# 漢字パターンデータの一圧縮方式について†

石田真也†† 永原隆嗣††† 小西良往††

本論文では，鉄道における乗車券発行システムに関連して開発された漢字パターンデータ圧縮方式について述べている．

本圧縮方式の特徴は，

（1） 漢字自身をあらかじめ圧縮に都合の良いようにいくつかの種類の基本図形—直線，長方形，平行四辺形，四辺形—のみで構成したこと，

（2） 各基本図形の位置，大きさ等を2次元ベクトルの系列として表わし，各ベクトルをその出現頻度を考慮し，可変長符号を用いて効率的に符号化したこと，

などである．

本圧縮方式を定期乗車券発着駅名用漢字（32×32 ドット）に適用した結果，平均約 1/3.2 の圧縮率を得た．

## 1. まえがき

鉄道駅務システムにおいて，漢字記号をドットパターンとしてコンピュータにより記憶，処理することは，定期券発行機，自動券売機，あるいは，TV 画面上に列車の種別，行先等をドット文字で表示する案内表示装置等において，広く用いられるようになってきている．この種の漢字パターン処理システムにおける共通の課題の一つは，いかに効率良くパターンデータを圧縮し，記憶あるいは伝送するかということである．

たとえば近鉄におけるオンライン定期券発行システム[1]では，駅窓口の端末装置から入力された乗車区間，期間，券種等のデータはセンターコンピュータに送られ，センターコンピュータで運賃計算，発売データ管理などの処理が行われた後，発券に必要なデータが返送され，端末装置のドットプリンタにより印刷発行されている．

定期券印刷に必要な文字パターンのうち固定的に使用される文字については，端末装置内に保持され，発着駅名用漢字など種類の多いものについては，センターコンピュータよりの返送データとして端末装置に供給される．

このセンターコンピュータより伝送される発駅名，着駅名はそれぞれ漢字 2～4 文字より構成され，データ圧縮しないとすれば最大約 800 バイトのデータとなる．センターコンピュータより返送されるデータには，発着駅名パターンのほか，必要に応じて経由表示用漢字パターンも含まれ，回線使用効率向上，あるいは，端末装置での発券応答時間の短縮の観点から，これらの漢字パターンデータをいかに能率良く圧縮するかが重要な問題となる．

図形ドットパターンを圧縮する問題は，ファクシミリにおける帯域圧縮，衛星写真等の画像処理，TV 画像伝送等に関連して各種の方式が報告されている（例えば文献 2) 参照）．これら既存の圧縮方式が対象としている図形が，印刷物，写真等の一般画像であるのに比べ，乗車券等に使用される漢字は，いくつかの固有な特徴を有し，これらは次のように要約される．

（1） パターンそのものを人工的にデザイン，作成することが可能である．

（2） パターンデータそのものをあらかじめ圧縮されたデータとして用意することができ，もとのパターンから圧縮データを得る“符号化”は実時間で行う必要はない．

（3） 乗車券の発券に際して一時に必要とする漢字は比較的少量であり，圧縮データよりもとのパターンを得る“パターンの復号”はそれほど高速性を必要としない．

（4） パターンの復号はマイクロコンピュータ等の電子計算機で行われるのを前提としてよい．

また一般の漢字処理システムと比べても，

（1） 定期券のように大小様々の字体を同時に印字する必要がある，

（2） 乗車券あるいは TV 案内表示装置などの性質上比較的大きな字体が要求される場合が多い，

† An Algorithm for Chinese Character Pattern Data Compression by SHINYA ISHIDA, YOSHIYUKI KONISHI (Research Laboratory, Kinki Nippon Railway Co., Ltd.) and TAKATSUGU NAGAHARA (Industrial Machinary Dept., Kinki Sharyo Co., Ltd.).

†† 近畿日本鉄道(株)技術研究所

††† 近畿車両(株)産業機械部

（3） 字体は“質”よりも見易さ，わかり易さが重視される，

などをその特徴として挙げることができる．

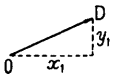
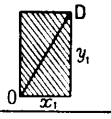
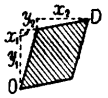
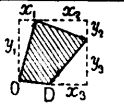
本文で述べる圧縮方式は，これらの点を考慮して，漢字パターン自身を圧縮に都合の良いいくつかの基本的図形（直線，長方形．平行四辺形，四辺形）のみで構成し，それらの基本的図形をいくつかのベクトルの系列で表現し，更に，それぞれのベクトルをその出現確率に応じた可変長符号により符号化することを特徴としている．以下本文では，乗車券発着駅名用漢字(32×32 ドット）について，圧縮の具体的手法，効果等について述べることとする．

## 2. 圧縮データの構成

本圧縮方式においては，すべての漢字パターンはあらかじめ**表1**に示す4種類の基本図形（直線，長方形，平行四辺形，四辺形）の組合せにより作成される．各基本図形は大きさの変化が許され，かつ長方形以外は回転が許されているので，漢字パターン作成に際しこの制約を課しても構成される文字パターンの自由度には実質的にほとんど影響を与えない．

データ圧縮は，これらの基本図形の種類，形状，相対位置関係を順次符号化することにより行われる．いま一つの漢字パターンが $n$ 個の基本図形から構成されるとき，それらの基本図形を符号化される順番に $F_1, F_2, \cdots, F_n$ とする．このとき，各基本図形 $F_i$ の種別，位置，形状を規定する符号ブロックをそれぞれ $C_i$，

**表1** 基本図形及びそのベクトルによる表現
Table 1 Basic elements and their vector representations.

| 基本図形 | 始点・終点 | ベクトルによる表現 (O---始点, D---終点) |
|---|---|---|
| 直線 | 両端点 |  $(x_1, y_1)$ |
| 長方形 | 相対する2頂点 |  $(x_1, y_1)$ |
| 平行四辺形 | 相対する2頂点 |  $(x_1, y_1), (x_2, y_2)$ |
| 四辺形 | 相隣る2頂点 |  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ |

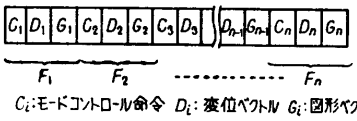



**図1** 圧縮データの構成
Fig. 1 Compressed pattern data format.

$D_i, G_i$ とし，これらを用いて漢字パターンの圧縮データを**図1**のように構成する．符号ブロック $C_i, D_i, G_i$ の各構成法を次に述べることとし，以下では，それらの役割，構成を考慮し，$C_i, D_i, G_i$ をそれぞれモードコントロール命令，変位ベクトル，図形ベクトル系列と呼ぶこととする．

（1） モードコントロール命令

基本図形の種別，すなわち，直線，長方形，平行四辺形，四辺形のそれぞれに値 0, 1, 2, 3 を対応させ，この値を図形モードと呼び，基本図形 $F_i$ の図形モードを $M_i$ で表わすこととする．基本図形 $F_i$ のモードコントロール命令 $C_i$ は $F_{i-1}$ の図形モード $M_{i-1}$ に対する図形モード $M_i$ の差分を規定するための符号ブロックである．すなわち図形モードを +1, −1 するための符号をそれぞれ u, d で表わし，基本図形 $F_{i-1}$ に対する図形モードの差分 $m_i$ を

$$m_1=M_1, \quad m_i(i>1)=M_i-M_{i-1} \pmod 4 \tag{2.1}$$

とするとき，基本図形 $F_i$ のモードコントロール命令 $C_i$ を

$$C_i=\begin{cases} \mathrm{u}^{m_i}, & m_i=0,1,2 \text{ のとき} \\ \mathrm{d}, & m_i=3 \text{ のとき} \end{cases} \tag{2.2}$$

のように構成する．ここで $\mathrm{u}^m$ は $m$ 個の u の連結 $\underbrace{\mathrm{u}\cdots\mathrm{u}}_{m}$ を表わし，特に $\mathrm{u}^0$ は符号ブロック $C_i$ が空であることを表わす．

（2） 図形ベクトル系列

漢字を構成する各基本図形 $F_i$ に対し，表1に示すような2頂点（または端点）を始点，終点として定めることにより，$F_i$ の大きさ形状をその始点より端点へ向う 1〜3 個の順序対で表わされるベクトルの系列として表わすことができる．すなわち図形ベクトル系列 $G_i$ は表1に対応して

$$G_i=\begin{cases} (x_1, y_1) & M_i=0,1 \text{ のとき} \\ (x_1, y_1), (x_2, y_2) & M_i=2 \text{ のとき} \\ (x_1, y_1), (x_2, y_2), (x_3, y_3) & M_i=3 \text{ のとき} \end{cases} \tag{2.3}$$

の形で表わされる．

（3） 変位ベクトル

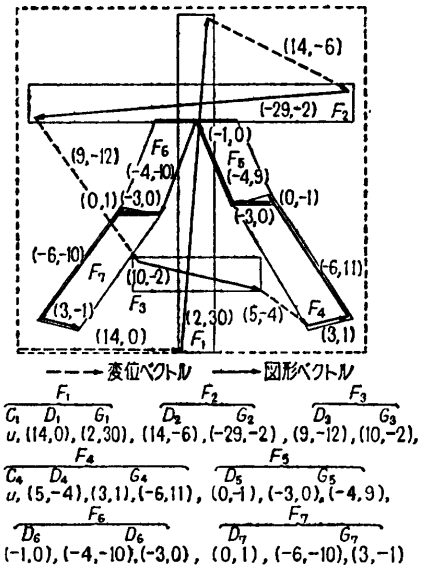



**図 2** ベクトル，モードコントロール命令系列の構成例
Fig. 2 An example of vector and mode control command sequence for a chinese character.

変位ベクトル $D_i$ は，基本図形 $F_{i-1}$ の終点を原点として，基本図形 $F_i$ の始点位置を示す1つの2次元ベクトル $(x_i, y_i)$ により構成され，基本図形 $F_i$ の $F_{i-1}$ に対する相対位置関係を規定する．

$$D_i=(x_i, y_i) \qquad (2.4)$$

であり，特に $D_1$ は漢字パターンを展開させるバッファ上の基準点からの $F_1$ の始点位置を表わすベクトルである．

以上述べた圧縮データ構成法により，一つの漢字パターンが与えられると**図2**にその例を示すように，圧縮データはモードコントロール命令 u, d と2次元ベクトルから成る系列として構成される．そして最終的な圧縮データは，それらのモードコントロール命令，2次元ベクトルの各々を次に述べる手法に従って符号化することにより得られることになる．

## 3. 符号化の手法

圧縮データを構成する2次元ベクトル，モードコントロール命令を符号化するに当っては，

（1） 圧縮データができるだけ短くなること，

（2） 圧縮データから元のパターンを得るパターンの復号が容易であること，

などを考慮する必要がある．（1）に関しては圧縮の対象となる漢字についての統計量，すなわち符号化対象項目の出現確率を知る必要があり，ここではランダムに選び出した約100個の乗車券発着駅名用漢字（表4にその一部を示す）について圧縮データを構成するのに必要な2次元ベクトルの出現頻度を実測した結果を**図3**に示す．

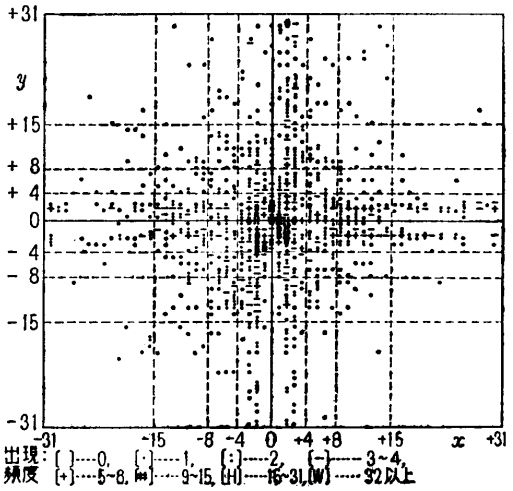



**図 3** ベクトル $(x, y)$ の出現頻度
Fig. 3 Statistical distribution of vector $(x, y)$.

この実測結果に示されるように，圧縮データを構成するベクトルは原点及び軸付近で高い出現確率を示し，$x$ 座標，$y$ 座標の絶対値がそれぞれ4以下であるベクトルは全体の約 45% に達する．従ってベクトルに対する符号の割当ては，まず原点附近のベクトル，次いで $x, y$ 軸近辺のベクトルの順に優先して長さの短い符号を可変長符号を用いて割当てることとする．

可変長符号の構成法にはファクシミリデータ圧縮に関するランレングスの符号化方法として各種の方法が知られており，後で本圧縮方式との比較対象としても引用する3ビットを1ブロックとする分割符号はそれらの符号の代表的なものである．これら既存の可変長符号を使って $x, y$ 座標のそれぞれを符号化することによっても，上述の優先順位に従った符号割当てが可能であるが，ここではそれらの方法より更に良好な結果を得ることができた可変長符号について述べることとする．

このベクトルを符号化するための可変長符号は，**図4**に示されるように4ビット単位で符号長が増加し，

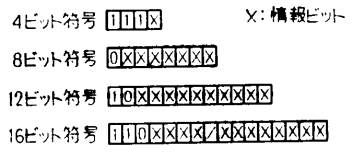



**図 4** 可変長符号の構成
Fig. 4 Format of variable length code.

表 2 ベクトル符号
Table 2 Code assignment for vectors.

| 符号長 | 符号の構成 | 対応するベクトル |
|---|---|---|
| 4ビット符号 | 1 1 1 0 | (0, 0) |
| 8ビット符号 | 0 0 0 $S_x$ $m_x$ | $(\pm m_x, 0)$ |
| | 0 0 1 $S_y$ $m_y$ | $(0, \pm m_y)$ |
| | 0 1 $S_x$ $m_x$ $S_y$ $m_y$ | $(\pm(m_x+1), \pm(m_y+1))$ |
| 12ビット符号 | 1 0 0 0 0 0 0 $S_x$ $m_x$ | $(\pm(m_x+16), 0)$ |
| | 1 0 0 1 0 0 0 $S_x$ $m_x$ $m_y$ | $(\pm(m_x+5), m_y+5)$ |
| | 1 0 0 $S_x$ $m_x$ $S_y$ $m_y$ $(m_x>4)$ | $(\pm m_x, \pm(m_y+1))$ |
| | 1 0 1 0 0 0 0 $S_y$ $m_y$ | $(0, \pm(m_y+16))$ |
| | 1 0 1 1 0 0 0 $S_x$ $m_x$ $m_y$ | $(\pm(m_x+5), -(m_y+5))$ |
| | 1 0 1 $S_y$ $m_y$ $S_x$ $m_x$ $(m_y>4)$ | $(\pm(m_x+1), \pm m_y)$ |
| 16ビット符号 | 1 1 0 $S_x$ $m_x$ $S_y$ $m_y$ | $(\pm m_x, \pm m_y)$ |

$S_x$, $S_y$ : 符号 (0……+, 1……−)
$m_x$, $m_y$ : 符号なしの2進数

上位 1〜3 ビットでその符号長を識別できるよう構成されている.

**表 2** は各ベクトルに対する具体的な符号の割当てを示すものである. この表において例えばベクトル (5, −7), (0, −10) はそれぞれ 12 ビット, 8 ビット符号により

1 0 1 1 1 0 0 0 0 0 0 1 1 0
0 0 1 1 1 1 0 1 0

と表わせる. (0, 0) ベクトルには最優先で 4 ビット符号が割当てられているのを始めとして, 出現頻度の高いベクトルには優先的に長さの短い符号が割当てられている.

さて, このベクトル符号化方式においては, 各符号の符号長は 4, 8, 12, 16 のいずれかであり, この符号長によりすべての符号を組分けし, それぞれの符号の組に対応するベクトルの組を図示するならば**図 5** のようになる.

表 3 モードコントロール命令符号
Table 3 Code assignment for mode control command.

| 符号長 | 符号の構成 | 対応する図形モードコントロール命令 |
|---|---|---|
| 4ビット符号 | 1 1 1 1 | u |
| 8ビット符号 | 0 0 0 0 0 0 0 0 | d |

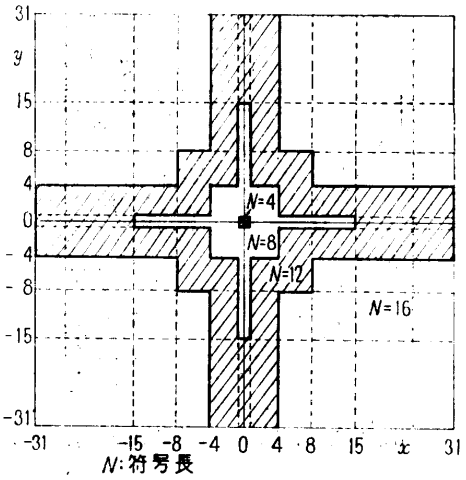



**図 5** 符号長別表現可能ベクトル
Fig. 5 Graphic representation of grouping of vectors by their code lengths.

なお, 図形モードコントロール命令 u, d に対しては, ベクトルの符号とは排他的に**表 3** のように符号化される.

## 4. 圧縮の効果

前記定期券発着駅名用漢字について, 本圧縮方式により漢字パターンの作成及び圧縮率の実測を行った. **表 4** には作成された漢字パターンの例, 圧縮率の実測結果, 及び比較のため実測した 1 ラインスキャニングによる分割符号化 2 値伝送方式[3], 2 ラインスキャニングによる分割符号化 2 値伝送方式[4], (以下それぞれ 1 ラインスキャニング方式, 2 ラインスキャニング方式と略す) による圧縮率も同時に示す.

1 ラインスキャニング方式, 2 ラインスキャニング方式は共にファクシミリにおける代表的データ圧縮方式である. 1 ラインスキャニング方式は漢字パターンを縦方向 1 ラインずつ読取り, 白黒の連の長さ (Run Lengh) を分割符号により符号化する方式である. すなわち**図 6** に示すように 3 ビットを 1 ブロックとして最初のビットを白, 黒 (白 : 0, 黒 : 1) の判定ビットに, 次の 2 ビットを連の長さを表わす 2 進号符号に割当て, 連の長さが 1 ブロックで表わせない場合順次ブロックを追加する.

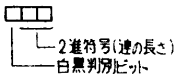



**図 6** 1 ラインスキャニング方式 1 ブロックの構成
Fig. 6 1 line scanning method code format.

表 4 圧縮率
Table 4 Chinese character pattern data compression ratio.

| 方式 / 文字 | 本方式 | 1ラインスキャニング方式 | 2ラインスキャニング方式 |
|---|---|---|---|
| 桜 | 1/2.8 | 1/1.5 | 1/1.6 |
| 矢 | 1/2.9 | 1/1.5 | 1/1.9 |
| 刀 | 1/5.9 | 1/2.2 | 1/2.7 |
| 林 | 1/4.0 | 1/1.9 | 1/2.3 |
| 倉 | 1/3.2 | 1/1.3 | 1/1.6 |
| 桑 | 1/2.3 | 1/1.1 | 1/1.2 |
| 江 | 1/5.5 | 1/1.7 | 1/2.2 |
| 美 | 1/3.5 | 1/1.2 | 1/1.5 |
| 天 | 1/4.3 | 1/1.4 | 1/1.7 |
| 小 | 1/6.6 | 1/2.5 | 1/2.9 |
| 弥 | 1/2.7 | 1/1.6 | 1/2.0 |
| 駒 | 1/2.2 | 1/1.4 | 1/1.8 |
| 全文字(約100文字)平均 | 1/3.2 | 1/1.5 | 1/1.9 |

2ラインスキャニング方式は漢字パターンの縦方向2ラインを同時に読取った際に生じる白白，白黒，黒白，黒黒の4種の連の長さを分割符号により符号化する方式である．連の種別は4種類あるため図7に示すように，その判定ビットとして2ビットを割当て，他の2ビットで連の長さを表わす．この4ビットで1ブロックを形成し，連の長さに応じてブロックを追加してゆく．

表4に示す圧縮率の実測結果では，本方式による圧縮効果は他の2方式に比べかなり良好であることを示している．なお，本圧縮方式における圧縮データは，基本図形をどのように組合せ漢字パターンをデザインするかということのほかに，それらの基本図形をどのような順序で取り上げる（符号化する）か，また各基本図形の始点，終点をどのように選ぶかによっていろいろ異なるものとなる．従ってこれらの点について最適化を計る必要があるが，実際にはパターン作成の際，通常の筆順に準ずる程度の“最適化”でも十分高い圧縮率を得ることができ，表4に示す圧縮率の実測結果，あるいは図2のベクトル出現頻度についても，特別の最適化処理は行っていない．

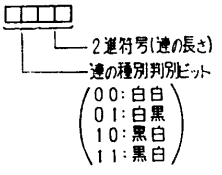



図 7 2ラインスキャニング方式 1ブロックの構成
Fig. 7 2 line scanning method code format.

## 5. パターンの復号

図8は圧縮データより元のパターンを得るパターン復号のフローである．プログラムは，M 6800 マイクロコンピュータ[5]を使って作成され，プログラムサイズはアセンブラ言語約800ステップ（約2kバイト），前記乗車券用発着駅名漢字の復号に要する時間は1文字あたり平均約60 msecであった．

本圧縮方式においては，パターン復号のアルゴリズムが文字の大きさに依存しないということも特徴の一つとして挙げることができる．

先に例として述べた1ラインスキャニング方式，2ラインスキャニング方式等は，圧縮の対象となるパターンが一定（長方形）の画面におさまっていることを

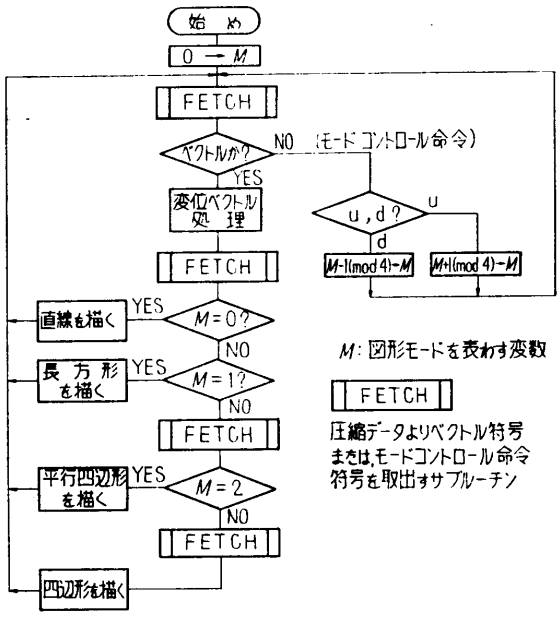



図 8 パターン復号のフロー
Fig. 8 Pattern decoding program flow.

前提にその画面を順次走査し，画面全体としてパターンを圧縮する方式である．従って実際のパターン復号に際しては，圧縮データのほか元の画面の大きさを知る必要があり，各文字について圧縮データのほかにこの大きさに関するデータを付け加え，そのデータに従って復号のフローを変える必要がある．

これに対して本圧縮，復号方式では，圧縮の対象となる基本図形を直接符号化し，復号化しているため，パターンは必ずしも一定の長方形の画面の中におさまっている必要はない．従って一文字単位の漢字のみならず例えば複数個の漢字から構成される単語，熟語等でも能率的に圧縮し復号することができる．このように本方式では乗車券等のように大小様々の文字を使用し，しかもある程度固定的に使用される単語，熟語で構成され，かつそれらを様々の形式にレイアウトすることが必要な場合には，特に他方式に比べ効率的なパターン圧縮，復号処理を行うことができるといって良い．

## 6. あとがき

現在わが国におけるコンピュータによる漢字処理の分野全体では，高速漢字プリンタを使った大規模な処理システムが大部分を占めているものと推測される．このような大規模システムに比べると，駅窓口における乗車券発行装置での漢字処理等においては一度に印字される漢字の量も少なく，そこでは高い処理能力よりも，プリンタ，メモリ等のコストの方が圧倒的に重視される．

本文はこのような小規模漢字処理システムにおける漢字パターンの圧縮方式に関する一つの提言であると同時に，そこで使用される漢字パターンそのものに対する提言でもある．

今後マイクロコンピュータあるいは分散処理技術の発展，普及に伴い，このような小規模漢字システムが増加してゆくものと思われるが，本文がこの分野においていささかでも役立てば幸いに思う次第である．

最後に本文に関して有益な助言を頂いた大阪大学通信工学科笠原正雄助教授，同電子工学科白川功助教授，日頃御指導頂いている同電子工学科尾崎弘教授及び近畿日本鉄道(株)中井実監査役，同技術研究所中川利雄所長に謝意を表します．

## 参考文献

1) 石田：画像伝送式乗車券発行システム，近鉄技報，Vol. 6, No. 1, pp. 44-53 (1974).
2) 尾上，岩下：計算機内における画像データ圧縮，情報処理，Vol. 18. No. 8, pp. 776-780 (1977).
3) 末広，松本：ファキシミリ信号帯域圧縮装置，日立評論，Vol. 52, No. 12, pp. 27-33 (1970).
4) 帯域 48 kHz FAX 信号を電話回線1チャンネルで伝送する，日経エレクトロニクス，1972-4-10, pp. 33-36 (1972).
5) Motorola Inc.: Microprocessor Applications Manual (1975).

(昭和53年3月3日受付)
(昭和53年6月12日採録)