十分である。この方式のおもな利点・欠点を列挙すると、次のようになる。

(i) 活字あるいはドットパターンによる表示方式に比べて、でき上りの文字の大きさ、縦横のプロポーションが任意に選べる。これは計算機を用いる場合の大きな特色であるプログラムによる柔軟性がよく生かされているためである。

(ii) 同じ部分パターンでもその形状を変化させることによって完成された字の形状も少しずつ変わり、文字そのものに個性をもたせることができる。

(iii) 入力は一視的に明確な部分パターンを用いるので、よみ方を知らない者でも容易に行なえる。

(iv) 250 程度の部分パターンさえあれば当用漢字はもとより、原理的にはあらゆる漢字を合成することができ、すべての漢字パターンを記憶しておく必要はない。

(v) 「為」のような字に対しては、現在のオペレータは無力である。このような文字は最初から部分パターンとして登録する方針をとっているが、その例は当用漢字の中では 10 個程度である。

(vi) 「太」や「犬」の「>」の位置を直接指定するオペレータがない。HとVおよび無駄要素を用いて「>」の位置をかえることが可能であるがかっこ式が長くなる。

(vii) かっこ式の ambiguity がある。たとえば、「競」は H(V(立, Q(ロ, ル)), V(立, Q(ロ, ル))), とも書けるし、V(H(立, 立), H(Q(ロ, ル), Q(ロ, ル))) とも書ける。しかし、これは HはVに優先するというように優先順位を決めておくことによって防ぎうる。

(viii) 機械的な領域分割によるので、でき上りの文字が少し不自然になる場合もある。

これまで述べたところでは、すべて漢字を対象にしてきた。漢字以外の文字へこの合成アルゴリズムが適用できるだろうか。それが可能であるためには、適用される文字が一定の規則的な構造をもっていることが必要である。漢字以外のこのような文字としてはハングル文字 (諺文) がある。ハングル文字は朝鮮の国字で、10 個の母音字と 14 個の子音字からなる表音文字である。

これらの母音字、子音字が組み合わさって、1つの音節をあらわす文字となるわけであるが、その結合関係は視覚的にかなり単純で、ほとんどオペレータ H, V, P, Q で位置関係を記述することができる。しかも、部分パターンは母音字、子音字あわせて 24 個あればよく、漢字よりも簡単にいままでの方法を適用することができる。このようにハングル文字はストロークが単純で、それぞれ字母の特徴がはっきりしており位置関係が明確で字母数が少ないなど、情報化時代の新しい文字として、再認識されてもよい興味ある文字である。

参考文献

- 1) R. Casey., G. Nagy: Recognition of Printed Chinese Characters, IEEE Trans. EC, Feb. 1966.
- 2) B. K. Rankin, W. A. Sillars and R. W. Hsu: On the Pictorial Structures of Chinese Char-

- acters. NBS Technical Note, Jan. 1965.
- 3) 坂井利之, 長尾真, 寺井秀一: 漢字の合成, 情報処理学会第9回大会, 昭 43-12.
- 4) 坂井利之, 長尾真, 寺井秀一: 部分パターンによる漢字の記述, 信学会オートマトン・インホメーション理論研究会資料, 昭 44-1.
- 5) 寺井秀一: 電子計算機による漢字の合成, 京都大学修士論文, 昭 44-3.
- 6) 坂井利之, 堂下修司, 長尾真, 西尾英之助: 周辺制御装置と入出力用 XY レコーダ, 信学会電子計算機研究会資料, 昭 43-7.
- 7) 岩井麟三: 電子計算機出力としての漢字プリンタ, 情報処理学会第9回大会, 昭 43-12.
- 8) H. Hayashi, S. Duncan and S. Kuno: Graphical Input/Output of Nonstandard Characters, CACM, Sept.-1968.
- 9) 中摩雅年, 吉本悠久, 後藤明也, 竹内慶光, 日高滋: 漢字の陰極線管表示方式, 情報処理, 昭 43-11 (昭和 44 年 3 月 20 日受付)

訂 正

Vol. 10 No. 3 掲載の穂坂 衛氏の論文「曲線、曲面および平滑化理論」は下記のとおり誤りがありますので訂正いたします。

	誤	正
p. 121 註	Filting	Fitting
p. 123 (18) 式	k_{i+1}	k_i
p. 125 (29) 式	$S_0(n, v)$	$S_0(u, v)$
p. 126 (37), (39), (40) 式	$\sum_{i=0}^m$	$\sum_{j=0}^m$
p. 128 (47) 式	$\frac{\partial u}{\partial y_i}, \frac{\partial u}{\partial y_{i'}}$	$\frac{\partial V}{\partial y_i}, \frac{\partial V}{\partial y_{i'}}$
p. 128 左下より 1 行目	f_1	f_i
p. 128 (51) 式	y^{i+1}	y_{i+1}