

解説

最近のデータ圧縮技術†



長 峯 禎 三^{††} 杉 山 守^{††} 山 田 豊 通^{††}

1. はじめに

情報化社会の進展に伴い、大量の情報を経済的に蓄積・伝送したいというニーズが高まってきている。このニーズに応えるため、データ通信や画像通信においては、データを効率良く表現して、データ蓄積経費やデータ伝送経費の経済化を計るデータ圧縮技術が種々考案されて実用化されている。

データ通信の分野での適用例としては、①オンライン文献検索システムなど、大容量データベースを効率良くファイル記憶装置に蓄積することによりセンタ設備の経済化を計るもの¹⁾、②漢字出力装置など、大容量となる漢字・図形パターン情報を効率良くメモリに格納することにより装置の低価格化を計るもの⁴⁾、③広域分散処理など、データを効率良く伝送することにより、伝送時間・伝送経費の削減を計るもの³⁾、④大規模な Sparse Matrix を扱う数値計算など、集合データを効率良く表現して、主記憶メモリ使用量の削減とファイル・アクセス回数の削減によるデータ処理の経済化を計るもの^{2), 24), 31)}などがある。

画像通信の分野では、①ファクシミリなど静止画像を電話網・DDX網を用いて効率良く伝送して、伝送時間・伝送経費の削減を計るもの^{38)-40), 45)}、②テレビ会議、テレビ電話など動画を効率良く伝送して伝送路の経済化と画像品質の向上を計るもの⁴¹⁾⁻⁴³⁾などがある。

本稿では、データ通信および画像通信特にファクシミリ通信の分野で実用に供されている圧縮技術を紹介すると共に最近の圧縮技術の動向を示す。

2. データ圧縮の概要

2.1 データ圧縮とは

† Trends of Recent Data Compression Techniques by Teizo NAGAMINE, Mamoru SUGIYAMA (Data Processing Application System Section, ECL, NTT) and Toyomichi YAMADA (Visual Communication Equipment Section, ECL, NTT).

†† 日本電信電話公社横須賀電気通信研究所

データ圧縮とは、データに内在する冗長度を抑圧して、より短いデータ長で簡潔に表現することである。

これには、圧縮されたデータが元のデータに忠実に再現できる可逆変換の手法と、忠実に再現できない不可逆変換の手法とがある。データ通信やデジタル、ファクシミリ等の分野では忠実な再現が要求されるため可逆変換の手法が用いられるが、テレビ電話、テレビ会議等の分野では視覚で許される程度の原画の情報量の削減を行い、より高い圧縮率を得る不可逆変換の手法が用いられる。

2.2 効率的なデータ表現

(1) 原データの修正簡略化

テレビ信号等の伝送においては、実用上支障のない程度に、原データを修正簡略化し伝送情報量を削減する方法が行われる。これには、①発生頻度の少ない孤立点をノイズと見なして除去する、②動画の動きが大きくなれば、変動の大きな部分のみを伝送する、③画面、走査線、画面情報を間隔をあけて伝送し、受信側でその間の情報を補間再生する、などの方法がある。

(2) 空データのサブレス

ソース・プログラムやプリント・イメージのデータなど、フォーマットを整えるために付加されているブランク文字列、あるいは未収集値の多い統計データや '0' エントリの多い Sparse Matrix などの表現では、エントリの空表示を行うために埋められる空データをサブレスする方法が行われる。

(3) 重複データの排除

データ間の関係を整理し、データ間の関係に適したデータ構造(木あるいは網構造)を用いて表現することにより重複データを排除する。

(4) 差分表現

測定データ、ソート済みデータ、画像データなどの集合データでは、隣接したデータ間の相関が強いので、データ間の差分を逐次表現する方法が行われる。

(5) 予測

画像データなど、データ間の相関が強く、次にくる

表-1 データ通信で用いられるデータ圧縮手法

項番	着目点	データ圧縮手法	適用例
1	データ中の多くの空データに着目する方法	ビット・マップ	Sparse Matrix 統計データ etc.
2	データ中の同一パタンの連続に着目する方法	ランレングス符号化	ソース・プログラム プリント・イメージ・ファイル etc.
3	データ中のパタンの出現頻度の偏りに着目する方法	ボタン置換	在庫管理データ etc.
		コンパクト・コード化	文章データ (35~45% 削減) etc.
		語単位、可変長符号化	文章データ etc.
		ハフマン符号化	文章データ (40~60% 削減) 数値データ、ソース・プログラム (55~75% 削減) etc. オブジェクト・プログラム (35~45% 削減)
4	データ間の相関に着目する方法	差分表現	ソート済みデータ
		差分・可変長符号化	測定データ etc.
		逐次ハフマン符号化	文章データ (50~80% 削減)

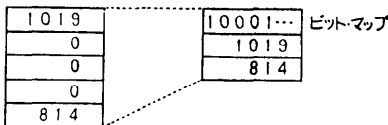


図-1 ビット・マップを用いたゼロ・データのサプレス

データがかなりの精度で予測できる場合には、予測処理を行い、予測誤りが生じるまでのランレングスを表現する方法が行われる。

3. データ通信分野の圧縮技術

3.1 各種の圧縮手法

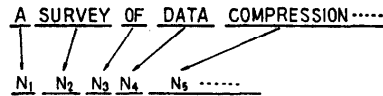
データ通信で用いられるデータ圧縮手法には表-1で示すようなものがある。以下に、これらの手法について簡単に説明する。

3.1.1 空データをサプレスする方法

ブランクやゼロで埋められる空エントリが多い集合データをコンパクトに表現するにはビット・マップ(Bit-Map)を用いる方法が有効である^{3),7),31)}。ビット・マップとは、集合データ中の空エントリに'0'Bを、データの存在する実エントリに'1'Bを割当てて、集合データ中の有効データの所在を表現するビット列である。この方法では図-1に示すように、ビット・マップをデータの先頭に付加して、その後有効データを詰合わせて表現する。

3.1.2 同一パタンの連続をランレングスで表現する方法

ソース・プログラムなど、文字列データ中に長いブランク文字列が多く出現する場合には、それらの文字列を文字列の長さ(ランレングス)で置換える方法が有効である^{3),7),21)}。この方法では、連続が始まること



(注) N_i で用いる可変長コード

1 0 0 0 0 0 0 0 *	1	1	パターン
1 x x x x x x x x	127	"	
0 x x x x x x x x 1 x x x x x x x x	16,384	"	
0 x x x x x x x x 0 x x x x x x x x		"	
1 x x x x x x x x	2,097,152	"	
1 0 0 0 0 0 0 0			は数値データを処理するための機能符号

図-2 単語単位のパタン置換の適用例

を示す特殊文字に続けてランレングスを表現する。

3.1.3 コンパクトなボタンに置換える方法

在庫管理データなど、限られた文字列のみが多く出現する場合には、それらの文字列をよりコンパクトなボタンに置換する方法が有効である^{3),7)}。

文章データでは、その構成要素である単語等の使用頻度の偏りに着目して、この単語、熟語あるいは単語の構成要素を単位として、使用頻度の高いものほど、よりコンパクトなボタンとなるように置換える方法が有効である^{3),18)}。この方法では、置換するボタンの種類が非常に多くなるため変換用辞書をもつこととなる。Heapsらの方法では、図-2に示すように、圧縮・復元処理を容易とするため、単語を使用頻度に従って1, 2, 3バイト表現の可変長コードに割当てている³⁾。

文章データをコンパクトに表現する別の方法として、Snyderman, Hunt, Jewellらによって提案されたコンパクト・コード(compact code)を用いる方法がある^{3),5)}。これは英文データを8ビット・コードで表現した場合(256文字の表現が可能)に生じる多くの未使用コードを用いて、使用頻度の高い文字列(2文字の列)を表現する方法である。Snyderman, Hunt

表-2 コンパクト・コード

マスタ文字 文字	16進	コンパイン グ文字	16進	ノンコンパ イング文字	16進	結合文字列 文字列	16進
U	58	U	00	J	15	UU	58
A	6D	A	01	K	16	UA	59
E	82	B	02	Q	17	UB	60
I	97	C	03	X	18		
O	AC						
N	C1						
T	D6	L	0A	0	35	AC	70
U	EB	M	0B	1	36		
		N	0B	1	37		
						0M	B7
		U	12	=	55		
		V	13	"	56		
		W	14	<	57	VW	FF

らの方法は、表-2に示すように、英文中での文字の使用状況に基づいてマスタ文字 (master character)、コンパイング文字 (combining character)、ノンコンパイング文字 (non-combining character) に文字をあらかじめ選別しておき、マスタ文字+コンパイング文字の順で文字が出現した場合に、この2つの文字の値を加算したコンパクト・コードで置換するものである。このコンパクト・コードの適用例を図-3に示す。Jwellの方法は、英文中での文字列の使用頻度の多いものから順次未使用コードを割当てていくものである。

3.1.4 シフト・コードを用いる方法

英文データなどの文章データは、通常8ビット・コードを用いて表現されているが、使用されている字種が少ない場合はシフト・コードを用いた5~6ビット・コードで表現する方法が有効である⁴⁾。このシフト・コードの適用例を図-4に示す。

3.1.5 パタンの使用頻度の偏りを利用する方法

データ中でのパタンの使用頻度の統計をあらかじめとっておき、使用頻度の高いパタンほど短いビット列を割当てる方法¹⁾で、先に述べた Heaps のコードやコンパクト・コードも、これに属するといえるが、代表的なものとしてはハフマン・コード (Huffman code) がある²⁾。ハフマン・コードは、情報源から発生する通報が互いに独立で、その発生頻度があらかじめ判っている場合に最も効率良く通報を表現できる符号 (最適符号) であるが、漢字データなど文字数の多いものに、そのまま適用すると符号器・復号器に要するメモリ量が多くなり実現が困難となる。メモリ量が多くなる原因は出現頻度の非常に少ない符号まで符号変換テーブル中に登録するためである。この欠点を除くため

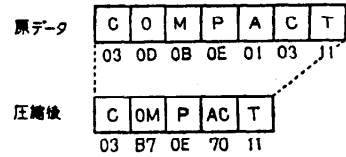
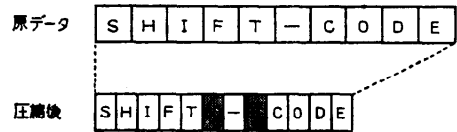
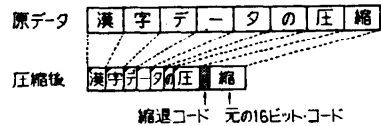


図-3 コンパクト・コードの適用例



(注) SI: シフト・イン・コード
SO: シフト・アウト・コード

図-4 シフト・コードの適用例



(注) ▣はハフマン・コードによる表現を示す

図-5 漢字データ用に改良したハフマン・コードの適用例

に、出現頻度の少ない文字群を1つの文字 (縮退文字と呼ぶ) とみなした縮退情報源に対してハフマン・コードを生成し、このコードと出現頻度の高い文字に対するコードのみ符号変換テーブルに登録する方法がある³⁴⁾。この場合、出現頻度の少ない文字は図-5に示すように縮退文字に対するハフマン・コード (縮退コード) に元の16ビット・コードを付加するため圧縮率は若干低下する。漢字データ用のハフマン・コードの例を表-3に示す。

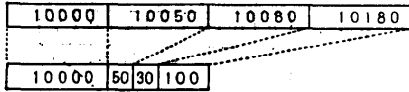
3.1.6 データ間の相関を利用する方法

ソート済みのデータや測定データなど、隣接データ間の相関が強い場合は、データ間の差分を表現する方法が有効である³⁾、⁴⁾。

表-3 漢字データ用ハフマン・コードの例

出現順序	文字	出現頻度 (%)	符号長	ハフマン・コード
1	■(注)	5.728	4	0000
2	の	4.956	4	0001
3	,	2.963	5	00100
4	に	2.563	5	00101
5	る	2.405	5	00110
	}	}	}	}
598	支	0.023	12	1111111111011
599	先	0.023	12	1111111111100
600	欠	0.023	12	1111111111101

(注) 縮退文字を示す。



(注) 差分表現は以下の可変長コードを用いる。
 $-2^2+1 \sim 2^2-1 = 0 S \times \times \times \times \times \times$
 $-2^{11}+1 \sim 2^8$
 $2^8 \sim 2^{11}-1$ } = $1 S \times \times \times \times \times \times 0 \times \times \times \times \times \times \times$
 $-2^{11}+1 \sim -2^{11}$
 $2^{11} \sim 2^{11}-1$ } = $1 S \times \times \times \times \times \times 1 \times \times \times \times \times \times \times$
 $0 \times \times \times \times \times \times \times$
 S = 符号ビット

図-6 数値データ例に対する差分表現の適用例

数値データに関しては、差分値の出現頻度に従って可変長コードを用いた図-6に示すような表現が³⁵⁾、ソート済みの文字列に対しては、図-7に示すような表現が用いられる⁴⁾。

文章データに対して文字間の相関を利用する方法としては、データ中の文字の出現頻度が直前の文字列に依存する特徴を利用する方法がある^{30), 33)}。この方法は、文章データを m 重マルコフ情報源 S^m から出力された文字列とみなして、 m 文字を記憶した状態における文字の条件付き出現確率に対してハフマン・コードを適用するものである。

3.2 圧縮技術の動向

データ通信においては、大容量データベースを取扱う情報検索の分野で多くの研究が進められている。この分野で扱う文章データを最も効率良く符号化できるのは Huffman の最適符号化法²⁾であり、英、仏、独などの欧文データでは約4ビット/文字、日本語のデータでは約8~9ビット/文字で表現できることは良く知られている。しかし、この符号化は可変長のビット列を扱うことによりアルゴリズムが複雑で処理時間が多くかかること、符号変換テーブルのメモリ使用量が多くなることなど、そのままでは実用化しにくい面がある。

アルゴリズムの簡略化を計るため、Snyderman, Hunt, Jewell らは、コード・テーブル中の未使用コードに出現頻度の高い文字列を割当ててコンパクト・コード³⁾を開発し、英文データで約5ビット/文字の圧縮効果を実現した。この方法は、固定長コードを扱うだけで良いため復号処理が高速に行える反面、符号化処理は高速に行えない欠点がある。

高い圧縮効果を追求する手法として文字間の相関を利用する研究が進められている。これには、文章の構成要素を単位として、よりコンパクトなパターンに置換する方法と、文字の条件付き出現頻度をもとにハフマ

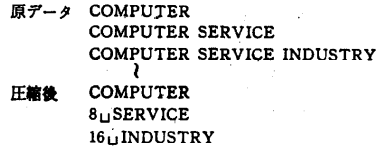


図-7 ソート済み文字列データに対する差分表現の適用例

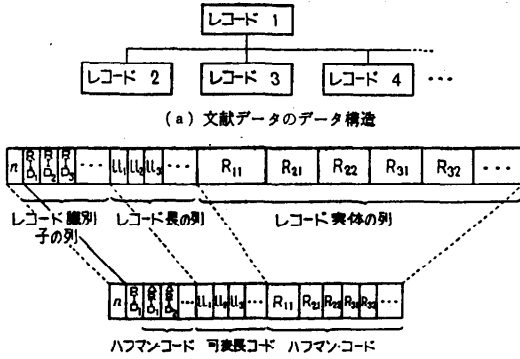
ン・コードを適用する手法が考えられている。パターン置換によるものには、単語等の自然な単位で置換する Heaps らの方法や、 N 個の文字列を単位とする N -gram 圧縮法^{8), 20)}があり、より高い圧縮率が得られるようなパタンの切り出し方の研究が進められている¹²⁾。これらの手法は圧縮単位の状態数が非常に多くなり変換辞書が必要になること、あるいは変換処理時間を多く要するなどの欠点があり、ハードウェア化による高速化も検討されている²²⁾。文字の条件付き出現頻度をもとにハフマン・コードを適用する方法では、多くの文字を記憶しておけばおくれ、出現頻度の偏りが大きくなり、極限状態では英文で1.7ビット/文字、仏文で3ビット/文字、独文で1.1ビット/文字と大幅な圧縮効果が得られることが判っているが、記憶する状態数が極めて大きくなるため実用化上は1文字を記憶する場合(英文で約3.4ビット/文字)が行える程度である³³⁾。

このように処理時間、メモリ使用量など実用上の問題点の解決を計るための手法が研究されてはいるが、ソフトウェア技術だけでは処理時間向上に限度があるため、ファームウェア化、ハードウェア化による高速化の研究がなされており、ファームウェア化で約1/30程度、ハードウェア化で約1/1,000程度の処理時間短縮が実現できるという研究報告もなされている^{4), 23)}。

今後も種々の応用分野で²³⁾⁻²⁹⁾、取扱うデータの特徴を巧みに捉えて種々の基本技術を効果的に組合わせたデータ圧縮手法が開発され、ファームウェアあるいはハードウェアとして実装されていくものと思われる。

3.3 圧縮技術の応用例

オンライン文献検索システムの応用例について述べる。オンライン文献検索は、①蓄積するデータ量が膨大で、かつデータの寿命が長い、②利用者が地理的に広域に分散する、などの特徴があり、データの圧縮格納、圧縮伝送によるメリットの大きい分野の1つである。ここでは、文献データのデータベース中での圧縮格納の方法について述べる。



(注) レコード識別子 (RID) の列は差分表現した後にハフマン・コードを適用する。
 レコード長 (l_{ij}) は可変長コードを適用する。
 レコード (R_{ij}) はハフマン・コードを適用する。

図-8 文献データのデータ構造と圧縮手法の適用方法

文献データは、図-8に示すように複数のレコードからなる木構造データとして表現でき、これらのレコードは、同時にアクセスされる頻度が高いため、記憶媒体上にひとまとめで格納する方法が一般に行われる。これを実レコードと呼ぶと、実レコードは各レコードに関するレコード識別子 (RID)、レコード長、レコード実体の情報の集まりであり、これを効率良く表現するには、以下に示すように複数のデータ圧縮手法を組合わせて行う方法が有効となる。

- ① レコードをレコード識別子の値によりソートし、レコード識別子を差分表現する (分類 & 差分処理)
- ② レコード識別子、レコード長、レコード実体を分離して各々をひとまとめで表現し、各々の統計的特徴をとらえた符号化を行う (分離 & 統計的符号化処理)。

数 値	可変長コード
1~16	1××××
17~128	1××××0×××
129~1,024	1××××1×××0×××

図-9 レコード長の表現に用いる可変長コード

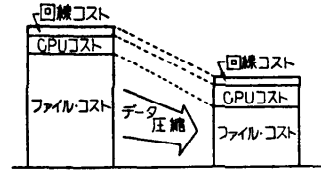


図-10 オンライン文献検索におけるデータ圧縮効果

文献 35) では、レコード識別子の差分をハフマン・コードで、レコード長を図-9で示す可変長コードで、レコード実体は各レコードに適したハフマン・コードを用いて表現し、約5割の圧縮効果を得ている。また代表的な文献検索モデルで行われた実測評価では、データ量が40~50%削減でき、圧縮・復元処理によるオーバーヘッドは10%以内で納まり、図-10に示すようにシステム全体の経費が約40%削減できる結果を得ている。

4. ファクシミリ通信分野の圧縮技術

4.1 各種の圧縮手法

ファクシミリ通信で用いられているデータ圧縮手法には、表-4で示すようなものがある。以下に、これらの手法について簡単に説明する。

4.1.1 一走査線間内の画素間の相間のみを利用する方法 (一次元符号化方式)

デジタル処理によって白黒の2値を持つ格子状の画素に細分化された画像データでは、白および黒画素

表-4 ファクシミリ通信のデータ圧縮手法

項番	着 目 点	主なデータ圧縮手法	利用している基本的な圧縮手法					
			ランレングス符号化	可変長符号化	差 分	予 測	ブロック化	二次元相関
1	一走査線内の画素間の相関に着目する方法	一次元ランレングス符号化	○	○				
2	複数の限定された一走査線間の相関に着目する方法	二走査線一括モードランレングス符号化	○	○			○	○
		変化点ブロック・パターン符号化	○	○	(画素レベル)		○	○
3	隣接する一走査線間の相関に着目する方法	二次元予測符号化	○	○		○		○
		変化点相対アドレス符号化 (RAC)	○	○	(画素間距離)			○
		境界差分符号化 (EDIC)	○	○	(画素間距離)			○
		相対画素位置測定符号化 (READ)	○	○	(画素間距離)			○

ラン長	白ラン用符号	黒ラン用符号
0	00110101	0000110111
1	000111	010
2	0111	11
3	1000	10
4	1011	011
5	1100	0011
63	00110100	000001100111

(a) ターミネート符号 (terminating code)

ラン長	白ラン用符号	黒ラン用符号
64	11011	0000001111
128	10010	000011001000
192	010111	000011001001
256	0110111	000001011011
1,728	010011011	0000001100101
EOL	0000000000001	0000000000001

(b) メイク・アップ符号 (make up code)

(注) 0~63 までのランレングスはターミネート符号のみで表現する。
EOL 符号は各走査線の終了を示す符号である。

図-11 モディファイド・ハフマン符号 (modified Huffman code)

が連続して並ぶことが多い。この特徴を利用して走査線方向の白および黒画素の連続をランレングスで置換える一次元ランレングス符号化がある³⁶⁾。このランレングスをコンパクトに表現する標準符号 (CCITT 勧告, G3 機標準符号化方式) としてモディファイド・ハフマン符号 (modified Huffman code) がある³⁷⁾。この符号はファクシミリで用いる符号器・復号器の構成を容易とするためハフマン・コードの適用方法を工夫したもので、図-11 に示すように、一走査線の画素数 1,728 に対する 0~1,728 のランレングスを初めから 64 個ごとにグループ分けをして、各ランレングスをグループを示す符号 (make up code) とグループ内通番を示す符号 (terminating code) との組合せで表現する。ここで、出現頻度の多い 0~63 のランレングスはターミネート符号のみで表わし、出現頻度の低い 64 以上のランレングスはメイク・アップ符号+ターミネート符号で表現する。また、白画素と黒画素とでは、ランレングスの統計量が大幅に異なるため、白ランと黒ランにそれぞれ独立な符号を用意し、白ランと黒ランが交互に出現することを利用して走査線の最初は白ランと約束しておき、白ランと黒ランの符号を交互に配列する方法を用いている。

4.1.2 複数の限定された走査線間の相関を一括利用する方法 (二次元一括符号化方式)

画素間の二次元相関を利用する方法で、二走査線上の上下の 2 画素を 1 ブロックとし、ブロックのモードとランレングスを符号化する二走査線一括モード・ランレングス符号化³⁶⁾や画素を差分表現 (変化点のみ 1

で表現) した後、2x4 のブロックに分割し、変化点を全く含まないブロックはランレングス符号化し、その他のブロックは出現頻度に対応したハフマン符号化を行う変化点ブロック・パタン符号化³⁸⁾などがある。

4.1.3 隣接する走査線間の相関を逐次利用する方法 (二次元逐次符号化方式)

画素間の二次元相関をより効率良く利用する方法で、二次元予測処理を利用する方式³⁶⁾や境界の形状の変化を符号化した境界差分符号化方式 (EDIC)³⁹⁾、変化点相対アドレス符号化方式 (RAC)³⁹⁾、EDIC と RAC の両方式の特長を生かした相対画素位置選定符号化方式 (READ)⁴⁰⁾、READ 方式に若干の変更を加えた後、CCITT にて G3 機の二次元符号化方式の標準方式となったモディファイド READ 方式などがある。

二次元予測処理は、図-12 に示すように画素 x を近

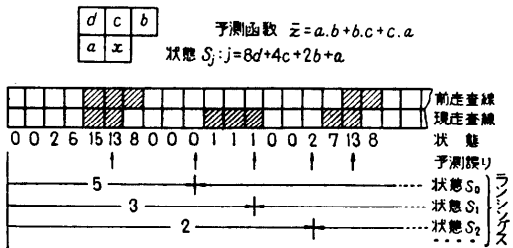
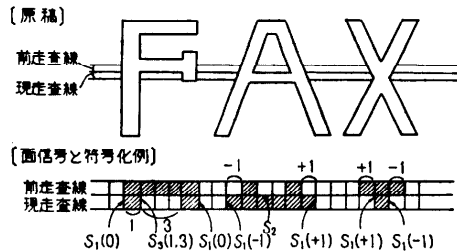


図-12 二次元予測符号化の原理



状態	符号	
$S_1(d)$	$d=0$	0
	$d=+1$	1 0 0
	$d=-1$	1 0 1
	$d \geq +2$	1 1 0 0 D ($ d $)
	$d \leq -2$	1 1 0 1 D ($ d $)
S_2	1 1 1 0	
$S_3 (l_1, l_2)$	1 1 1 1 L (l_1) L (l_2)	

・|| は絶対値記号
D($|d|$) には 0 0 1 を割当てて
 $|d|-2$
L(l_1), L(l_2) にはモディファイドハフマン符号を割当てて

図-13 境界差分符号化の原理

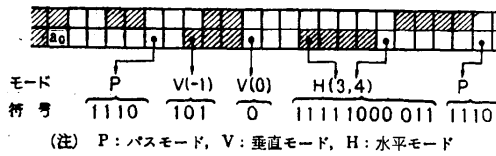


図-14 相対画素位置選定 (READ) 符号化の例

傍画素 $a \sim d$ の値を用いて予測し、予測誤りが発生するまでのランレングスを符号化するものである³⁶⁾。

境界差分符号化方式では、図-13 に示すように境界のずれを①境界が前走査線から現走査線に続いている状態、②境界が前走査線上で上側に折返している状態、③境界が現走査線上で下側に折返している状態の3つの状態に分け、これらの状態の出現頻度に応じた符号化を行う。

READ 方式では、図-14 に示すように境界の形状をパス・モード、水平モード、垂直モードに分けて符号化を行う。モディファイド READ 方式は、READ 方式に対して、垂直モードを変化点のずれ3ビット以下の時のみ使用するように変更して簡略化するとともに、一次元のモディファイド・ハフマン符号と同じ周期符号を使えるようにしたものである。図-15 に一次元、二次元一括、二次元逐次符号化方式の電送時間の比較を示す。

4.2 圧縮技術の動向

ファクシミリにおけるデータ圧縮技術は、昭和48年の電電公社の電話網の開放に伴うファクシミリの普及に並行して、デジタル・ファクシミリの重要な技術の1つとして発展してきた。今日までに、走査線方向の画素の相関に着目した一次元符号化方式の開発とともに、とくに日本では世界に先がけて走査線間の画素の相関をも着目した圧縮効果の優れた二次元符号化方式の開発が進められ種々の方式が実用に供されてきた³⁶⁾。

デジタル・ファクシミリの世界的普及に先立って、符号化方式の標準化の必要性が高まり、CCITT では符号化アルゴリズムが簡単で符号誤りに強く、かなりの冗長度抑圧効果が得られる一次元ランレングス符号化方式を標準方式として選定した。なお、ランレングス符号としてはモディファイド・ハフマン符号 (MH) が使われることから本符号化方式は一般には MH 方式と呼ばれる。その後、高圧縮率方式の標準化の要望に応じて、二次元符号化方式をオプションとして規定することとなり、昨年 (1979 年) 日本統一案として提案された READ 方式を MH 方式の拡張とし

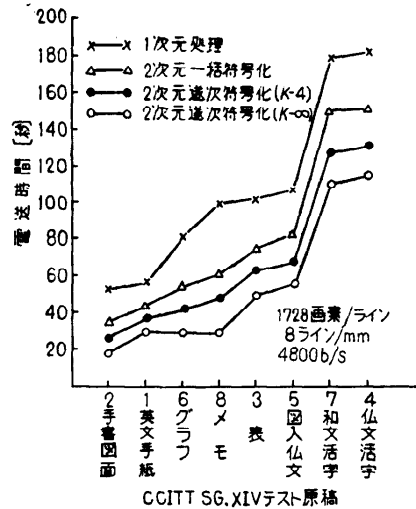


図-15 2次元逐次処理方式と他方式との電送時間比較
 (注) K=4 は4ラインごとに一次元ランレングス符号化を行うこととなります

図-15 2次元逐次処理方式と他方式との電送時間比較

て利用できるように一部変更を加えたモディファイド READ (MR) 方式が二次元符号化方式の標準方式として選定されるに至っている。なお、日本では二次元符号化方式+9,600 bit/s MODEM で1分機に対して30秒機、20秒機として商品化されているものが多い。

CCITT において一次元符号化方式と二次元符号化方式の標準化作業が終了したことにより、一般的な2値ファクシミリ信号に対するデータ圧縮方式の研究実用化および標準化は一段落した。今後は適用領域を限定した疑似中間調、中間調やカラーファクシミリ信号のデータ圧縮方式あるいは2値の場合でも細め処理や、よりパタン認識に近い処理等、不可逆変換手法を利用したデータ圧縮方式の研究・実用化が進められていくものと思われる。

5. おわりに

データ圧縮技術はファイル記憶装置や伝送路の経済化を計る上で有効な技術である。この技術は、ハードウェア技術の進歩により、ファイル・コストや伝送コストが今後とも急速に下がるとしても、蓄積/伝送される情報量が飛躍的に伸び続けている状況が続く限り、今後ともシステム経費の経済化を計る重要な技術として生き続けて行くであろう。

参考文献

1) Shannon, C. E.: A Mathematical Theory of Communication, Bell Syst. Tech. J. (1948).

- 2) Huffman, D. A.: A Method for the Construction of Minimum Redundancy Codes, Proc. IRE, 40, p. 1098 (1952).
- 3) Data-Compression-A Comparison of Methods, NBS Special Publication 500-12 (1977).
- 4) Martin, J.: Computer Data-Base Organization-Data Compaction, Prentice-Hall (1975).
- 5) Computer Programming and Management-Data Base Compression, Auerbach (1975).
- 6) Shuegraf, E. J.: A Survey of Data Compression Methods for Non-Numeric Records, Canadian Journal of Information Science (1977).
- 7) Stephen S. Ruth and Paul J. Kreutzer: Data Compression for Large Business Files, Datamation (1972. 9).
- 8) Bruce Hahn: A New Technique for Compression and Storage of Data, Communications of the ACM, Vol. 17, No. 8 (1974).
- 9) Ling, H. and Palermo, F. P.: Block-Oriented Information Compression, Information Compression (1975. 3).
- 10) Dance, D. L. and Pooch, U. W.: An Adaptive On-Line Data Compression System, The Computer Journal, Vol. 19, No. 3 (1975).
- 11) Weiss, S. F. and Vernon, R. L.: A Word-Based Compression Technique for Text Files, Journal of Library Automation (1978. 6).
- 12) Sheugraf, E. J. and Heaps, H. S.: Selection of Equipfrequent Word Fragments for Information Retrieval, Information Storage and Retrieval, Vol. 9 (1973).
- 13) Shuegraf, E. J. and Heaps, H. S.: A Comparison of Algorithms for Data Base Compression by Use of Fragments as Language Elements Information Storage and Retrieval, Vol. 10 (1974).
- 14) Robert A. Wagner: Common Phrases and Minimum-Space Text Storage, Communications of the ACM, Vol. 16, No. 3 (1973).
- 15) Radhakrishnan, T.: Selection of Prefix and Postfix Word Fragments for Data Compression, Information Processing and Management, Vol. 14 (1978).
- 16) Mayne, A. and James, E. B.: Information Compression by Factorising Common Strings, The Computer Journal, Vol. 18, No. 2 (1973).
- 17) Clare, A. C. Cook, E. M. and Lynch, M. F.: The Identification of Variable Length, Equipfrequent Character Strings in a Natural Language Data Base, The Computer Journal, Vol. 15, No. 3 (1971).
- 18) Rubin, F.: Experiments in Text File Compression, Communications of the ACM, Vol. 19, No. 11 (1976).
- 19) Thiel, L. H. and Heaps, H. S.: Program Design for Retrospective Searches on Large Data Bases, Information Storage and Retrieval, Vol. 8 (1972).
- 20) Lynch, M. F. Petrie, J. H. and Snell, M. J.: Analysis of the Microstructure of Titles in the Inspec Data Base, Information Storage and Retrieval, Vol. 9 (1973).
- 21) Lynch, M. F.: Compression of Bibliographic Files Using an Adaptation of Run-Length Coding, Information Storage and Retrieval, Vol. 9 (1973).
- 22) Lea, R. M.: Text Compression with an Associative Parallel Processor, The Computer Journal, Vol. 21, No. 1 (1976).
- 23) Johnston, H. C. and Rhoare, C. A.: Matrix Reduction-an Efficient Method, Communications of the ACM, Vol. 1, No. 3 (1975).
- 24) Tarjan, R. E. and Yao, A. C. C.: Storing a Sparse Table, Communications of the ACM, Vol. 22, No. 11 (1979).
- 25) Blasgen, M. W. Casey, R. G. and Eswaran, K. P.: An Encoding Method for Multifield Sorting and Indexing, Communication of the ACM, Vol. 20, No. 11 (1977).
- 26) Davisson, L. D.: An Approximate Theory of Prediction for Data Compression, IEEE Transactions on Information Theory, Vol. IT-13, No. 2 (1967).
- 27) Chen, T. C. and Ho, I. T.: Storage Efficient Representation of Decimal Data, Communications of the ACM, Vol. 18, No. 1 (1975).
- 28) Marey, K.: Compressed Tries, Communications of the ACM, Vol. 19, No. 7 (1976).
- 29) Even, S. and Rodeh, M.: Economical Encoding of Commas Between Strings, Communications of the ACM, Vol. 21, No. 4 (1978).
- 30) 浅倉: 英文字列中のマルコフ性の起源とそれが及ぶ範囲について, 電子通信学会論文誌, Vol. 12, J61-D, No. 12 (1978).
- 31) 大附, 川北: スパース行列処理技法, 情報処理, Vol. 17, No. 1~3 (1976).
- 32) 尾上, 岩下: 計算機内における画像のデータ圧縮, 情報処理, Vol. 18, No. 8 (1977).
- 33) 杉山, 中村: データ圧縮アルゴリズムの一方, 情処全国大会 (1978).
- 34) 杉山, 長峯他: 日本語データの効率的な格納法, 情処全国大会 (1979).
- 35) 長峯, 北村, 杉山: 情報検索プログラム —DO-RIS-2—, 情処第17回データベース管理システム研究会資料 (1980).
- 36) 沢田, 山田: 高速ファクシミリの技術動向, 画像電子学会誌, Vol. 6, No. 3 (1977).
- 37) 安田: モディファイド・ハフマン符号, 画像電子学会誌, Vol. 6, No. 3 (1977).
- 38) 山田, 中川, 結城他: 高速ファクシミリの冗長

- 度抑圧符号化方式, 電気通信研究所, 研究実用化報告, Vol. 28, No. 5 (1979).
- 39) 山崎, 若原: 国際通信における高速デジタル・ファクシミリ, 画像電子学会誌, Vol. 6, No. 3 (1977).
- 40) 結城, 山岡, 山崎, 若原: ファクシミリ信号の符号化方式, 画像電子学会予稿, (78-06-4).
- 41) 大川原, 金谷, 松岡: フレーム間符号化方式の概要, 電気通信研究所, 研究実用化報告, Vol. 26, No. 1 (1977).
- 42) 安田, 黒田, 川西: フレーム間符号化方式の選択, 電気通信研究所, 研究実用化報告, Vol. 26, No. 1 (1977).
- 43) 沢田, 小寺: NTSC カラーテレビ信号 DPCM 符号化方式の設計, 電気通信研究所, 研究実用化報告.
- 44) 佐橋, 堀口: 漢字パタンデータの伝送圧縮に関する一検討, 信学技報, IE 79-5 (1979).
- 45) 岸本, 三ツ矢他: 静止画像のブロック符号化法, 信学会論文誌, Vol. 62-B, No. 1 (1979).
(昭和55年3月4日受付)
-