

## C-Pack アルゴリズムを拡張した主記憶データの非可逆圧縮手法

馬場裕之<sup>†</sup> 請園智玲<sup>‡</sup> 佐藤寿倫<sup>‡</sup>  
福岡大学<sup>†‡</sup>

## 1.はじめに

近年, 4K ディスプレイや 8K ディスプレイと呼ばれる超高解像度ディスプレイが普及してきている. 画像の解像度が上がるにつれ, 画像を表現するためのデータ量も同様に増加する. そのため, 画像を処理するプロセッサはチップ外にある記憶容量が大きく, 安価な主記憶上に大きな画像データを配置し, それを読み出しながら処理を進める必要がある.

主記憶に DRAM 等の安価で大容量の記憶素子を用いる場合, プロセッサの動作速度に比べ低速な主記憶の動作を許容しなければならない. また, オフチップに主記憶が存在する場合は, データバスの配線距離の問題から, 転送の速度もオンチップに比べ増加する. このプロセッサと主記憶の動作速度の差による処理性能低下はメモリウォール問題として広く知られる.

グラフィック処理に高い解像度が要求され, プロセッサが扱うデータ容量が増加する場合, メモリウォール問題による処理性能低下が問題となる. 本研究はグラフィックデータを扱う際のプロセッサと主記憶間のデータ転送効率に着目し, メモリウォール問題による処理性能低下を緩和するための主記憶データの圧縮手法を提案する. 提案手法は従来のメモリ値を圧縮する手法を拡張し, 非可逆の圧縮方針に修正することで, 圧縮率を向上させる. グラフィックデータはモニタなどのアナログデバイスに最終的に表示するためのデータであることから, 近似の値となった場合であっても, 誤差を許容できる可能性をもつ. 本研究はこの特性に着目した非可逆圧縮手法を提案する.

## 2. C-Pack

従来の主記憶値の圧縮手法に C-Pack[1]がある. C-Pack は圧縮したデータをキャッシュメモリ内に配置する. これは, データ圧縮が仮想的にキャッシュメモリ容量を増加させる効果をもつ.

Lossy Compression Technique for Main Memory Data  
Extending C-Pack Algorithm

Hiroyuki BABA<sup>†</sup>, Tomoaki UKEZONO<sup>‡</sup>, Toshinori SATO<sup>‡</sup>

<sup>†</sup>Grad School of Electronics and Computer Sci., Fukuoka Univ.  
8-19-1 Nanakuma, Jonan-ku, Fukuoka 814-0180, JAPAN.

<sup>‡</sup>Dept. of Electronics Eng. & Computer Sci., Fukuoka Univ.  
8-19-1 Nanakuma, Jonan-ku, Fukuoka 814-0180, JAPAN.

{td172009@cis, tukezo@, tsato@}.fukuoka-u.ac.jp

表 1 C-Pack の圧縮規則.

Code	Pattern	Output	Length(bit)
00	ZZZZ	(00)	2
01	XXXX	BBBB(01)	34
10	MMMM	bbbb(10)	6
0011	MMXX	bbbbBB(0011)	24
0111	ZZZX	B(0111)	12
1011	MMMX	bbbbB(1011)	16

C-Pack で圧縮したデータを主記憶に配置することはできない. これは圧縮率がデータ値に依存しており, 非効率な圧縮を強いられるデータ値に対しては, 圧縮データのサイズが非圧縮データのサイズをこえることに起因する. この問題を回避するために, 圧縮データと非圧縮データを主記憶に混在させるとするならば, 圧縮領域と非圧縮領域の色分け制御のための情報を主記憶に存在させる必要がある. 言い換えれば, キャッシュメモリ程度の容量に対する色分け制御の情報は比較的小さく, 実現が容易であるとも言える.

本研究で提案する圧縮機構は非可逆性を持つ代わりに, 必ず圧縮データのサイズが非圧縮データのサイズを上回ることがない. このため, 圧縮データを主記憶に配置できる. また, 必ず圧縮データの先頭から解凍することが可能である特性をもつことから, フロントサイドバス上で主記憶データのバースト転送を途中で中断することが可能となり, データ転送効率の向上を実現できる. 本稿ではこの手法を AC-Pack (Approximate C-Pack) と呼ぶ.

## 3. AC-Pack

C-Pack は辞書を用いた圧縮アルゴリズムである. 圧縮に使用した辞書の構成要素は圧縮データに内在することから, 圧縮処理後に辞書は破棄可能であり, 解凍時に辞書が再構成できる. また, 算術的な演算を必要としないため, 高速なハードウェアで実現可能である. C-Pack は表 1 で示す圧縮規則をもつ. C-Pack は 4 バイト単位でデータを圧縮する. Pattern の Z は 1 バイトの値 0 を示す. X は圧縮できない 1 バイトデータ, M は辞書に一致した 1 バイトデータを示す. C-Pack は Pattern で示された 4 バイトの組み合わせのどれかに分類し圧縮する. Output は圧縮後

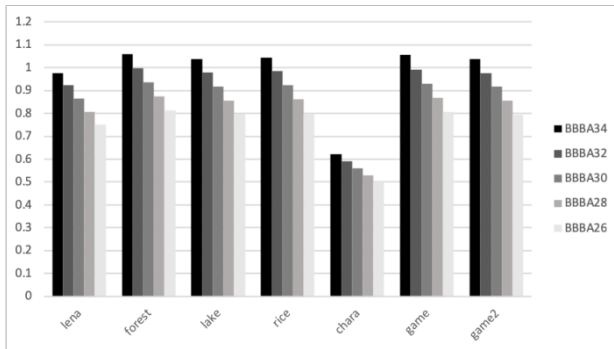


図 1 圧縮率評価.

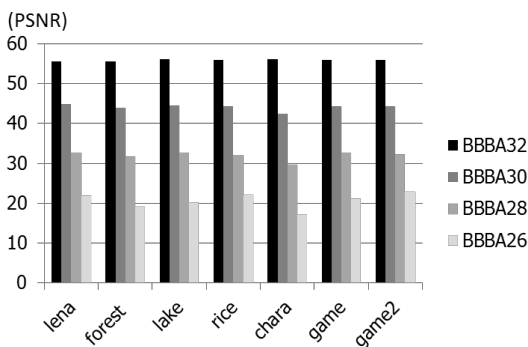


図 2 PSNR 評価.

のデータ形式である。カッコ内の数字は分類先を示すコードである。B は 1 バイトの非圧縮データを示し、b は辞書のインデックスを示す 1 ビットを意味する。Length は Output が計何ビットとなるかを示している。Pattern XXXX の 34bit が圧縮データを非圧縮データより大きくする唯一の要因である。このため、AC-Pack は Pattern XXXX の最下位バイトの中の下位ビットを切り捨てるように圧縮形式を修正した。部分的に切り捨てられるバイト位置を A とすると、Pattern XXXX の Output は BBBA(01)となり、もし A の下位 2 ビットが切り捨てられるとするならば、Pattern XXXX の Length は 32 となり、圧縮後のデータは必ず非圧縮データよりサイズが小さいかまたは等しくなることを保証できる。

#### 4. 評価

AC-Pack を画像データの圧縮に適用した場合、圧縮率はどの程度改善され、画像の品質がどの程度低下するのかを評価するために、シミュレータを作成した。作成したシミュレータは画像ファイルを入力とし、AC-Pack で圧縮した後に、即座に解凍することで、圧縮率を計算する。それに加え、シミュレータは解凍後のデータを用いて画像ファイルを再構成し、出力できる。この出力画像をもとに PSNR(Peak Signal Noise Ratio)で画像の劣化を計測し比較する。

図 1 に AC-Pack の圧縮率を示す。凡例の BBBA の後に続く数字は XXXX 圧縮時の圧縮後

総ビット数を示す。BBBA34 は従来の C-Pack の圧縮方針を示している。BBBA32~26 は AC-Pack である。グラフの横軸は画像を示す。本稿評価では、7 つの画像について評価した。それぞれの画像に便宜的に lena, forest, lake, rice, chara, game, game2 と名前を付けた。これらの画像はインターネットから取得したカラーの画像データである。画像データは主記憶上に展開される際に PPM(Portable Pixmap Format)形式のデータ整列順で配置されることを想定した。

グラフの縦軸は圧縮率の割合である。1 ならば、圧縮後にデータサイズが変わっていないことを意味する。AC-Pack のすべての圧縮結果で 1 を下回ったことから、AC-Pack の圧縮データのサイズが非圧縮データを超えないことが確認できる。C-Pack の圧縮では、lena と chara を除いた 5 つの画像で 1 を超える結果となっている。図 1 より、バイト位置 A の下位ビットを削るほど、圧縮率が上がることがわかる。これは画像データに Pattern XXXX が頻出し、これが圧縮率に大きく影響を与えたことを意味する。すべての画像を通して、切り捨てビット数設定の範囲内で約 10%~20%程度の圧縮率の幅が観測された。

図 2 に PSNR を示す。すべての画像で、ほぼ同様の画像品質低下の傾向が観測された。使用に耐えうる画像品質は 30dB 以上であると一般的に言われている。このことから BBBA26 は使用に耐えられない誤差を含む画像となったことがわかる。

#### 5. おわりに

本研究は C-Pack を拡張し、非可逆圧縮を実現することで、圧縮データのサイズが非圧縮データのサイズを超えないことを保証する手法を提案した。また、Pattern XXXX の下位ビットを切り捨てるビット幅によって圧縮率が変動し、画像の品質も同様に変わることが確認できた。

本稿の評価を通して、主記憶上に展開される画像の画素値データの規則性を確認できた。これに合わせて更に圧縮率を上げるためのビットの切り捨て手法が考えられる。切り捨て手法のバリエーションを増やし、更に評価を進めることが、今後の課題として挙げられる。

#### 謝辞

本研究は JSPS 科研費 JP17K00088 の助成を受けた。

#### 参考文献

- [1] Xi Chen, Lei Yang, Robert P. Dick, Li Shang, and Haris Lekatsas, "C-pack: A high-performance microprocessor cache compression algorithm," IEEE Trans. Very Large Scale Integr. Syst., 18(8):1196-1208, August 2010.