解　説

# 出力技術としての漢字パターン処理†

森　克己††

## 1. まえがき

最近，ワードプロセッシング，日本語情報処理あるいはオフィスオートメーションなど，さまざまな形で情報処理技術が社会活動へ導入されてきている．

我が国では，このような情報処理システムにおいて漢字の取り扱いが不可欠である．漢字処理技術には多くの困難な問題が山積しているが，ここでは，出力技術としての漢字パターン処理について述べる．

漢字パターンの出力技術は経済的な漢字パターンの発生技術に集約できる．漢字パターン発生の経済化には2通りの方法があり，1つはパターンメモリを減らすデータ圧縮であり，他の1つは本資料で述べる変換技術である．すなわち，1種類の基準パターンをもとにサイズや書体の異なる漢字パターンを発生させることにより等価的に経済化を図るものである．

以下，漢字ドットパターン（漢字パターンと略す）を対象として，2章では漢字パターンの作成技術としての整形処理技術，3章では次数変換処理技術，4章では書体変換処理技術について技術の現状を述べる．

## 2. 漢字パターンの整形処理

### 2.1 整形処理の必要性

漢字パターンを出力するためには必要な漢字パターンを作成しなければならない．

漢字パターンは，従来，デザイナらの人手によって作成されていたが，システム対応に種々のパターンを用意することには対応できなくなってきた．そこで，より効率的な作成法として，印刷された漢字母形をサンプリングしディジタルのドットパターンを得る方法が検討されるようになってきた[1),2)]．

しかし，漢字母形を単純にディジタル化して得られるドットパターンは線分の周りの凹凸（以下，ノッチと呼ぶ）や，線幅の不揃いのために漢字パターンとして使用することはできない．そこで，ディスプレイ装置を介して計算機と人間が会話処理を行うことにより，パターンを修正する作業が必要となるが，このノッチの除去と線幅の規格化に修正作業の多くが占められ，その自動化が必要となる．

一方，このノッチの発生と線幅の不揃いは印刷されたパターンをディジタル化する際に常に問題となるものであり，漢字パターンの作成においてのみならず，ディジタルファクシミリ電送での文字品質の劣化や帯域圧縮効率の低下などを引き起こすことになる．

漢字パターン整形処理技術はこのような分野においても要求される．

### 2.2 整形処理[3)]

漢字パターンには縦，横直線が多く含まれていることは良く知られているが，通常使用される 32×32 ドット程度の大きさの漢字パターンではパターンの表現に大幅な制約を受け，線分の微妙なニュアンスの表現は不可能となり次のような特徴が追加される．

① 縦，横方向に近い傾きの線分はすべて完全な縦，横直線としてしか表現できないため，縦，横直線の割合が活字パターンに比べて大きくなる．

② 線幅は1～3ドットの範囲で，ドット単位で量子化される．

③ 斜線，曲線の滑らかな表現は困難であり，ある程度の品質劣化は避けられない．

④ ドット配置の乱れは縦，横直線部で特に目立ち，斜線，曲線部ではあまり目立たない．

以上のような，ドット表現された漢字パターンの性質を考慮すると漢字パターンの整形処理は縦，横直線と斜線，曲線の分離処理が有効となる．

**図-1** に整形処理の流れを示す．大きく次の4つの処理から成っている．

（1） 縦，横直線と斜線，曲線の分離

この縦，横直線と斜線，曲線の分離精度が整形処理全体の効果を左右することになる．この分離処理の手

† Processings for Kanji Pattern Output by Katsumi MORI (Yokosuka Electrical Communication Laboratory, N.T.T.).

†† 日本電信電話公社横須賀電気通信研究所画像応用研究室

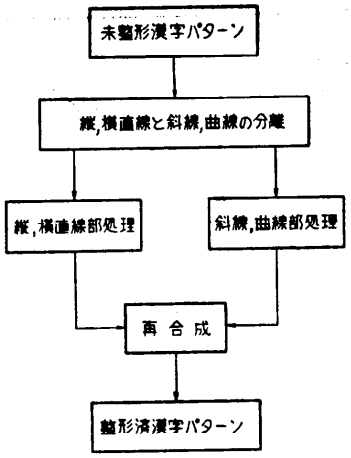



図-1 漢字パターンの整形処理の流れ

| | | |
|---|---|---|
| d | d | d |
| $\phi$ | a(i,j) | $\phi$ |
| d | d | d |

| | | |
|---|---|---|
| d | $\phi$ | d |
| d | a(i,j) | d |
| d | $\phi$ | d |

d: dont Care

If
$a(i+1,j)=\phi$ and $a(i-1,j)=\phi$
or
$a(i,j+1)=\phi$ and $a(i,j-1)=\phi$,
then $a(i,j)=\phi$

図-2 ノッチ除去フィルタ

法としては縦，横方向へ，所定の長さ以上連なる黒ドットを“ラン”として抽出し，さらに，隣接し合うランを“ラン群”として統合した後，ラン群の幅，傾き，両端の状態から，縦，横直線を構成するラン群か否かを判定することにより行っている．

（2） 縦，横直線部処理（線幅の規格化）

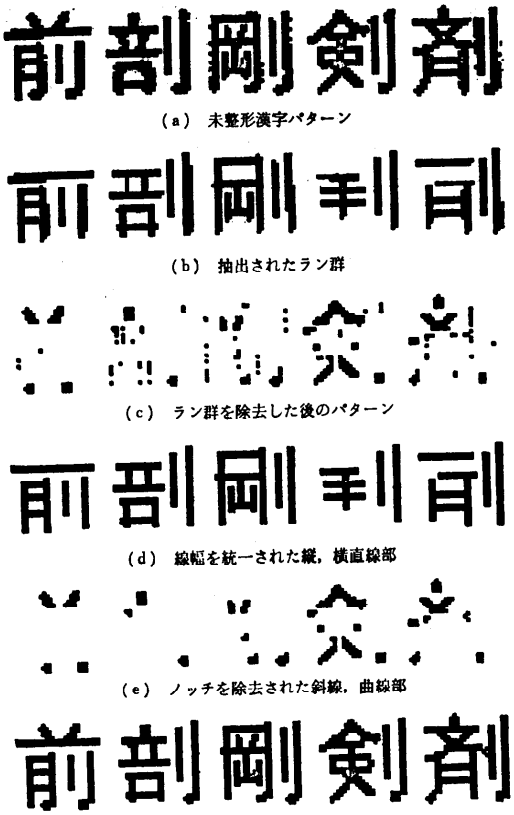
縦，横直線の構成ラン群と判定されたラン群から，直線の始点と終点を検出し，それらの始点，終点を用いて所定の幅の直線を新たに発生することにより線幅が統一された漢字パターンの縦，横直線部を作成する．

（3） 斜線，曲線部処理（ノッチの除去）

未整形の原漢字パターンから縦，横直線のラン群を分離した後には斜線，曲線と直線に付随するノッチが残される．このうち，ノッチは幅1ドットの短直線あるいは孤立点として現われるため，図-2に示すフィルタで容易に除去することができる．

（4） 再 合 成

（2），（3）の過程で得られた線幅が統一された縦，横直線部と，ノッチを除去された斜線，曲線部を合成することにより，整形処理された漢字パターンが得られる．



（a） 未整形漢字パターン

（b） 抽出されたラン群

（c） ラン群を除去した後のパターン

（d） 線幅を統一された縦，横直線部

（e） ノッチを除去された斜線，曲線部

（f） 整形後の漢字パターン

図-3 整形処理過程での漢字パターンの例

図-3に一連の処理過程でのパターンの例を示している．

## 3. 次数変換処理

1枚の文書の作成においても本文用，見出し用，脚注用，さらには図表への挿入用と種々のサイズの漢字パターンが必要となる．さらに，ディスプレイやハードコピー装置を含む複合システムでは，その種類はさらに多くなる．

このように多くの種類の漢字パターンをすべてパターンメモリに記憶しておくためには膨大なメモリ容量が必要となる．たとえば，16×16，24×24，32×32，64×64 の4種類の漢字パターンを JIS 第2水準まで含めて6,500字種ずつ記憶する場合には約5メガバイトのメモリ量となり，メモリが安価になったとはいえ装置構成時の大きなネックとなる．そこで，1種類の基準漢字パターンから次数（漢字パターンの縦，横方向のドット数を次数と呼ぶ）の異なる漢字パターンを

作成する次数変換処理の研究が行われるようになってきた[1]-[3].

漢字パターンの次数変換法としては種々のものが提案されているが，大きく

① 漢字パターンの特徴を利用した次数変換法

② 図形としての次数変換法

の2つに分けることができる．以下，この分類に従って，いくつかの方法について概説する．ただし，各方法の名称は筆者が便宜上つけたものも多い．

なお，次数変換処理の検討に際しては次のような条件が一般に設定される．

① 誤字とならないこと

② 縦，横直線の線幅がそれぞれ統一されること

③ 文字バランスが劣化しないこと

④ 大きな次数で斜線が滑らかであること

### 3.1 漢字パターンの特徴を利用した次数変換法

漢字パターンの次数を変えるためにはドットの挿入あるいは削除を行うことが必要である．したがって，次数変換処理は挿入・削除を行うドットの選択問題に帰着される．以下，この節では漢字パターンが縦，横直線を多数含んでいるという特徴を考慮して，ドット挿入・削除を行および列を単位として行う方法について述べる．

（1） 行・列選択法[4]

各行および列（行および列を行［列］と略記する）内に含まれる，最も長い黒ドットのランの長さを調べ，その長さの短い行［列］から必要な個数だけ挿入・削除する行［列］を選択する方法である．この方法では行［列］の選択が局所的判断で行われるため，文字バランスの劣化が大きいという欠点がある．

（2） 線分の比例配置法[4]

この方法は行・列選択法の文字バランスの劣化をなくし，かつ，変換後の漢字パターン（以下，変換パターンとよぶ）の線幅を揃えるようにした方法である．

図-4 に線分の比例配置法による変換前の原パターンと変換パターンとの間の行［列］番号の対応関係を示している．図で横軸と縦軸はそれぞれ原パターンと変換パターンとの行番号を，また，破線は理想的な比例変換を行った時の対応関係を示している．

原パターンには番号 5, 6, 12, 13, 16, 17 の行に幅2ドットの横直線が，また，14, 15 の行には幅2ドットの最小線分間隙が含まれている．番号 5, 6 の行に含まれる横直線は ■—■ の対応関係により幅1ドットとして比例変換位置に配置される．また，行番号

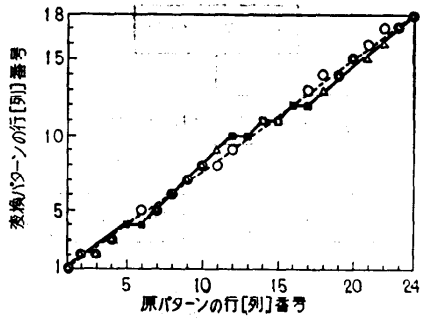



図-4 線分の比例配置法の説明図

12～17 に存在する2本の横直線と最小線分間隙は高線分密度部として一体化され，破線（比例的対応関係）上に最も重なるように置かれた ■—■—□—□—■—■ の対応関係により，それぞれ幅1ドットとして配置される．その後，各線分の間をそれぞれ比例的に配置する．

このように，本方法では□と■印で示される対応部の形状を決めることで線幅と最小線分間隙幅を制御し，これらの対応部を比例変換位置にできるだけ重なるように配置することで文字バランスの劣化を最小とするように制御している．

縦，横直線の検出は周辺分布のピークを検出することにより行っている．

図-5 に変換パターンの例を示す．

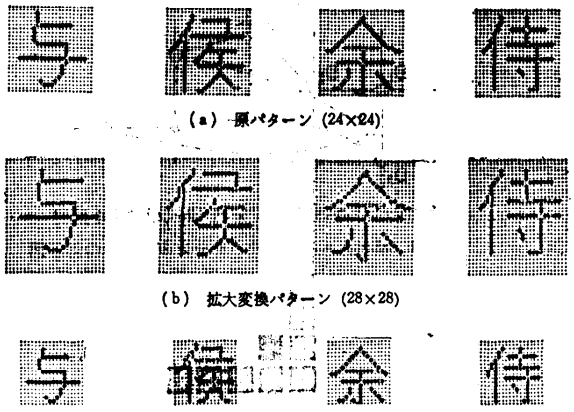
（3） 制御情報付与方式[5],[7]

前述の線分の比例配置法は自動的に行［列］番号の対応関係を作成するため，次数変換用に付与すべき制御情報はなかったが，処理時間が長く変換速度が遅いという問題があった．

そこで，挿入・削除すべき行［列］をあらかじめ求め，たとえば，反復挿入あるいは削除すべき行［列］に "1"，そうでない行［列］に "$\phi$" の，制御ビットを付与しておくことにより変換処理の高速化を図ることができる．

この場合には，$m \times n$ ドットの原パターンから変換パターンを得るためには，変換パターン1個当り（$m+n$）ビットの制御情報が必要となる．これは，32×32 の原パターンから 48×48 の変換パターンを得る場合を例にすれば 1/36 にデータを圧縮したことに相当する．

この方法は変換速度が速いことに加えて，変換パターンの文字品質を制御情報の作成時に一度確認できることが実用上大きな特長となろう．しかし，このこ



(a) 原パターン (24×24)

(b) 拡大変換パターン (28×28)

(c) 縮小変換パターン (18×18)

図-5 線分の比例配置法による変換パターン例

とは逆に，変換パターンごとに制御情報を作成する必要があり大きな作業量を伴う．そこで次のような制御情報の自動作成法が考えられている．これらの方法は処理時間さえ許せば，当然，独立した次数変換法として用いることが可能である．

① 線分の比例配置法

前述した通りである．

② 区間分割処理法[5]

原漢字パターンを行［列］方向に変換次数の差と同じ区間数に等分割し，各区間内で行［列］に優先順位を付けて挿入・削除の可否を判断する方法である．この時，縮小する場合には字画を含む行［列］は削除不可とし，拡大する場合には，まず，字画を含む行［列］から反復挿入することにより字画の幅を統一するように考慮されている．

しかし，この方法では一部に誤字や線幅の不揃いが生じることは避けられないため，人手による制御情報の修正が必要である．

③ 半自動処理

図-6 は人手によって作られた制御情報により 32×32 の原パターンから 18×16 の変換パターンを作成した例であるが，初めから 18×16 の漢字パターンとしてデザインされたパターンと比較して遜色のないパターンが得られている．

そこで，この制御情報の作成過程で人間が行った判断を統計的に解析して，その結果を用いて制御情報の自動作成を行う方法が考えられる[7]．

ドットの状態を，それを囲む8個のドットを含めて，3×3 のマトリクスパターンとして定義し，数百字についてデザイナが行った制御情報の作成過程で，各状態のドットが削除されなかった割合をドットの重要度と定義する．さらに，各行［列］内に含まれるドットの重要度の和として行［列］の重要度を求め，重要度の低い行［列］から順次，必要な個数だけ削除すべき行［列］と判断する方法である．

図-7 に変換パターンの例を示している．

### 3.2 図形としての次数変換法

ここでは，漢字パターンを一般の2値図形とみて次数変換を行う方法について述べる．

(1) 黒ドットの比例配置法[4]

漢字パターンを構成する黒ドットの位置座標に変換倍率を乗じて比例的に写像する最も単純な方法である．

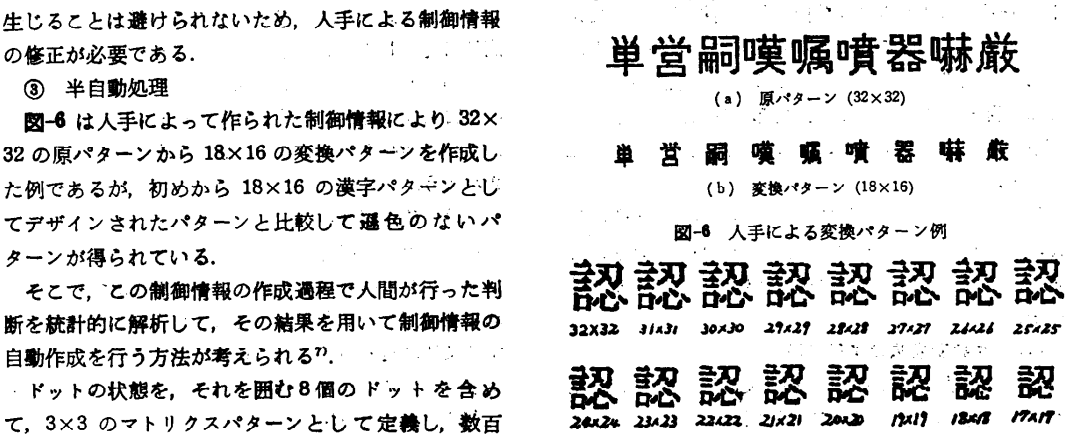
ドットの大きさを 2×2 や 3×3 倍の大きさにする整数倍の拡大法もこの方法の特殊な場合である．

この方法は処理が単純で，かつ，本質的に文字バランスの劣化がないという特長をもつ反面，縮小時の線分間隙の消滅や，拡大時の線幅の不揃い，斜線部の品質劣化という欠点がある．

しかし，変換倍率の大きな拡大変換を行う場合には1～2ドットの線幅の不揃いは問題でなく，文字バランスの保存と斜線部の滑らかさが文字品質上重要となる．そこで，斜線部のスムージング処理（補間処理）と併せて，拡大次数変換法として用いられることが多い[6],[8]．

(2) 部分パターンによる分割処理

原漢字パターンを 2×2～4×4 程度の部分パターンへ分割し，各部分パターンを単位に次数変換する方法である．この方法では変換倍率ごとに変換テーブルを



(a) 原パターン (32×32)

(b) 変換パターン (18×16)

図-6 人手による変換パターン例



図-7 次数縮小変換例（原パターン 32×32）

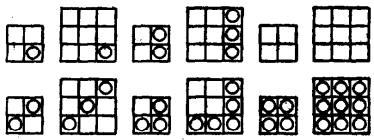


図-8　3/2 倍拡大用変換テーブルの1部

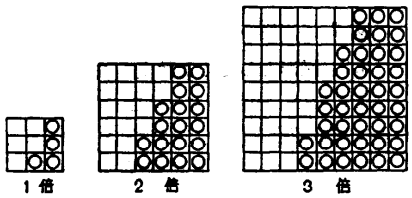



図-9　補間拡大用部分パターンの例

もつことが必要である．図-8 に 3/2 倍に拡大する場合の変換テーブルの例を示す．

また，倍率の大きな拡大変換の場合には 図-9 に示すような補間拡大用の部分パターンを用いて斜線部等の平滑化を図っている[6]．

さらに，分割処理の1種として，3.1 節の(3)で述べた区間分割処理法と同様に，各行［列］を変換次数差と同じ個数の1次元ブロックに分割し，各ブロックから1ドットずつ削除することにより縮小変換する方法もある[9]．

(3)　曲面補間による次数変換[10]

図-10 に示すように，原パターンと変換パターンの間にはドットの1対1の対応関係はつかず，一般に，変換パターンのドットの原パターン上での対応点Pはドットの間に位置し，ドットの状態(白または黒)を一意的に決めることができない．そこで，漢字パターンを多値パターンと見て，点Pの値を周囲の4個のドット $D_1$〜$D_4$ の値から2変数補間(曲面補間)によって求め，その後，閾値処理により2値化する方法である．

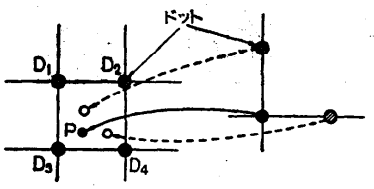
この方法では，曲面補間することから推測されるように，図-11 に示すように直角部分の内側に余剰の黒ドットが発生しやすいため，これを除去する処理が施される．

以上，次数変換法としていくつかの方法を概説した．これらのほかに直線分離法や差分パターン法があるが，これらについては書体変換処理として4章で述べる．

### 3.3　次数変換法の適用領域

次数変換法は変換範囲，変換速度，変換文字品質，制御情報の有無，などによって適用領域は異なってくる．

(1)　適用領域の分類




原パターン上のドット配列　変換パターン上のドット配列

図-10　原パターンと変換パターンのドットの対応図

図-11　直角部分内側の雑音

(A)　適用領域A

文書編集などへの適用であり，本文用の漢字パターンを原パターンとしてもち，それ以外の標題用，脚注用などの漢字パターンを次数変換処理により発生させる．この適用領域では本文用の漢字パターンが他の大きさの漢字パターンと比較してはるかに多く使用されるため，原パターンの発生速度に比べて変換パターンの発生速度は1〜2桁程度遅くても許される領域である．一方，図表まで含めた編集を考えると，文字サイズは特定できないため制御情報を用いない変換法が望ましい．

また，記録として残せるだけの良好な変換文字品質でなければならない．

(B)　適用領域B

テレビ受像機やファクシミリ受信機など解像度の異なる装置へ，それぞれ適切な大きさの漢字パターンを次数変換処理を用いて供給する場合である．

この適用領域ではすべての大きさの漢字パターンを高速に発生してやることが必要となる．文字品質は，表示用の漢字パターンは記録用のそれに比べると若干の品質劣化は許容されると考えられるので，記録用の高次の漢字パターンからの高速な縮小変換技術が要求されよう．

(C)　適用領域C

梱包ラベルや荷札などの宛名書き等のスタンドアロン形システムへの適用である．この適用領域ではきわめて広範囲での拡大変換が要求されるが変換速度に対する条件は緩やかである．変換文字品質については，文字バランスの良さ，斜線部の滑らかさが重要となる．

(2)　次数変換法の比較

表-1 に前述の次数変換法（直線分離法と差分パター

表-1 次数変換法の比較

| 分類 | 変換方式 | 変換範囲*) | 処理量 | 変換文字品質 | 適用領域 | 備考 |
|---|---|---|---|---|---|---|
| 漢字パターンの特徴を利用した次数変換 | 行・列選択法 | 小 | 中 | 文字バランスの劣化大. | | 基本的な提案. |
| | 線分の比例配置法 | 中 | 多 | 文字バランスの保存，線幅の統一，共に良い．斜線部の劣化. | A | 線幅の制御可. |
| | 制御情報付与方式 | 小 | 少 | | B | 変換倍率ごとに制御情報が必要. |
| | 直線分離法 | 中 | 多 | | A | 書体変換も可能. |
| 図形としての次数変換 | 黒ドットの比例配置法 | 中 | 少 | 文字バランスの保存良．線幅の不揃い. | A | 処理が単純. |
| | 補間拡大法 | 大 | 多 | | C | 斜線部のスムージング. |
| | 部分パターンによる分割処理 | 中 | 中 | | A | 変換倍率ごとに変換テーブルが必要. |
| | 曲面補間法 | 大 | 多 | | C | 多値パターンとして処理. |
| | 差分パターン法 | 中 | 多 | 良好. | A | 書体変換も可. |

*) 変換範囲　小：1/2 から高々2倍程度　中：高々3～4倍　大：5倍以上

ン法は次章で述べる）の比較を示す．ただし，各方法について同一条件で客観的に比較された研究はないので，筆者の主観を混じえた概略比較である．表1で処理量は計算機シミュレーションでのダイナミックステップ数を念頭においたものである．

同表に示すように，一般に，漢字パターンの特徴を利用した次数変換法は比較的変換範囲の狭い適用領域A，Bに，図形としての次数変換法は適用領域Cに適しているということができる．

## 4. 書体変換処理

我が国では代表的な書体として明朝体とゴシック体が用いられている．ここでは，これらの書体間の変換処理について述べる．

次数変換処理が基本的にはパターンの相似変換であるのに対して，書体変換は形状の変換であるため，より高度な処理が要求される．

（1） 直線分離法[11]

明朝体とゴシック体の主要な相違点は

① 縦，横直線の線幅比

② 縦，横直線の線端飾りの有無

にある．そこで漢字パターンから縦，横直線を分離抽出し，その線幅と線端飾りの有無を制御することにより書体変換を行うことができる．

線幅の制御については2章の整形処理で述べたのでここでは，線端飾りの制御について述べる．

図-12 に代表的な線端飾りを示す．

今，漢字パターンから抽出された縦，横直線をそれぞれ，{(始点座標)，(終点座標)} で表現して，

縦直線；$\{(I_{sv}, J_v), (I_{ev}, J_v)\}$

横直線；$\{(I_h, J_{sh}), (I_h, J_{eh})\}$

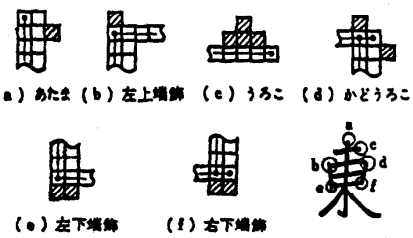



図-12 線端飾りの種類

とし，「カドウロコ」を例に述べる．

ある横直線の終端が，いずれかの縦直線の始点に一致するとき，すなわち，

$I_h = I_{sv}$，かつ，$J_{eh} = J_v + 1$

が成立するとき，2点 (図-12(d)の斜線部)

$(I_h - 1, J_{eh} - 1)$ と $(I_h + 1, J_{eh} + 1)$

のドットの状態を反転することにより，「カドウロコ」の付加，あるいは削除を行う．

以下，同様にして縦，横直線の線端飾りの制御が行われる．

図-13 に，2章で述べた黒ドットのランを用いて縦，横直線を分離抽出した後，書体変換処理を行った例を示している．

なお，この直線分離法は直線の始点，終点の座標変換を行うことにより容易に次数変換法としても適用できる．

（2） 差分パターン法[12]

拡大次数変換により漢字パターンの高品質化を行う場合でも，書体変換においても，パターンに対して新たな情報を付加してやることが必要である．しかし，付加すべき情報の量は，同一字種間の変換であれば，


(a) ゴシック体


(b) 明朝体

図-13 書体変換例

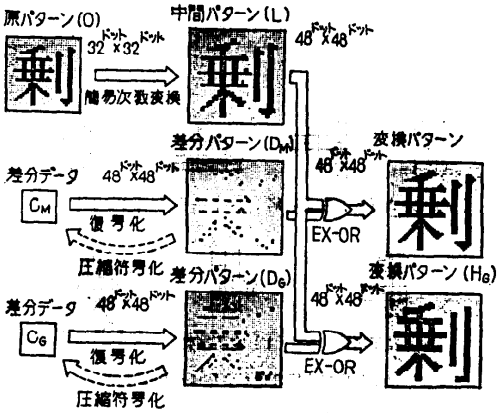



図-14 差分パターン法の原理

独立した漢字パターン1個のデータ量より少ないことが期待できる．そこで，図-14に原理を示すように，変換パターンと原パターンの差分パターンのデータを制御情報としてもつことにより，少ないデータ量で変換パターンを得ようとする方法である．

特に，次数変換を伴う場合には，原パターンを簡易な次数変換処理により変換パターンと同一次数のパターン（中間パターン）へと変換した後，変換パターンとの差分をとることで差分パターンのデータ量を減らす工夫がなされている．

簡易次数変換法として，3.2節で述べた分割処理の1種である一次元ブロックマッピングを用いて，独立して漢字パターンをもつ場合に比べて1/4～1/6のデータ量で32×32から48×48への次数変換が，また，1/3～1/4のデータ量で次数書体変換が可能となる．

なお，この方法では効率的な差分パターンの作成法が問題となる．

## 5. むすび

現在用いられている活字の種類を考えると，情報処理システムの本格的な普及とともに，システムに要求される漢字パターンの種類はきわめて多くなるものと予想される．これらの多くの種類の漢字パターンをメモリ内にもつことは，いかにメモリが安価になってきたとはいえ，大きな経済的負担となる．

そこで，本資料ではこの問題の解決法の1つとして漢字パターンの変換技術をとり上げ，その技術の現状について述べた．

これらの漢字パターン変換技術の研究は未だ始まって日が浅いため，必ずしも満足すべきものではないが，たとえば，制御情報を用いた次数変換法や補間処理を併用した拡大法など，適用領域によっては充分実用に供し得るものがあると考える．

この分野には，新たな漢字パターン変換手法の開発や，LSI技術を用いたハードウェア化，さらには，基本的な文字品質評価法など多くの研究課題が残されているが，これらの研究の進展により，我が国固有の漢字情報処理システムの普及に貢献できる多くの技術が生み出されることが期待される．

## 参考文献

1) 小田，福森，金出：マン・マシン対話方式によるドット文字，ドットパターン作成方式とその実施例，情報処理，Vol. 16, No. 8, pp. 685-691 (1975).
2) 小畑他：漢字パターン処理システム，51年信学全大，No. 1133.
3) 森，中野：漢字ドットパターンの作成処理，情報処理学会イメージプロセッシング研究会 17-1 (1978).
4) 森：ドット漢字パターンマトリクスの次数変換法，信学論，Vol. 60-D, No. 10 (1977).
5) 渡部他：漢字パターンの拡大・縮小法の一考察，信学技報 IE 79-2.
6) 井上，木村，武川：漢字パターンの拡大・縮小法，信学技報 IE 79-1.
7) 中野，森：ローカルな性質を利用した漢字パターンの次数縮小変換，信学技報 IE 79-3.
8) 大川他：漢字パターン拡大方式の検討，画像電子学会予稿 76-05-2.
9) 斉藤，松葉，杉山：インクジェット式漢字プリンタ用画素形文字構成法，信学論 Vol. J 62 D No. 11.
10) 塩野，真田，手塚：漢字ドットパターンの次数変換と整形の一手法，信学論，Vol. J 63-D, No. 7.
11) 中野，森：直線分離法による漢字パターンの処理，信学論，Vol. 62-D, No. 12 (1979).
12) 小川，中根：漢字パターン変換処理方式，信学技報 IE 79-91.

(昭和56年3月12日受付)