

ブロックソートの PNG 圧縮への応用に関する検討

鈴木正人，三浦康之，渡辺重佳[†]湘南工科大学[†]

はじめに

デジタル画像は膨大なデータ量を有することから、これまでにデータ量の削減のためにさまざまなアルゴリズムの研究が行われてきた。一方、データ圧縮の前処理に適した手法として、ブロックソート (BWT: Burrows-Wheeler Transform, 以後 BWT と呼ぶ) と呼ばれる、データに偏りを作る処理が提案されており、ファイル圧縮などに用いられている。多くの画像圧縮方式では、圧縮する前のデータが偏りを持ちやすいという特徴を活かし、圧縮処理の中でハフマン符号化などの可逆圧縮技術が用いられている。これらの技術は BWT との相性が良いことが知られていることから、BWT を画像圧縮に応用することで圧縮率が向上することが期待される。そこで本稿では、一般的に広く用いられている可逆画像圧縮である PNG について可逆符号化前の処理として、いくつかの手法で BWT を実行し、減色した画像に対して適用することで生じる圧縮率の変化について評価する。

2. ブロックソート

ブロックソート (BWT: Burrows - Wheeler Transform) [1]は、1994 年に Michael Burrows と David Wheeler によって提案された手法で、可逆圧縮のモデル化に利用される。BWT はデータの並び換えによりデータの偏りを作り圧縮率を向上させる手法である。まず長さ n の文字列を巡回シフトし、 $n \times n$ 行列の配列に格納する。その配列を辞書順にソートし、 $n \times n$ 配列の n 列目と元の文字列が格納されている行の番号を出力する方法である。図 1. 図 2. に TOMATO の変換例を示す。

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | T | O | M | A | T | O |
| 2 | O | M | A | T | O | T |
| 3 | M | A | T | O | T | O |
| 4 | A | T | O | T | O | M |
| 5 | T | O | T | O | M | A |
| 6 | O | T | O | M | A | T |

図 1. 巡回シフト

Examination of the application to PNG compression of the Burrows-Wheeler Transform

[†]Masato Suzuki, Yasuyuki Miura, Shigeyoshi Watanabe, Shonan Institute of Technology

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | A | T | O | T | O | M |
| 2 | M | A | T | O | T | O |
| 3 | O | M | A | T | O | I |
| 4 | O | T | O | M | A | I |
| 5 | T | O | M | A | T | O |
| 6 | T | O | T | O | M | A |

図 2. ソート後

3. PNG

PNG (Portable Network Graphics) [3]は、静止画に対する可逆圧縮法の一つである。PNG にはチャンクと呼ばれる画像についての様々な情報を保持したデータブロックが複数ある。ほとんどのチャンクには決まった処理順番がなく、どのチャンクを処理する場合でもチャンク特別コードによりどのように処理するか判断できるようになっている。また圧縮には LZ77 とハフマン符号化の 2 段階で行う。

4. BWTのPNG圧縮への応用

PNG 圧縮に対して BWT を応用した 2 つの手法を用いて圧縮を行い、圧縮率の検討を行った。減色した静止画は、減色による画素値の拡散が発生する。BWT を用いた場合、拡散した画素値の集中を促すことが可能と考えられる。そこで、可逆圧縮である PNG に BWT を用いることで圧縮率が向上するか検討を行った。

1 つ目の手法として、RGB 3 色のうち G 成分に着目して BWT を行った。これを G ソートと呼ぶ。2 つ目の手法として、RGB の 3 成分を合わせたデータに対して BWT を行った。これを RGB ソートと呼ぶ。RGB3 成分を 1 個のソートの単位にすると 24 ビットになる。BWT は、膨大なデータを行うために、バケットソートをベースにした方法を用いる。そのため、24 ビットを単位としたソートは実装が困難であることから実装上の工夫が必要になる。そこで下のような方法を用いる。

1. 3 バイトをひとまとまりのデータとする。
2. 3 の倍数個目のデータのみをひろう。
3. ソートをする。
4. 後ろ 3 バイトを拾って並べる。

図 3. と図 4. に変換例を示す。

図 4. ではソートした後の最後の列である 2 列目の

3つのデータずつ出力することで変換を行う。出力は(255, 124, 100)、(175, 168, 192)となり変換前と変わらない結果になった。これは変換するデータが少ないためである。

| | 1 | | | 2 | | |
|---|-----|-----|-----|-----|-----|-----|
| 0 | 255 | 124 | 100 | 175 | 168 | 192 |
| 3 | 175 | 168 | 192 | 255 | 124 | 100 |

図3. 改良したBWTの巡回シフト

| | 1 | | | 2 | | |
|---|-----|-----|-----|------------|------------|------------|
| 3 | 175 | 168 | 192 | <u>255</u> | <u>124</u> | <u>100</u> |
| 0 | 255 | 124 | 100 | <u>175</u> | <u>168</u> | <u>192</u> |

図4. 改良したBWTのソート後

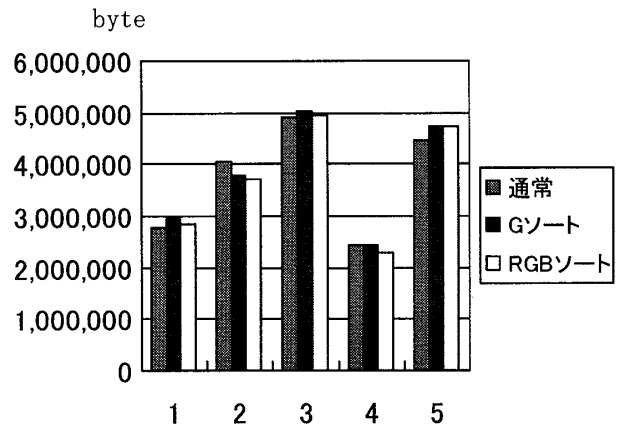


図8. 写真への応用結果

4.1. PNGへの応用の評価

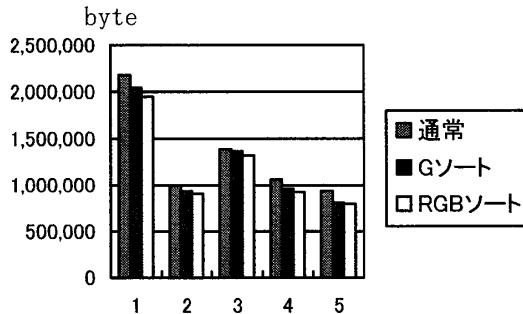


図5. アニメ画像への応用結果

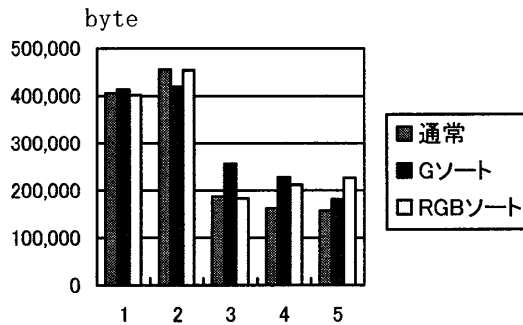


図6. イラスト画像への応用結果

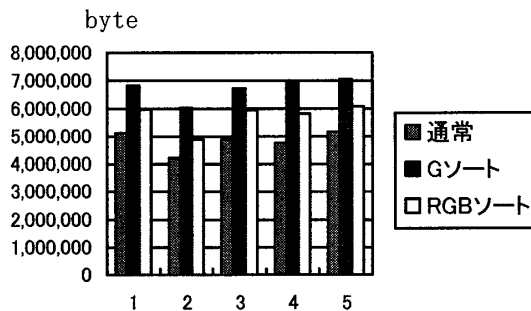


図7. 減色前の写真への応用結果

評価結果を図5～図8に示す。概ね、RGBソートのほうが圧縮率が高くなったが差は非常に少ない。これは減色した画像は色のパターンが少ないため、RGBすべてを使ってソートする効果は限定的であることによると思われる。しかし減色を行っていない写真の結果からRGBソートのほうがGソートよりは良いことがわかる。

通常の処理と比較すると減色したアニメの画像は提案した手法のほうが良い場合が多い。ただし、減色していない写真については通常の処理のほうが良い。このことからブロックソートによってLZ77+ハフマン符号化と相性の良いデータができやすいのは、誤差拡散法によって画素値に一定のパターンが連続で発生している場合であることが分かる。よってこれらの手法には画像の性質によって得手、不得手があることがわかった。

5. 今後の課題

今後、提案した手法は画像によって圧縮率の良し悪しが変わるので通常の処理と提案した2つの処理を実行して、良い結果が出たものを出力するようにする方式などの改良をしておく必要がある。

参考文献

- [1] M. Burrows, D. J. Wheeler "A Block-sorting Lossless Data Compression Algorithm", Digital Systems Research Center Research Report 124, not published, 1994.
- [2] PNG Development Group, "PNG (Portable Network