文書画像に対する新しい圧縮技法の提案

3M－4

村本 典子　田中 譲
北海道大学 知識メディアラボラトリー

1 はじめに

情報が大量に行き交う現代社会において，それらを管理しさらに保存するということは重要なことである．それらの点において大変有用である電子化があらゆるもので行われている．最近では，インターネットの普及により電子化した情報を誰でも簡単に複製・配布でき，ますますその重要性が認識されている．

電子化された文書に対して，それらの管理・保存を目的としたトランスメディアシステムが研究・開発されている．トランスメディアは画像文書に対して，文字認識を行わずに画像のままで文字列の検索や文章の編集などを行う．文字列をある特徴量を用いて擬似コードに変換し，それらのマッチングにより同じ画像であると判断し検索などの処理が行われる．

本稿ではそのトランスメディアを用いて，画像文字を辞書に登録することで行われる圧縮技法を提案する．ここでの手法は従来の画像圧縮のように忠実に画像を再現するものではない．文書であることを前提として文字領域のみが保存される．さらにその文書中に複数存在する文字列は，復元の際に全て同じ画像に置き換わる．このシステムでは2値のビットマップ画像を用いている．

2 辞書登録

文書を圧縮する際には，まずその文書中から同じ文字，もしくは文字列を検索することから始める．そのマッチング法は3節で述べる．このときそれらの文字列は辞書に登録され，そのindex により文書全体は符号化，つまり圧縮される．圧縮率を上げるためには，辞書のindexは少なく文字列はより長いものが符号化されるほうがよい．言い換えれば，長い単語で同一化されるものが多いほど圧縮率は高くなる．ここではそれらをふまえた上で，圧縮作業を効率的に行うのにテキスト文書のための辞書登録の技術であるLZW [1] の手法を取り入れることにする．以下からそのLZWを用いたトランスメディアでの辞書登録について述べる．

このシステムでは画像の文書を扱うためにLZW本来のものとは違い2つの辞書を使うことにする．一方は従来のLZWでの辞書と同じ役割を果たすものであるが，他方は文字画像を登録した辞書である．ここで便宜上前者を文字列辞書，後者を文字画像辞書と呼ぶことにする．つまり文字列辞書には画像のテンプレート自体は登録せず，文字画像辞書のindexを参照する形で文字列を表現する．文字列辞書のみであると重複して登録される文字が多数存在するため，その重複を避けるのが目的である．文字画像辞書は文字のテンプレートを保存するために作成され，これにより文字画像は1つに付き1つだけ保存される．2つの辞書は文書の符号化と同時に作成，登録していく．

例として，文章の一部である以下の文字列を符号化することを考える．

［これまでに･･･これまでに･･･］

初期状態として2つの辞書には何も登録されていない．最初の文字は こ である．これをまず文字画像辞書に登録し，そのindex 01を文字列辞書に登録する．そして こ は文字列辞書のindex 01で符号化される．さらに次の文字との連結文字列 これ を文字列辞書に登録する．ところが文字 れ は文字画像辞書に登録されていない．そこでそれを文字画像辞書に登録し，そのindex 02を文字列辞書に登録する．連結文字列 これ は，文字列辞書に文字画像辞書のindexより01-02と登録する．

以上のような処理を繰り返すことにより5番目の文字 に までで文字画像辞書はindex 05まで，文字列辞書はindex 09まで登録されており，出力される符号は［01 02 04 06 08］となる．

表 1: 文字列辞書

index	string	index	string	index	string
01	01	06	04	51	01-02-03
02	02	07	03-04	52	03-04-05
03	01-02	08	05	⋮	⋮
04	03	09	04-05		
05	02-03	⋮	⋮		

表 2: 文字画像辞書

index	image	index	image
01	こ	04	で
02	れ	05	に
03	ま	⋮	⋮

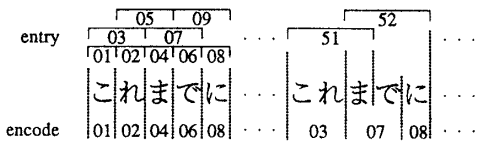

図 1: LZWの文書画像への適用

さらに先で先頭文字が こ であるところに来たとする．こ から連なる文字列のうち，辞書に登録されている最も長いものを使って符号化する．この例では これ

A New Approach to Document Image Compression
Noriko Muramoto, Yuzuru Tanaka
Meme Media Laboratory, Hokkaido University

を示すindex 03が最も長い．よって03で符号化し，次の文字との連結文字列 これま つまり01-02-03を文字列辞書に登録する．

2つの辞書は表1，2のようになり，出力される符号は[01 02 04 06 08 ··· 03 07 08 ···]である．

3 文書文字列と登録文字列のマッチング

辞書登録をする際に，文書中の文字列と登録文字列のマッチングを行う必要がある．マッチングは1文字ごとに密度比による特徴量から生成された擬似コードを用いる[2]．この擬似コードは同じ文字列であっても違うコードを生成することがある．そのためマッチングする擬似コード同士のミスマッチ数が，文字数に比例したあるしきい値を越えていなければマッチしたことにする．その値をビット捨て値と呼ぶ．実験より日本語1文字の擬似コードは36bit，ビット捨て値は8bitが最適である．

2節で述べた2つの辞書には，その擬似コードを全て登録する．仮にindexが文書中4ヶ所で用いられたとすれば，最初にそのindexを生成したときのものと併せて擬似コードは5つ登録される．これら登録された複数のコードのうち，どれか1つでも条件を満たしていればマッチする．登録されたコードの数が多いほど条件は緩くなり，ヒットするコードの範囲は広がる．

擬似コードがマッチしても実際の文字列画像が異なる場合を考慮して，さらにテンプレートマッチングを行う．表2の文字画像辞書に登録している文字画像は，そのindexにマッチした全ての文字の平均画像である．ここでの元画像は2値のビットマップであるが，この登録画像はグレースケールである．

例に沿って実際のマッチング法と辞書登録法を述べる．文書中の まで という画像を符号化するとき，辞書が図2のような状態であるとする．その文字列 まで と文字列辞書のindex 30と50で擬似コード及びテンプレートマッチングが適合したとき，さらにそのindex 30と50でテンプレートマッチングを行う．適合したときは片方にもう一方の情報を付加，つまりpseudo codeの付加とimageを両者の平均画像とする．加えた方のindexは消去する．適合率が高いのが50の方であるとすれば，30が消去される．そしてその50に現在の文字列 まで を登録する．30と50が適合しなかったら，50の方に文字列を登録するのみで，30は変わらない．どちらにしても，今検証している文書中の画像 まで は50で符号化される．

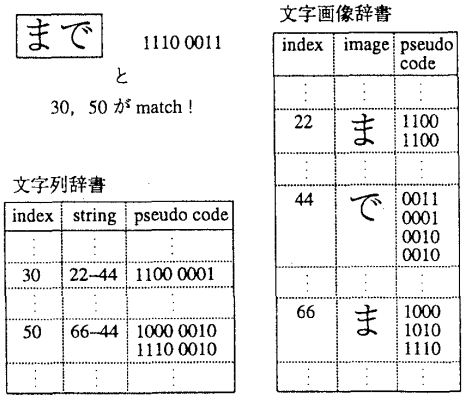

図 2: マッチング法の例 (1)

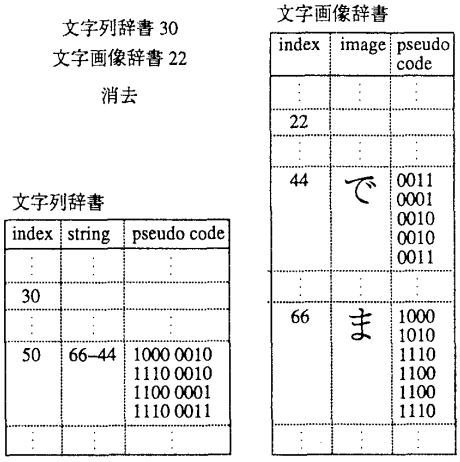

図 3: マッチング法の例 (2)

4 更なる圧縮

符号化と同時に作成した文字列辞書と文字画像辞書は，復元するときには不要な情報が含まれている．そこで符号化された文書とともに保存する辞書を新たに作成する．文字列辞書において一度も参照されていない，つまり符号化されていないindexは必要ない．また各々の要素で必要なものは文字列辞書はindexとstring，文字画像辞書はindexとimageである．この最終のimageは登録されていたグレースケールの平均画像ではなく，そこに登録されたいずれかの元画像文字を用いる．

5 おわりに

辞書登録によって行われる文書画像に対する圧縮技法について示した．この手法により元文書のフォントを失わない圧縮が可能となる．現在実験段階のため，圧縮率等の詳しい数値は得られていない．

参考文献

[1] T.A.Welch, "A Technique for High-Performance Data Compression", *IEEE Computer*. 8-19, June 1984.

[2] 遊佐実, 田中譲, "トランスメディアシステムの日本語への拡張", 情報処理学会 第49回全国大会, 1994.