解　説

日本語文字発生方式†

野村仙一†† 小池博之††

1. はじめに

わが国における情報処理の特徴として日本語情報の処理がある．従来，コンピュータの入出力情報といえば演算処理を目的とした数値情報が中心であった．その後，情報処理の適用分野の拡大に伴って，事務処理業務が激増し，数値情報と共に文字や文章あるいは図表といった非数値情報の取扱いが要求されるようになった．特にわが国の事務処理で扱う非数値情報では，漢字や仮名文字（ひらがな，カタカナ）からなる日本語情報が大きな役割りを占めている．

日本語情報の特質は，漢字の種類が非常に多いことである．最も一般的な制定漢字として当用漢字があるが，この制限枠内の漢字でさえも1,850字種ある．また新聞や雑誌に使用されている漢字の種類は，国立国語研究所の調査報告によると約3,000字種であり，さらに姓名や地名等の顧客情報サービスに必要とされる文字種は約10,000字種といわれる．ちなみに漢和字典に収録されている新旧漢字・俗字等を含めた最大の文字種は約50,000字種にもなる．

1978年1月1日付で制定された情報交換用漢字符号系 JIS C 6226 では，第1水準漢字集合2,965字，第2水準漢字集合3,384字の計6,349字が選定されている．

こうした文字種の多さと共に，漢字の字形が非常に複雑なこともあって日本語情報のコンピュータ処理を難題にしている．欧米諸国の文字・言語情報が26種の基本文字からなるローマ字情報で十分なことと比較して，わが国の日本語情報はコンピュータにとって非常に扱いにくい言語といえる．入出力装置の構造においては，全文字種を直接入出力しようとすれば，入力操作する文字キーの数が100倍以上にも達するために装置は大型化して操作性が非常に劣化してしまう．また出力装置は文字発生器が必要となり，その機構は複雑・大型化し，装置の価格も非常に高価なものにしている．

本解説では，特に出力装置を複雑・大型化して高価格化の主因をなしている文字発生器について，その文字発生方式の分類と特徴，および小型・低価格化を図るための圧縮方法を概説する．

2. 文字発生方式の分類と特徴

これまでに研究開発されている文字発生方式を分類すると，表-1に示すようにアナログ型とディジタル型とに大別できる．アナログ型は，字母そのものを活字としてあるいはレンズやフィルム上に撮影しておき，これを打鍵あるいは光学的処理によって文字発生する方式であり，ディジタル型は，字母を線や点の集合としてディジタル記憶（数値化）しておき，これを復元処理して文字発生する方式である．その性質上，文字発生されるパターンの品質は，アナログ型では自然文字を得ることができるが，ディジタル型では疑似文字になる．しかしながら文字パターンの記憶媒体としてコンピュータ用メモリ（コアメモリ，IC メモリ，磁気メモリ等）を使用できるディジタル型は，処理の容易さと，装置の機構の簡素化，規模の小型化等に多くの特長を有している．最近のメーカの開発動向も，商用印刷を目的にした電算写植システムを除いて，多く

表-1　文字発生方式の分類

- 文字発生方式
  - アナログ型
    - 活字式
    - 光学式
      - フライングスポット式
      - ビジコン式
      - 字母光学式
      - 特殊管式
    - ホログラム式
  - ディジタル型
    - ドットマトリックス式
    - ラインドット式
    - ストローク式
    - 部分パターン合成式
    - 構成ユニット合成式

† Japanese Character Generating Method by Senichi NOMURA and Hiroyuki KOIKE (Development Department, JIPDEC).

†† (財)日本情報処理開発協会開発部

はディジタル型，特にドットマトリックス式が主流になっている．

以下に各文字発生方式の基本的な原理と特徴を述べる．

（1） 活 字 式

この方式は，活字式インパクトプリンタに使用されている方式であり，活字をタイプパレットあるいは活字ドラムに収容しておき，この中から該当文字を選択してハンマで打鍵印字する純機械式である．活字の材質として金属性のものとプラスチック性のものがある．文字サイズやピッチが固定的で，しかも文字発生，印字速度が遅く，騒音の問題もあるが，印字品質はタイプ印刷と同程度に高品質で複写がとれ，安価であるという特徴を有す．

（2） フライングスポット式

この方式は，文字を写真撮影して収容した文字マトリックス板（高解像度写真乾板）の中から，該当する文字をフライングスポット CRT のビーム偏光によって選択走査する．CRT 走査により選択された文字形の光信号は光電子増倍管によって増幅され，整形回路を経てプリント CRT に1字ずつプリントされる．プリント CRT のビームはフライングスポット CRT の文字走査偏光信号に同期しているので，プリント CRT には文字が忠実に再現できるという特徴がある．読出し部の構造は複雑であるが，多字種の文字を高品質で任意の形・大きさで文字発生できる利点がある．

（3） ビジコン式

この式は，文字円板，フラッシュランプ，ビジコン (Vidicon) の組合せで文字を発生させる．同心円上に文字を配列した円板を回転させておき，該当文字位置にきたときフラッシュランプを閃光させてビジコン上に円板の1列分を投影させて蓄積する．この文字群はビジコン光電面の蓄積効果によって記憶されているので，この中から電子ビームで該当文字1字を選択走査し，文字パターンに従ってビデオ信号を得る．文字を読出す際に円板の一部しか使用しないことから比較的解像度が良い．また1枚の円板上に数千の文字が収容できるため，文字発生機構はわりと小さくて済む．ただし文字発生速度は遅い方である．

（4） 字母光学式

この方式は，機械式の写植装置に多く用いられており，その原理は回転している文字盤をフラッシュランプで照射し，必要とする文字をレンズ系によりプリント面に撮影する方式である．文字盤の種類によって回転円板型と回転円筒型とに分類される．文字発生，出力速度は遅いが，安価で良好な印字品質を得ることができる．

（5） 特 殊 管 式

文字発生用の特殊管として，ガラスバブル管内に透過形文字パターンを内蔵して電子銃のカソードから放射した電子ビームを加速し水平・垂直の偏向電極により文字パターン上の希望する文字を走査して文字信号を得るモノスコープ管や，CRT の中に文字板を封入したキャラクトロン管，ビジコンに文字選択格子を封入したライノトロン管などを用いた方式である．

（6） ホログラム式

この方式は，文字パターンメモリとしてホログラムを使用したものであり，レーザ光によりホログラム文字板を偏光選択し，撮像装置により文字信号を得ることができる．ホログラムは，透明な特殊高分子物質の板の上に刻んだ位相ホログラムであり，高い再生効率を持っている．またホログラムは互いに共役な波面が同時に再生されるため発散性の像と収束性の像とが得られることから，特に高品質のレンズ系などを用いなくても容易に結像が行える．

（7） ドットマトリックス式

この方式は，文字を点(ドット)の集まりとして表現する方式である．英数字の場合は 5×7 ドット程度で表現され，漢字や仮名の場合は 16×16～32×32 ドット程度にして使用されている．格子(マトリックス)を密にすればいくらでも高品質文字を得ることができる．ドットマトリックスの文字パターンを記憶するのに，コアメモリあるいは IC メモリ，磁気ディスクなどの記憶素子を大量に必要とするため，文字品質とメモリ量，文字発生速度，価格等で最適化を図る必要がある．

（8） ラインドット式

この方式は，文字を縦あるいは横のラインで分割し，そのライン上での白（地）と黒（文字部分）との境界の座標点を記憶しておく方式である．文字発生時には，各ライン座標と境界点座標の集まり(1文字分)をメモリから読出して画像信号に変換する．この方式は，XY 方向のいずれかがドットではなく直線で再現できるために文字品質はドットマトリックス方式よりも優れている．しかし，メモリ量は文字品質（ライン分割数）に比例して大量に必要である．

（9） ストローク式

この方式は，文字をストローク（直線）に分解して表現する方式であり，文字の中心線を1本あるいは複

数本でストローク化する方法と，文字の輪郭をストローク化する方法がある．ストロークは，始点と終点の座標によって表現するか，あるいは始点の座標と終点までの長さと方向とによって表現する．曲線部分は折れ線で表現しなければならないが，文字サイズを容易に変更できる利点がある．メモリ量は，中心線1本でストローク化すれば文字品質は低下するが小量で済み，ほかの方法では高品質文字が得られるが大量のメモリを必要とする．またストローク信号によって動作する出力装置の構造が複雑になる．

(10)　部分パターン合成式

この方式は，文字の構成要素(部分パターン，構成パターン)を組合せて文字発生する方式である．漢字は，「へん」や「つくり」，「かんむり」，「たれ」などの部首を構成要素にしており，約250種の部分パターンと，これらの配置関係を表わす約10種のオペレータによってかなりの文字種を表現できる．同一部分パターンが多くの漢字に共通に用いられることから，パターンメモリ量の減少が図れる．しかしながら，同一パターンでも文字の中の相対的な位置関係によって形が異なる性質があるために合成文字の品質は劣る．なお各部分パターンは，ストローク情報等で記憶されている．

(11)　構成ユニット合成式

この方式は，文字の構成要素を部分パターン合成式よりさらに細分化した構成ユニットの集まりとして表現する方式である．この方式では，明朝体文字特有の「はね」や「おさえ」用などの構成ユニットを用意しておけば，かなり品質の良い合成文字ができる．ただし合成のためのアルゴリズムは複雑である．

3. 文字パターンの圧縮方法

前述したような各種の文字発生方式が研究開発されているが，ディジタル型の文字発生方式に共通した問題は，文字パターンを記憶するメモリ（文字パターンメモリ）を大量に必要とすることである．ディジタル型の文字発生方式の場合，いずれも文字パターンを線（ストローク）あるいは点（ドット）の集合として表現することから，文字パターンメモリ量はストローク数やドット数のいわゆる文字パターンの分解数と，その符号化法で決まってくる．

文字パターンをできるだけ少ない分解数で表現するように考えることは，文字パターンのデザインの問題であるが，文字品質に直接影響することであり，十分な配慮が必要である．一般にドットマトリックス式の場合，ドット分解数による評価は表-2のようである．縦横両方向のドット数が20ドット以下を低品質文字パターン，30ドット以上を高品質文字パターン，この中間を中品質文字パターンと分類される．また60ドット以上になると電算写植システム等の超高品質の文字パターンとして使用することができる．

表-2　ドット構成と文字品質

ドット構成	文字の品質
18×18 ドット	文字の骨組を表現することしかできない．画数の多い字に対しては略字体を用いなければ表現できない．たとえば当用漢字の中でも「雲」「雪」「雷」「鶏」の4字は表現できないほか，約4,000字の漢字に対して約30字の漢字が略字体となる．この文字は帳票用を主とした特殊用途の文字である．
24×24 ドット	ほとんどの文字を略字体なしで正確に表現することができ，明朝体での表現もある程度できる．
32×32 ドット	すべての文字を正確に表現することができ，明朝体，ゴシック体の区別が行える．しかし，20画以上の文字になるとこの区別が難しい文字もでてくる．斜線，曲線の表現が難しい．
64×64 ドット	書体の区別は確実に行えるが，明朝体の線端の表現および，ひらがなの曲線の微妙な差異が表現しにくい．一般の事務用文書や帳票の版下として使用できる．
96×96ドット以上	線の太さ，線端のアクセント，曲線なども十分に表現できる．商用印刷にも使用することも可能である．

当用漢字の全文字種(1,850字)を略記なしで表現する場合，最低20ドットは必要である．最近の漢字プリンタおよびディスプレイには24ドットあるいは32ドットの文字パターンが最も多く使用され，約3,000～16,000字種を表現している．

ストローク式の場合，文字パターンの各ストロークを座標点を使って表現しているが，この座標点はドットマトリックスに分解した格子点の位置情報であることから，原形の文字パターンをドットマトリックス式と同様に捉えることができる．ラインドット式の座標点もこれと同様に考えられる．しかし，自然文字（明朝体文字）をドットマトリックスに分解した文字パターン（自然文字の特徴が表現できる24×24ドット以上）を原形にしてストローク式あるいはラインドット式でそのまま表現したのでは，記憶のためのメモリ量を減少することはあまり期待できない．これは自然文字の「おさえ」や「はね」等あるいは曲線部分の表現に多くの座標点が要求されるためである．

以下に述べる文字パターンの圧縮方法では，上記のような文字パターンの原形を損ねることなく文字発生できることを前提にした方法である．また，それぞれの方法でメモリ量の減少がどれくらい期待できるかを求めるために行った統計調査法とその結果を表に示す．参考に，前述したラインドット式についても調査

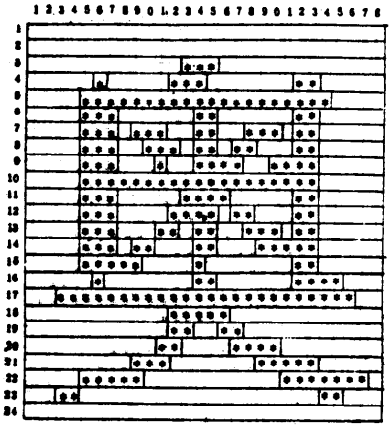

図-1　1行単位の分割

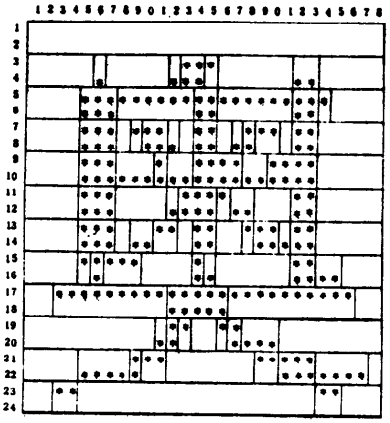

図-2　2行単位の分割

結果を掲げる．

なお，原形にしたドットマトリックスパターンは，横方向28ドット，縦方向24ドットに分割したものである．また文字種は，当用漢字のほか，追加漢字，仮名文字，英数字，記号を含んだ2,583種である．

（1）　ランレングスコード法による圧縮法

この方法は，1行あるいは2行を単位にして，同じパターン（1行単位の場合は白または黒，2行単位の場合は縦に2つのパターン$\begin{pmatrix}白,&白,&黒,&黒\\白,&黒,&白,&黒\end{pmatrix}$）が連続する状態を調査し，この連続する部分をパターンの種類と連続する長さによって表現する圧縮法である．そのため連続する部分が多いほど圧縮率は良くなり，連続しないで交互に別なパターンが出現するようなら悪くなる．

①　1行単位の統計調査

図-1のようにドットマトリックスパターンを行単位に分割して，各行ごとに1ドットずつ，白または黒の小パターンが連続する回数をカウントし，これを全体（28ドット/行×24行）にわたって調べていく．これを全文字種について行う．調査結果は表-3のとおりである．

②　2行単位の統計調査

図-2のようにパターンを2行単位に分割して，各2行ごとに上下2ドットの小パターン$\begin{pmatrix}白,&白,&黒,&黒\\白,&黒,&白,&黒\end{pmatrix}$が連続する回数をカウントし，これを全体（28小パターン/2行×12行）にわたって調べていく．全文字種についての調査結果は表-4のとおり．

（2）　ハフマンの符号化法による圧縮法

この方法は，文字パターンのドットマトリックスを2×2あるいは4×1といったサブパターンに分割した上で，サブパターンの種類ごとの出現確率を求め，この確率からハフマン（Huffman）の符号化法でコード化を行って平均語長を最小にする方法（単純ハフマン法による方法）と，隣接するサブパターン同士の関連性を確率的に求めてコード化する方法（複合ハフマン法による方法）とがある．複合ハフマン法による方法では，分割したサブパターンにあらかじめ順序（左から右，上から下）を決めておき，この順序に従って，各サブパターンの後に続くサブパターンの種類とその

表-3　ランレングス法の1行単位の統計

連続する回数	(a)文字部分(黒)	(b)地の部分(白)	(a)+(b)
1	15,661	15,705	31,366
2	82,594	30,560	113,154
3	20,768	31,773	52,541
4	4,901	26,155	31,056
5	2,532	23,298	25,830
6	1,556	15,580	17,136
7	1,290	10,408	11,698
8	958	8,549	9,507
9	1,178	7,222	8,400
10	1,158	5,358	6,516
11	904	3,220	4,124
12	1,087	3,346	4,433
13	856	5,270	6,126
14	866	1,609	2,475
15	619	944	1,563
16	788	1,044	1,832
17	577	821	1,398
18	620	739	1,359
19	305	515	820
20	338	462	800
21	196	587	783
22	269	340	609
23	256	251	507
24	469	116	585
25	176	54	230
26	134	9	143
27	0	0	0
28	1	9,112	9,113

表-4 ランレングス法の2行単位の統計

連続する回数	(a)(白/白)の部分	(b)(白/黒)の部分	(c)(黒/白)の部分	(d)(黒/黒)の部分	(a)+(b)+(c)+(d)
1	11,511	17,996	18,289	15,007	62,803
2	18,424	7,067	6,913	43,924	76,328
3	16,362	3,352	2,542	6,888	29,144
4	12,917	1,529	1,755	958	17,159
5	10,699	1,093	1,138	284	13,214
6	6,892	969	1,043	150	9,054
7	4,360	669	710	87	5,826
8	3,473	585	629	86	4,773
9	2,942	385	377	94	3,798
10	2,103	283	268	76	2,730
11	1,320	252	182	72	1,826
12	1,309	257	210	131	1,907
13	2,156	116	72	78	2,422
14	695	84	102	65	946
15	392	61	31	42	526
16	455	41	56	65	617
17	380	47	35	30	492
18	317	46	55	30	448
19	232	36	33	5	306
20	249	20	12	29	310
21	333	34	4	15	386
22	187	7	3	10	207
23	129	6	2	16	153
24	67	2	20	41	130
25	41	0	6	7	54
26	8	0	11	0	19
27	0	0	0	0	0
28	2,873	1	0	0	2,874

表-5 単純ハフマン法の統計 (2×2)

サブパターン	サブパターン番号	頻 度	割 合
□	1	238456	0.54951
▨	16	35003	0.08066
▭	4	26696	0.06152
▤	13	25718	0.05927
◪	6	19336	0.04456
◧	11	19263	0.04439
◩	9	9938	0.02290
◪	2	9882	0.02277
▨	15	9635	0.02220
▨	8	9479	0.02184
◪	3	7501	0.01729
▨	12	7451	0.01717
▨	14	7149	0.01647
◪	5	7026	0.01619
◪	7	860	0.00198
◪	10	551	0.00127
——	計	433944	——

出現確率を文字種全体について求める．こうして求めた確率から，16種のサブパターンごとに，後に続くサブパターンのコードをハフマンの符号化法によってつけていく．

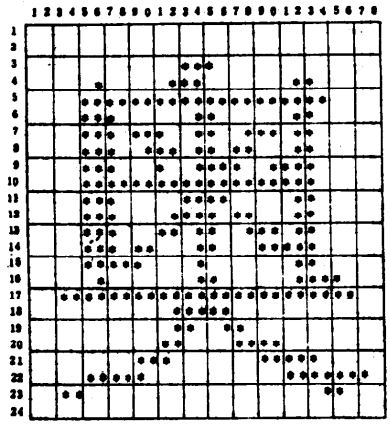

図-3 2×2 サブパターンへの分割

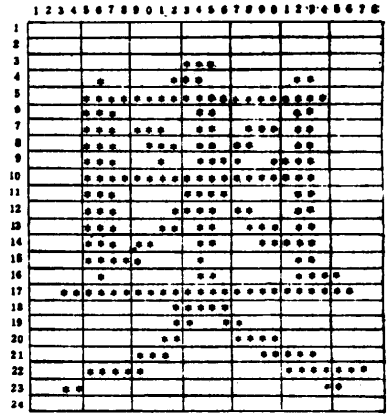

図-4 4×1 サブパターンへの分割

① 単純ハフマン法のための統計調査

文字パターンを2×2と4×1のサブパターンに分割した場合の，各サブパターンの種類ごとに出現確率を文字種全体について求める．調査結果は表-5，表-6のとおりである．

② 複合ハフマン法のための統計調査

文字パターンを2×2と4×1のサブパターンに分割した場合について，それぞれのサブパターンに順序付けし，この順で隣接（後続）パターンの種類ごとに出現確率を求める．なお先頭のサブパターンについては，最初の行の先端（左端）にのみ空白サブパターンを入れ，以下の行は末端（右端）と次行の先端を接続した状態にして統計をとった場合と，各行の先端の前に空白サブパターンを入れた状態にして統計をとった場合の2通りについて求める．それぞれの調査結果

表-6 単純ハフマン法の統計 (4×1)

サブパターン	サブパターン番号	頻度	割合
	1	222169	0.51198
	16	36905	0.08505
	2	28658	0.06604
	13	28009	0.06455
	4	27792	0.06405
	9	27460	0.06328
	7	24169	0.05570
	15	11671	0.02690
	8	10350	0.02385
	10	4245	0.00978
	3	3731	0.00860
	5	3672	0.00846
	12	2475	0.00570
	14	2041	0.00470
	11	355	0.00082
	6	242	0.00056
———	計	433944	———

表-7 複合ハフマン法の統計 (2×2, 最初の行のみ空白サブパターン)

先行するサブパターンの番号	(a) 頻度	(b) 割合	(c)平均語長	(b)×(c)
1	238,491	0.54959	1.73019	0.95090
2	9,882	0.02277	3.10584	0.07073
3	7,501	0.01729	1.62911	0.02816
4	26,696	0.06152	2.29210	0.14101
5	7,026	0.01619	3.06575	0.04964
6	19,336	0.04456	1.97207	0.08787
7	860	0.00198	2.83139	0.00561
8	9,479	0.02184	3.14400	0.06868
9	9,903	0.02282	1.74038	0.03972
10	551	0.00127	2.94374	0.00374
11	19,263	0.04439	1.81436	0.08054
12	7,451	0.01717	2.54529	0.04370
13	25,718	0.05927	2.25803	0.13382
14	7,149	0.01647	2.95719	0.04872
15	9,635	0.02220	2.47628	0.05498
16	35,003	0.08066	3.00220	0.24216
計	433,944	—	—	2.04997 ビット (51.2493%)

(注; サブパターン番号は単純ハフマン法と同じ, 以下同様)

表-8 複合ハフマン法の統計 (2×2, 各行の先頭に空白サブパターン)

先行するサブパターンの番号	(a) 頻度	(b) 割合	(c)平均語長	(b)×(c)
1	240,159	0.55343	1.72706	0.95581
2	9,882	0.02277	3.10584	0.07073
3	6,717	0.01548	1.66264	0.02574
4	26,696	0.06152	2.29210	0.14101
5	7,026	0.01619	3.06575	0.04964
6	19,336	0.04456	1.97207	0.08787
7	860	0.00198	2.83139	0.00561
8	9,479	0.02184	3.14400	0.06868
9	9,242	0.02130	1.77505	0.03780
10	551	0.00127	2.94374	0.00374
11	19,040	0.04388	1.82116	0.07991
12	7,451	0.01717	2.54529	0.04370
13	25,718	0.05927	2.25803	0.13382
14	7,149	0.01647	2.95719	0.04872
15	9,635	0.02220	2.47628	0.05498
16	35,003	0.08066	3.00220	0.24216
計	433,944	—	—	2.04992 ビット (51.2479%)

表-9 複合ハフマン法の統計 (4×1, 最初の行のみ空白サブパターン)

先行するサブパターンの番号	(a) 頻度	(b) 割合	(c)平均語長	(b)×(c)
1	222,170	0.51198	2.05347	1.05133
2	28,658	0.06604	2.51563	0.16613
3	3,731	0.00860	2.36880	0.02037
4	27,792	0.06405	2.75194	0.17625
5	3,672	0.00846	2.43355	0.02059
6	242	0.00056	2.83471	0.00158
7	24,169	0.05570	2.19698	0.12236
8	10,350	0.02385	2.68289	0.06399
9	27,460	0.06328	2.52061	0.15950
10	4,245	0.00978	2.85583	0.02794
11	355	0.00082	2.82535	0.00231
12	2,475	0.00570	3.23313	0.01844
13	28,008	0.06454	2.38646	0.15403
14	2,041	0.00470	2.94659	0.01386
15	11,671	0.02690	2.20538	0.05931
16	36,905	0.08505	2.20991	0.18794
計	433,944	—	—	2.24594 ビット (56.1485%)

は, **表-7**, **表-8**, **表-9**, **表-10** に示すとおりである.

(3) ラインドット式による圧縮法

ラインドット式による文字発生方式は前述した. この方式についての統計調査は, ランレングスコード法による1行単位の分割 (図-1) と同様にして, 文字部分 (黒) の数と空白行の数を文字種全体について求める. 調査結果は次のとおりである.

- 文字部分の数……141,057
- 空白行の数…… 9,112

4. 圧縮方法の比較

前述したランレングスコード法, ハフマンの符号化法, ラインドット法のための各統計調査結果を基にして, 次のような各算出法から圧縮率 (期待値) を求めて比較する.

28×24 ドットマトリックスの原パターンでは, 1字当り 28×24=672 ビット (84バイト), 文字種全体 2,583 字で 1,735,776 ビット (216,972 バイト) のメ

表-10　複合ハフマン法の統計（4×1，各行の先頭に空白サブパターン）

先行するサブパターンの番号	(a) 頻　度	(b) 割　合	(c)平均語長	(b)×(c)
1	240,386	0.55396	2.08229	1.15349
2	28,658	0.06604	2.51563	0.16613
3	3,698	0.00852	2.36857	0.02018
4	27,792	0.06405	2.75194	0.17625
5	2,951	0.00680	2.44832	0.01665
6	242	0.00056	2.83471	0.00158
7	23,991	0.05529	2.19807	0.12152
8	10,350	0.02385	2.68289	0.06399
9	18,568	0.04279	2.61169	0.11175
10	4,245	0.00978	2.85583	0.02794
11	352	0.00081	2.82670	0.00229
12	2,475	0.00570	3.23313	0.01844
13	21,332	0.04916	2.39166	0.11757
14	2,041	0.00470	2.94659	0.01386
15	9,959	0.02295	2.20534	0.05061
16	36,904	0.08504	2.20989	0.18794
計	433,944	—	—	2.25020 ビット (56.2549%)

モリ量となる．

（1）　ランレングスコード法による圧縮率

原パターンの1行の大きさは最大 28 であり，これを表現するのに5ビットを必要とする．またパターンの種類として，1行単位の場合2種類を1ビットで，2行単位の場合4種類を2ビットで，それぞれ表現できる．これに連続する回数を表わす部分の長さ（ビット数）として，1ビットから5ビットを仮定する．

このように想定したビット長から，文字種全体を表わすのに必要な総メモリ量を求めて，原パターンの総メモリ量に対する比率を得ればこれが圧縮率となる．

算出した圧縮率を表-11 に示す．この結果，2行単位で長さを2ビットにした場合が最も圧縮率が良く，およそ 73% である．

（2）　ハフマンの符号化法による圧縮率

単純ハフマン法による場合は，16種類のサブパターンの出現確率を基にして作ったコード表（トリーテーブル）を用いて各文字パターンをコード化し，これを全文字種について行った総コード長が，文字パターンの総メモリ量である．これにトリーテーブルのメモリ量（3バイト×16 サブパターン＝48 バイト）を加えて算出した圧縮率を，2×2 と 4×1 のサブパターンについて求める．この結果は，2×2 の場合で約 65% の圧縮率である．

複合ハフマン法による場合は，隣接するサブパターン同士の関連性（結合度）を求めて作った16種類のトリーテーブルを用いて各文字パターンをコード化し，

表-11　ランレングスコード法による圧縮率

分割単位	ビット長 1	2	3	4	5
1行単位	1.0833	0.9948	0.9719	1.0383	1.1895
2行単位	0.8099	0.7298	0.7445	0.8141	0.9281

表-12　圧　縮　率

圧縮法			圧縮率（%）
ランレングスコード法	1行分割		97.19
	2行分割		72.98
単純ハフマン法	2×2		65.14
	4×1		65.31
複合ハフマン法	2×2	最初の行のみ	51.60
		各行	51.60
	4×1	最初の行のみ	56.50
		各行	56.60
ラインドット法			83.89

これを全文字種について行った総コード長が，文字パターンの総メモリ量である．これに 16 種のトリーテーブルのメモリ量（3バイト×16 サブパターン×16 種＝768 バイト）を加えて算出した圧縮率を，2×2 と 4×1 の各サブパターンの最初の行にのみ空白サブパターンを入れた場合と各行に入れた場合とについて求める．この結果は，空白サブパターンの挿入数にかかわらず2×2の場合で約51.6% である．

なお，トリーテーブルの1サブパターン当り3バイト（24 ビット）の内容は，サブパターンの種類用に4ビット，コード長に4ビット，コード 16 ビットとしている．

（3）　ラインドット法による圧縮率

原パターンの1行の大きさは 28 であるから，X 座標を表現するにも，また連続する長さを表わすにも5ビットで十分である．さらに空白行を識別するのに5ビットを用いて，5ビットすべてがオンの状態と定義することができる．よって統計調査で得られた文字部分の数と空白行の数にそれぞれ 10 と 5 を乗じて加えたのが総メモリ量になる．これを原パターンの総メモリ量で除算して得られた圧縮率は約84% である．

以上，3種の方法で求めた圧縮率を1つの表にしたのが表-12 である．最も圧縮率が良いのは，複合ハフマン法の 2×2 のサブパターンによる方法であり，約 51.6% の圧縮率が得られている．これは原パターンで記憶した場合に約 217 K バイトであることから，104 K バイトのメモリを節約できたことになり，文字

パターンメモリは約113Kバイトで十分なことを示している．

5. おわりに

日本語文字発生方式の終局的な目標は，安価で高速に高品質文字を発生し表示出力できるようにすることである．しかしながら，高速アクセスメモリを用いれば高価になり，高品質文字を発生できるようにドットやストロークの分割数を増せば大量のメモリを必要としてやはり高価になる．

この解決策は，安価で高速な大容量メモリを開発することである．最近のLSI技術の革新をみると，その実現も近いと予測されるし，一方では新しい記憶素子の開発も進められているので，ますます期待が大きいものとなっている．メモリに関する一般的な動向を，ビット当りのメモリコストを示した図-5と，各種メモリのアクセスタイムの概略を示した図-6からみると磁気バブルメモリが最も期待できるように思われる．実用化も除々に進められており，MOS, ROMと共に近い将来，文字パターンメモリとして多数利用されることであろう．

一方，もう1つの課題として，文字パターンの標準化の推進がある．これまで，文字パターンは各メーカが独自にデザインをして，各種の字形をした文字パターンを自由に使ってきた．その品質の差が商品としての価格に影響しているわけであり，またマスクROMの大量生産を妨げてきた原因でもある．これを標準的な文字種についてパターンを規格化すれば，マスクROMの大量生産が可能になり，統一された品質のもとに低価格化を図ることができるようになる．

本解説でとりあげた文字パターンの圧縮によるメモリ量の減少法は，こうした課題が解決されるまでの過渡的技術として重要なことである．

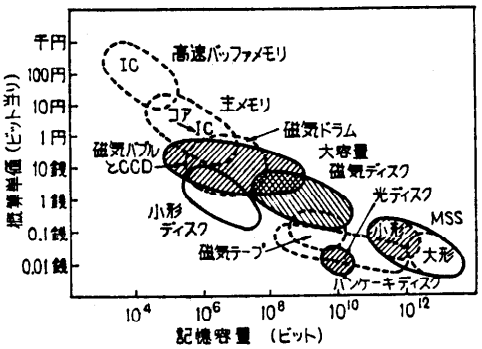

図-5 ビット当りのメモリコスト

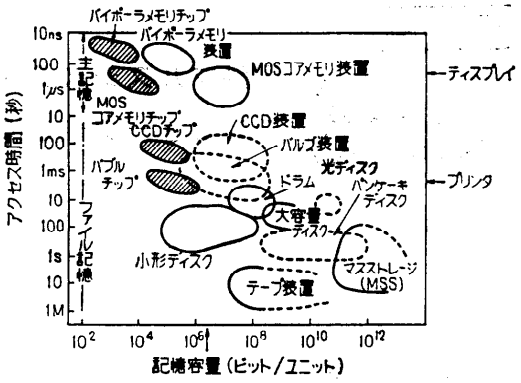

図-6 各種メモリのアクセスタイム

最新の高速プリンタは，レーザビームによる電子写真方式で普通紙へ大量・高速印刷が可能になり，これに出力装置の低価格化が図れれば，さらに日本語情報処理の普及拡大が期待できる．

参考文献

1) 日本情報処理開発協会：日本語情報処理技術の研究開発，52-S 001分散型リソース処理技術の研究開発，pp. 471-582 (1978).
2) 日本情報処理開発協会：日本語情報処理技術の研究開発，53-S 001分散型リソース処理技術の研究開発，pp. 189-370 (1979).
3) 日本情報処理開発センター（現協会）：日本語情報処理の技術動向調査報告書 (1973).
4) 長谷川実郎：パターン合成による漢字入出力処理，情報処理，Vol. 16, No. 9, pp. 808-817 (1975).
5) 坂井利之，長尾 真，寺井秀一：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).
6) 祢津孔二：画素間の相互情報を利用した文字パターンの符号化法，電子通信学会誌，'72/4-Vol. 55-D, No. 4, pp. 277-284 (1972).
7) 内藤祥雄，南谷 崇：PROMによる漢字発生器，電子科学，1976年5月号，pp. 37-43.
8) 尾上守夫，岩下正雄：計算機内における画像のデータ圧縮，情報処理，Vol. 18, No. 8, pp. 776-780 (1977).
9) 石田真也，永原隆嗣，小西良往：漢字パターンデータの一圧縮方式について，情報処理学会論文誌，Vol. 20, No. 2, pp. 99-104 (1979).
10) 福沢空也：ホログラムメモリと漢字システム，bit, Vol. 19, No. 12, pp. 51-57 (1977).
11) 小谷進太郎，大迫昭和，安孫子一松：各種漢字パターン・メモリーの実例と得失，日経エレクトロニクス，pp. 126-148 (2. 20, 1978).
12) 小池博之，野村仙一：日本語端末の文字発生方法，情報処理学会第20回全国大会講演論文集，1 F-6 (1979).

（昭和55年6月30日受付）