# 【MSX SCREEN5専用圧縮フォーマット CS5】

(1) 圧縮は PC上で実施する
(2) 展開は MSX実機上で動作する
(3) MSX実機上での展開は「ゲームの画像パーツ読み込み」として実用レベルの速度を実現する


パレットの情報はオプションであり、存在しない場合はパレット情報を含まない。
ゴロム符号テーブルはオプションであり、存在しない場合は 0～16 が順に格納されていると見なす。

ゴロム符号テーブルは、5bit の値が 17個並んでいる。
ゴロム符号(mod 3) の 0～16 に対応するビット列が、ID#0～16 の何に対応するかを示している。
ID#0～15 は、画素値そのものを示す ID である。
ID#16 は、下記の構造を持つ。

| | |
|---|---|
| ID#16 golomb | 2～8bit |
| 直前の位置 | 8bit: 直前の 256画素を蓄えた画素列の何番目と同じ画素列かを示す。 |
| 画素列の長さ | 2～86bit: 一致する長さ。(長さ - 3) をゴロム符号にした値。 |
| 反復回数 | 2～45bit: 繰返し回数。(回数 - 1) をゴロム符号にした値。 |

圧縮時に、ID#0～16 の発生頻度(ヒストグラム)をとり、最も頻発する ID に一番短いゴロム符号を割り当てる。
このヒストグラムは、画像によって異なるため、ゴロム符号テーブルで違いを吸収している。

【考察】

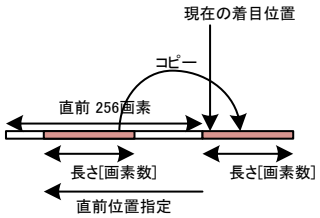
画像データの符号

画素値そのものを表現する符号

| | |
|---|---|
| 画素値 | 16通り |

直前の画素列の反復を表現する符号

| | |
|---|---|
| 反復識別子 | 1通り |
| 直前位置指定 | 256通り |
| 長さ | 256通り |




仮に上記の表現を採用する場合、具体的にどのようなビット列で表現するかを考える。

| | |
|---|---|
| 画素値 | 16通り |
| 反復識別子 | 1通り |

17通りの符号語をゴロム符号(m=3)で表現してみる。

| 値: 符号 (ビット列) |
|---|
| 0: 10 |
| 1: 110 |
| 2: 111 |
| 3: 010 |
| 4: 0110 |
| 5: 0111 |
| 6: 0010 |
| 7: 00110 |
| 8: 00111 |
| 9: 00010 |
| 10: 000110 |
| 11: 000111 |
| 12: 000010 |
| 13: 0000110 |
| 14: 0000111 |
| 15: 0000010 |
| 16: 00000110 |

17通りの符号の出現頻度(ヒストグラム)を採って、頻度が高い順に短いビットの符号を割り当てる。
左図の値は、その対応表のインデックスとして利用する。
対応表は、ヘッダに付ける。

対応表は下記のようなフォーマットにする。
(1) 一つの値は 5bit固定長。
(2) 値が 17個並んでいる。
(3) 0～15 は画素値。16 は反復識別子。17～31 は予約。
(4) 85bit のテーブルである。

下記のような画像の圧縮を考えてみる。

4bit * 16pixel で 64bit

| | | | |
|---|---|---|---|
| 符号語1 | □ | 画素値 | |
| 符号語2 | ■ | 画素値 | |
| 符号語3 | □ | 画素値 | 長さ1の「類似」では画素値の方を採用する |
| 符号語4 | ■ | 画素値 | 長さ1の「類似」では画素値の方を採用する |
| 符号語5 | | 直前4画素のコピー | |
| 符号語6 | | 直前8画素のコピー | |

使われている符号は、画素値 "白", 画素値 "灰色", 直前コピー の 3種
類のみで、それぞれ 2回ずつ発生するので頻度に偏り無し。
短いゴロム符号から順に割り当ててみる。

| | | | |
|---|---|---|---|
| 符号語a | □ | 画素値 | 10 |
| 符号語b | ■ | 画素値 | 110 |
| 符号語c | | 直前のコピー | 111 |

10 110 10 110 111_00000100_00000100 111_00001000_00001000
→ 48bit

しかし、5つ目の符号語c は、符号語a と b を組み合わせた方がより短くなる。

10 110 10 110 10 110 10 110 111_00001000_00001000
→ 39bit

圧縮にはなるが、割と繰り返しているパターンであってもあまり潰れない。
この理由は、
そもそも画素値が 4bit なのに対して、画素値表現は最高でも 2bit までにしか縮まないので MAX50% である。
コピーの表現は、位置・長さを 8bit 固定にしてしまうと、短い長さの反復では、かえって語長が長くなってしまうことである。
これをこのまま課題として改善を考えてみる。

2画素ペアの 1byte を処理の単位にしてみる。8bit になるので、最高 2bit だとすれば MAX25% まで圧縮できる。
コピー表現も、位置・長さの指定を可変長にしてみる。
MSX の画像サイズからすれば、この位置・長さのバリエーションもある程度限られるため、可変長の対応表のヘッダを付けてもいいかもしれない。

[長さ] の次に [反復回数] も付けることとすれば、

10 110 111_00000010_00000010_00001110
→ 32bit

[位置][長さ][反復回数] をゴロム符号で表現するようにすれば、さらに短くなる。

圧縮について考える

反復表現の部分はなるべく長い画素列に適用した方が圧縮率が高くなるが、
使える符号語が限られているとすれば、全部を置換するのは好ましくない。
そのため、反復表現で圧縮できる画素列において、反復表現に置き換えるか、複数の反復表現に置き換えるか、
あるいは画素値表現に置き換えるかを決定するアルゴリズムが必要となる。