1I2-2

DICOM形式医療用画像のための高速・高圧縮規格の提案

A High Speed and High Compression Standard Practice for a DICOM Medical Image

六井　淳[*1]　長谷川　栄司[*2]　永岡　和子[*2]　相田　範夫[*2]
Jun Rokui　Eiji Hasegawa　Kazuko Nagaoka　Norio Aida

[*1]島根大学総合理工学研究科数理・情報システム学専攻
dept of Mathematics and Computer Science Interdisciplinary Faculty of Science and Engineering, Shimane University

[*2]株式会社テクノプロジェクト
TechnoProject Ltd.

In this study, we achieved high compression performance by combining medical images of DICOM standard practice using various techniques. We combined five techniques (run length encoding, Huffman encoding, Move To Front (MTF), Burrows–Wheeler Transform (BWT), LZSS (Lempel-Ziv-Storer-Szymanski) and Dynamic Range Coder (DRC)). High compression performance is obtained not in original compressed format of each company, which is in the conventional DICOM format but in the combination of a general-purpose compression method. We verified the validity of the proposed technique experimentally.

1. はじめに

DICOM(Digital Imaging and COmmunication in Medicine) [1, 2] は医療用画像の標準規格として広く知られている．特に，DICOM 形式は CT(Computed Tomography) や MRI(Magnetic Resonance Imaging)，CR(Computed Radiography) などで利用される．DICOM 形式は医用画像機器間の通信プロトコルを定義しているため，今後のネットワークなどへの幅広い利用形態が期待されている．DICOM 形式は撮影後の画像生成は RAW 形式を用いる場合が多く，保存には，ランレングス符号化 (RLE) [5] やロスレス JPEG [9] などの可逆圧縮符号化が用いられる．近年では，DICOM サーバから DICOM ビューアへの通信効率を改善するために JPEG2000 [8] などの不可逆圧縮符号化が利用される場合も増えている．DICOM 形式では様々な形式のデータを混在させることが可能である．このため，不可逆圧縮符号化では異なった形式の情報に欠損を生じる危険性がある．故に，DICOM 形式の保存には可逆圧縮符号化が望ましい．最近では，医療画像機器メーカーの多くは DICOM 形式に適した独自の可逆圧縮符号化方式の開発を進めている．残念なことに，各メーカーが保存形式として独自の圧縮方式を用いるため，DICOM 準拠を謳っているにもかかわらず，相互運用が行えない問題が多く発生しているのが現状である．本来，保存形式を規格化することが望まれる．求められる保存形式は普及した汎用手法の範疇であり，且つ，高い圧縮性能が求められる．加えて，ネットワーク利用を考えた場合，必要とされる圧縮方式は平易で高速に処理可能な手法でなければならない．

本研究では，いくつかの汎用手法を組み合わせることで，DICOM 形式の画像に対して高い圧縮性能を与える枠組みを提案する．汎用手法として，本研究ではランレングス符号化 (RLE)，ハフマン符号化 (Huffman encoding) [6]，MTF(Move To Front) [4]，BWT(Burrows Wheeler Transform) [3]，ダイナミックレンジコーダー (DRC) [7], LZSS(Lempel-Ziv-Storer-Szymanski) [12, 11] を取り上げる．これらの手法は単体では特に高い圧縮性能を与えないが，広く普及しており，アルゴリズムも比較的平易なものである．BWT に限っては，圧縮手法ではなく，データを整列させる前処理である．私たちが闇雲にこれらの手法を組み合わせれば，ほとんどの場合，圧縮性能は低下する．本研究ではこれらを組み合わせ，実験的に高い圧縮性能が得られることを確認した．

連絡先: 六井　淳，島根大学総合理工学研究科数理・情報システム学専攻，〒 690-8504　島根県松江市西川津町 1060, rokui@cis.shimane-u.ac.jp

2. 組み合わせ圧縮手法

本研究では，RLE, ハフマン符号化，MTF,BWT,DRC,LZSS の 6 つの手法を順次組み合わせる圧縮手法を検証する．本研究では組み合わせる最大数を 8 回に設定した．組み合わせ回数を最大で 8 回とした場合，原理的には 1,679,616 通りの組み合わせとなるが，MTF,BWT は前処理であるため，実際には 65,536 通りの組み合わせについて検証を行った．各手法の表記を本論文内では下記のように定める．

- b: BWT
- m: MTF
- r: RLE
- h: ハフマン符号化
- c: DRC
- l: LZSS

2.1 個々の圧縮手法の性能評価

本研究では，図 1 に示すような CR 画像を性能評価実験に使用した．CT 画像や MRI 画像と比べ CR 画像は比較的画像データサイズを大きく扱う場合が多く，高い圧縮率が求められる画像である．CR 画像サイズは 7,535,950 バイトのものを 2 種類使用した．

検証実験に使用した計算機の主な仕様は表 1 に示す通りである．

圧縮率の定義式は式 (1) の通りである．

図 1: CR 医療用画像

表 1: 検証実験に使用した計算機仕様

CPU	Intel Celeron 2.19 Ghz
Memory	2 GB
OS	Windows 7
Compiler	Microsoft Visual C++
Version	7.1.3091

$$\text{圧縮率} = \frac{\text{圧縮後のデータサイズ}}{\text{圧縮前のデータサイズ}} \times 100 \qquad (1)$$

図 2 は個別の圧縮手法を用いて圧縮を行った際の圧縮率を示したものである．図中で”bl”と示してあるものは，BWT の処理を行った後に，LZSS の処理を行ったという意味である．単体の圧縮手法では，ハフマン符号化が最も高い圧縮率を示しているが，LZSS,DRC と顕著な差は見られない．前処理として，BWT を行うことで LZSS と RLE には大幅な圧縮率の向上が確認された．特に，RLE は単体ではほとんど圧縮効果が見られないが，BWT と組み合わせることで高い圧縮効果が得られている．BWT 処理の後に，MTF を施した場合，ハフマン符号化と DRC で高い圧縮効果が得られている．実験結果より，BWT と MTF を用いた前処理は LZSS,RLE, ハフマン符号化,DRC の圧縮効果を高める機能を与えることが確認された．

次に，独自の DICOM 保存形式を採用する圧縮方式と市販 ZIP との比較を考える．2 方式をそれぞれ A（独自 DICOM 保存方式），B（市販 ZIP）とする．A の圧縮方式との圧縮性能比を式 (2),B の圧縮方式との圧縮性能比を式 (3) で定義する．

$$\text{圧縮性能比 } A = \frac{\text{圧縮後のデータサイズ}}{A \text{ の圧縮方式でのデータサイズ}} \times 100 \quad (2)$$

$$\text{圧縮性能比 } B = \frac{\text{圧縮後のデータサイズ}}{B \text{ の圧縮方式でのデータサイズ}} \times 100 \quad (3)$$

図 3 は独自の圧縮方式と個々の圧縮手法で圧縮したデータサイズの比率を示している．単体の圧縮手法では A との比率で 2〜3 倍，B との比率では 1.5〜2 倍の差を生じている．BWT と MTF を前処理に加えた場合の比較では，A との比率では

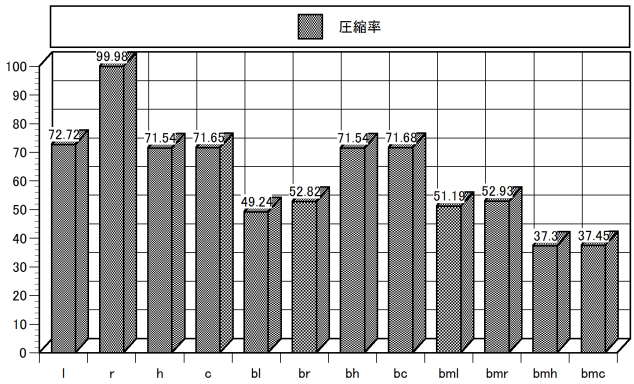

図 2: 個々の圧縮手法の性能評価

1.2〜1.7 倍の差を生じている．B との比率では，ハフマン符号化と DRC を組み合わせた場合において B の圧縮方式よりも高い圧縮性能が得られている．このように，平易な手法を組み合わせることで，効果的に圧縮性能を上げる可能性があることが確認された．

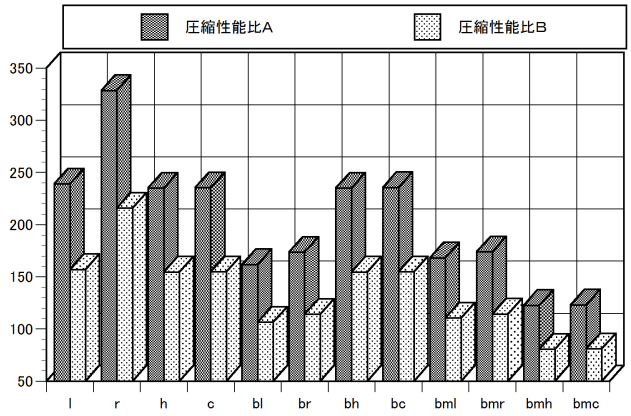

図 3: 個々の圧縮手法と市販圧縮方式との性能比

2.2 多重組み合わせを用いた圧縮性能評価

本研究では 65,536 通りの組み合わせを行い，高い圧縮性能が得られた上位 10 種類の組み合わせを示す．図 4 は多重組み合わせ上位 10 手法の圧縮率を示したものである．いずれの組み合わせも高い圧縮率を示している．特に，”bmrhbmrc”は圧縮率 28.84% と最も高い圧縮性能が得られた．

図 5 は多重組み合わせ上位 10 手法と市販圧縮方式とのデータサイズ比率を示している．A，B いずれの市販圧縮方式よりも高い圧縮性能が得られている．汎用の圧縮手法を組み合わせるだけで，高コストの市販圧縮方式と比べて同等もしくはそれ以上の圧縮性能が得られることが確認された．

2.3 多重組み合わせ手法の処理時間

前節では，汎用手法の組み合わせによって高い圧縮性能が得られることを示したが，組み合わせ数の増加に伴い，処理時間の増加が懸念される．図 6 は多重組み合わせ上位 10 手法の処理時間を示している．組み合わせ数が多い手法程，処理時間を多く要していることが分かる．処理時間のほとんどが BWT に伴う処理時間である．BWT は処理するデータサイズによっては多くの計算量を必要とする．本研究で用いた CR 画像の場合，7〜8sec の処理時間を要することが確認された．MTF

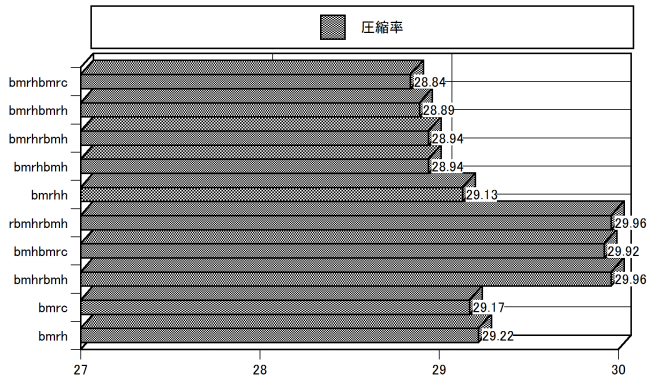

図 4: 多重組み合わせ上位 10 手法の圧縮率

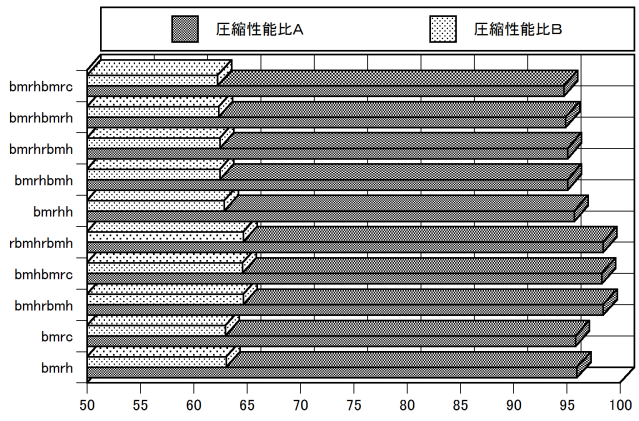

図 5: 多重組み合わせ上位 10 手法と市販圧縮方式との性能比

は 1.3sec 前後，ハフマン符号化，RLE,DRC は全て 1sec 未満の処理時間であった．BWT は効果の高い前処理である反面，処理時間が多くなることが知られている．BWT の処理時間をいかに抑えるかが課題となる．

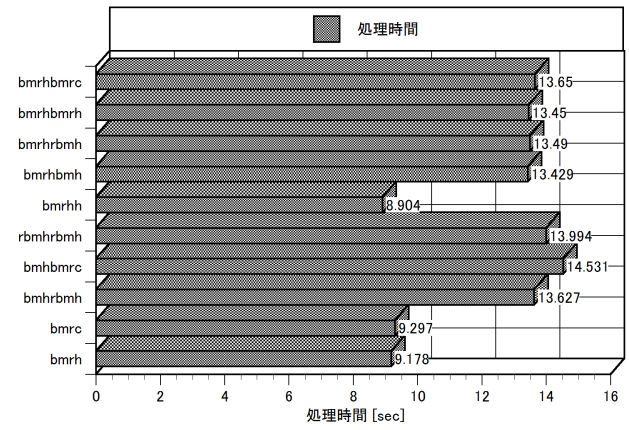

図 6: 多重組み合わせ上位 10 手法の処理時間

3. BWT 処理速度改善

BWT は処理の最初に，入力されたデータを解析する．データ解析では，データの出現個数をカウントし，ソート結果を格納する SortArray を初期化する．SortArray の初期化の段階で，出現個数をもとに，ソート対象データの先頭文字を元にした簡易ソートを行うことで，後のソート処理の処理対象を絞り込み，効率化を図っている．本研究では，BWT の処理速度を改善するために，DiffCharPoint 配列を作成した．SortArray を用いた場合，処理時間に関して次の 2 つの問題が発生する．

1. 同じ文字が連続している場合，先頭文字から順次比較していくと，比較できるコードが見つかるまで無駄な比較が行われる．

2. QuickSort を行う際，分割されていくコードのカウントに大きな偏りが発生し，$\log 2$ のオーダではなく，N（データ数）のオーダで比較が行われる．

DiffCharPoint 配列の配列番号は，SortArray と同じで，SortArray 上にあるコードに対し連続文字数をカウントし，次に異なるコード（比較を実行して意味のあるコード）が存在するオフセット値を表す．このオフセット値をもとに，比較対象の 2 つのコードについてオフセットの最小値を求めることで，比較対象コードの位置を容易に計算することができる．DiffCharPoint 配列の初期化は，SortArray 初期化のタイミングで実行できる．SortArray に配置する文字を読み込む際，連続文字の個数をカウントし，異なる文字が見つかった時点で，連続文字に対しオフセット値を減らしながら設定する．

図 7 は SortArray を用いた場合と DiffCharPoint 配列を用いた場合の処理時間を示している．

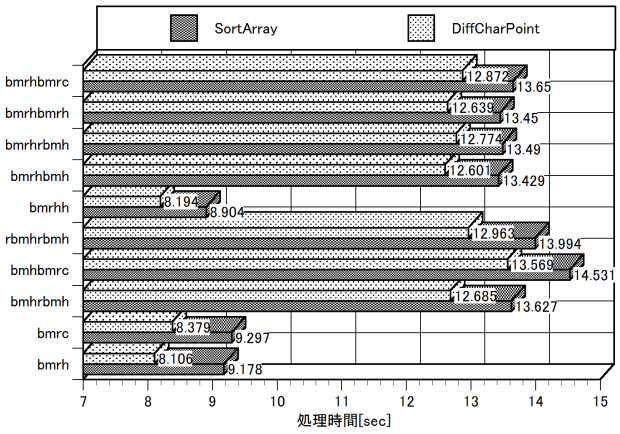

図 7: SortArray と DiffCharPoint の処理時間比較

図 7 より，BWT に伴う処理時間が 1sec 程度減少していることが確認できる．BWT で使用する配列処理を変更することで大幅に処理時間の短縮が実現できた．

4. まとめ

本研究では，汎用の圧縮手法を組み合わせることで，高い可逆圧縮性能を与える手法をいくつか示した．65,536 通りの組み合わせを検討した結果，処理時間を考えないならば，”bmrhbmrc” が最も高い圧縮性能が得られることを実験的に確認した．BWT に伴う処理時間を考慮した場合，”bmrhh” が圧縮性能と処理時間の双方のバランスが取れた組み合わせであることが確認された．

BWT に伴う処理時間増加の問題点を解決するために，従来の SortArray を用いた配列処理ではなく，DIffCharPoint 配列を配列処理として導入することで，BWT の処理時間の短縮を

図った．実験的に，DiffCharPoint 配列を用いることで BWT の処理時間の短縮ができることを確認した．

今後は，より高速化を図るべく BWT の処理時間改善方法を考案する．

参考文献

[1] Rosset A, Spadola L, Ratib O, *OsiriX: An Open-Source Software for Navigating in Multidimensional DICOM Images*, Journal of Digital Imaging Volume 17, Number 3, 205-216, 2004.

[2] Mildenberger P, Eichelberg M, Martin E, *Introduction to the DICOM standard*, Eur Radiol 2002.

[3] Manzini G, *An analysis of the Burrows–Wheeler transform*, Journal of the ACM (JACM) Volume 48, Issue 3, May 2001.

[4] Arnavut Z, *Move-to-front and inversion coding*, Proc of Data Compression Conference 2000, 193–202, 2000.

[5] Hinds SC, Fisher JL, D'Amato DP, *A document skew detection method using run-length encoding and the Hough transform*, Proc. of Pattern Recognition 10th International Conference 1990, Issue 16-21, 464–468 vol. 1, 1990.

[6] Iyengar V, Chakrabarty K, Murray B.T, *Huffman encoding of test sets for sequential circuits*, IEEE Transactions on Instrumentation and Measurement, Volume 47, Issue 1, 21–25, 1998.

[7] Mielikainen J, Toivanen P, *Clustered DPCM for the lossless compression of hyperspectral images*, IEEE Transactions on Geoscience and Remote Sensing, Volume 41, Issue 12, 2943–2946, 2003.

[8] Christopoulos C, Skodras A, Ebrahimi T, *The JPEG2000 still image coding system: an overview* IEEE Transactions on Consumer Electronics, Volume 46, Issue 4, 1103–1127, 2000.

[9] Langdon G, Gulati A, Seiler E, *On the JPEG model for lossless image compression* Proc of Data Compression Conference 1992, 172–180, 1992.

[10] Moffat A, Bell TC, Witten IH, *lossless compression for text and images* International Journal of High Speed Electronics and Systems, vol.8, no.1, 179–231, 1997.

[11] Pereira ZC, Pellenz ME, Souza RD, Siqueira MAA, *Unequal error protection for LZSS compressed data using Reed–Solomon codes*, Communications IET, Volume 1, Issue 4, 612–617, 2007.

[12] Fraser CW, *An instruction for direct interpretation of LZ77-compressed programs* Practice and Experience, Volume 36, Issue 4, 397–411, 2006.