

限定色画像の圧縮における2次元モデルの検討

高瀬哲嗣

萩原剛志

金田悠紀夫

神戸大学

〒 657-8501 兵庫県神戸市灘区六甲台町 1-1

(078) 881-1212 (内線)6324

takase@maestro.cs.kobe-u.ac.jp

ogihara@cs.kobe-u.ac.jp

kaneda@cs.kobe-u.ac.jp

あらまし インターネット等で多く用いられている画像の種類の一つに限定色画像がある。これは画像の中でよく使われている色を代表色として選び出し、その限定された色だけで画像を表現する。限定色画像の無損失圧縮には、これまで LZW 法などの汎用の圧縮法が利用されてきた。しかし、この方法では画像データを1次元的に捉えて符号化しているため、効率が良くない。そこで本研究では、画像の持つ2次元の情報を有効に利用するための符号化モデルを提案する。このモデルでは、注目画素の近傍内に同じ色があるかどうかに基づいて近傍画素にラベル付けを行い、画素値推定の文脈として利用する。また、この符号化方式の圧縮能力についても報告する。

キーワード 限定色画像, 有限文脈モデル, 2次元モデル, インターバル符号

Compression Method for Limited Color Images Using Two-dimensional Model

Satoshi Takase

Takeshi Ogihara

Yukio Kaneda

Kobe University

1-1 Rokkodai-cho, Nada-ku, Kobe, 657-8501 Japan.

(078) 881-1212

takase@maestro.cs.kobe-u.ac.jp

ogihara@cs.kobe-u.ac.jp

kaneda@cs.kobe-u.ac.jp

Abstract Usual compression methods for limited color images do not use effectively two-dimensional context of images. In this paper, we propose a variation of finite context model to improve compression ratio. In this method, a context is made from the neighborhood of the current pixel to refer to two-dimensional information. We also propose a technique to reduce execution time using simply structured model. Instead of color indexes, a context is constructed from labels which represent the location of pixels with the same color in the neighborhood. We will also show the results of compression experiments.

key words Palettized color image, Finite-context model, Two-dimensional model, Interval coding

1 はじめに

限定色画像の無損失圧縮には、これまで LZW 法などの汎用の圧縮法が利用されてきた。しかし、この手法では画像データを 1 次元的に捉えて符号化しているため、画素間の 2 次元方向の関連が圧縮に反映されず、圧縮率の向上に寄与していない。

画像データを 2 次元方向に捉えながらもメモリ容量にあまり負担をかけないラベリングモデル [1] が既に提案されているが、実行時間の減少と共に、圧縮率も 1 次元モデル [2] より低下するという結果になった。

そこで本研究では、このラベリングモデルを改良して、実行速度を維持しながら圧縮率を向上する方法を提案する。

2 限定色画像について

フルカラー画像はそのままでは非常に大きなデータ量となるため、何らかの方法でデータの縮小を図る場合が多い。限定色画像は、画像の中で良く使われている色を代表色として選び出し、その限定された色だけで画像を表現する。フルカラーの画像を 256 の代表色のみで表示すると、もとのデータ量の 3 分の 1 に抑えることができる。

限定色画像は、代表色の情報が記録されたパレットと呼ばれる部分と、その画像における代表色の番号(パレット番号)の並びが記録されたデータ部分とに分かれている(図 1)。

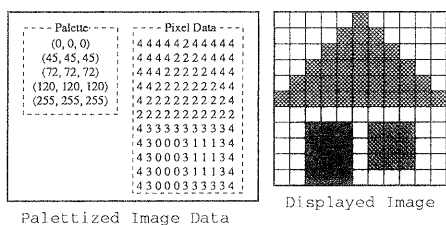


図 1: 限定色画像

3 有限文脈モデル

圧縮手法の一つに有限文脈モデルを用いた符号化 [2][3] がある。このモデルは、その時点での文脈と出現頻度情報から次のシンボルの出現確率を推定する方法である。文脈とは、ある時点までに符号化されたシ

ンボルの情報を用いて、その時点の状態を表現したものと行うことができる。通常は、これから符号化するシンボルの直前に出現したシンボル列を文脈として用いる。文脈として参照するシンボルの個数を、有限文脈モデルの次数と呼ぶ。

文脈ごとに、その文脈の直後に現れたシンボルの出現頻度を記録しておき、これを文脈表と呼ぶ。シンボルは、それが出現した時の文脈の持つ出現頻度情報を用いて符号化、および復号化を行う。文脈表は符号化、復号化に伴って動的に更新して行く。

モデルの次数が上がれば、シンボルの出現確率をより正確に予測できるので、そのシンボルを符号化するのに必要なビット数は少なくて済む。つまり、圧縮能力は向上する。しかし、次数が上がるにつれて作成される文脈表の数が増え、個々の文脈自体の出現頻度が低くなるため、シンボルの出現確率を正しく推定できなくなる。また、文脈表の増加に伴う使用メモリ量は文脈のべき乗で増加してしまい、大きくシステムに負担をかけてしまうという問題もある。

4 ラベリングモデル

限定色画像に有限文脈モデルを適用しようとする場合、直前に出現した画素、つまり注目画素の左方向の画素だけではなく、上方向の画素も含め、注目画素の近傍の画素を用いて文脈を構成した方が圧縮能力は高くなる [2]。しかし、近傍の色番号を用いて文脈を構成する場合、現在の文脈にはない色の組合せが出現する度に新たに文脈表を作成する必要がある。色数が多いと必要以上に文脈表の数が増え、圧縮能力が低下する可能性もある。

限定色画像は、異なる色を交互に並べて中間色を表すなど、少ない色数で多くの色を近似的に表現する事が多い。そのため、ラベリングモデル [1] では中間色などを表現する時に組み合わせる色の配置パターン情報に注目して文脈を構成している。

注目画素に対して 2 次元的に捉えた近傍の色を基にラベルと呼ばれる番号を付ける。注目画素からの物理的距離によって近傍に優先順位を付け(図 2a)、注目画素の左の画素を 0 番のラベルとする。また、変数 i を用意し、 $i=0$ としておく。以下、優先順位の順に近傍画素にラベル付けを行う。その画素が、すでにラベル付けされた画素と同じ色であれば同じ番号をラベルとし、同じ色の画素がなければ i を 1 インクリメ

ントし、これをラベルとする。このラベル付け作業は参照する近傍の画素数(次数)だけ行われる。この様にして、近傍の画素番号を全てラベルで表現する(図2b)。

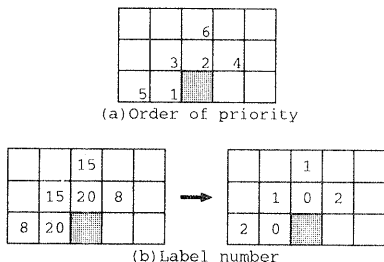


図 2: 6 近傍の時のラベリング

このラベルを用いて文脈を構成し、注目画素の値を推定する。注目画素は、同じ色の画素が近傍内に存在すれば同一ラベルを付け、同じ色の画素が存在しなければ ESC ラベルを付けて文脈表の出現頻度を更新する。ESC ラベルには近傍画素数と同一の数を用いる。

近傍内のラベルの種類は、近傍を構成する画素数よりも少ないため、圧縮過程で作成される文脈数は、画素番号を用いて文脈構成する方法よりも極めて少なくなる。

このようなラベリングは、インターバル符号 [4] を 2次元に拡張したものとも考えることもできる。しかし、ラベルによって参照されるシンボルは直前に符号化されたものではないという点で異なる。

5 ラベリングモデルの改良

ここでは上述のラベリングモデルを改良して圧縮率を向上させる手段を提案する。

5.1 近傍の拡張

ラベリングモデルではラベルを付ける近傍の画素数は次数に等しい。例えば 3 次のモデルでは、3 近傍のラベルを参照して文脈を構成し、ラベルで表現された注目画素を推定している。しかし、ラベル付けされた画素が少ないと、近傍内に注目画素と同じ色が現れない場合が多くなり、符号化効率が悪くなるという問題がある。

		18	19	12	20		
	17	14	8	6	9	15	
	13	7	3	2	4	10	16
	11	5	1				

図 3: 3 次 20 近傍のラベリング

そこで、ラベル付けする近傍の画素数を増やし、その中で注目画素に近い少数の画素のラベルのみを文脈として用いるように改良した。これによって、同色の画素が近傍に現れる可能性が増えたと考えられる。

20 近傍にラベル付けした 3 次の近傍状態の例を図 3 に示す。実線内の 20 個全ての近傍にはラベルが付けられている。文脈を構成するために参照する画素は破線内の 3 個のみとなる。

5.2 平均値の導入

5.2.1 平均値を利用したモデル (1)

ラベリングモデルにおいて、画像のある領域に同一の色が多く現れる傾向があるならば、近傍のラベルは 0 に近い番号を付けられる事が多くなり、逆に色の変化の激しい領域では異なる番号を付けられる事が多くなるであろう。つまり、ある領域内のラベルの平均値が大きいか小さいかでその領域の色の変化度合を測る事ができる。そこで、文脈を決定する要素のひとつとして近傍内のラベルの平均値を利用することを考える。

従来モデルでは、1 次の文脈に当たる、注目画素の左隣の画素のラベルは常に 0 であり、画素値の推定に寄与していなかった。このモデルでは、1 次の文脈としてラベルの平均値を用いることにする。平均値自体は実数値であるが、取りうる値の範囲をあらかじめいくつかの領域に分割しておき、実際の平均値が何番目の領域に含まれていたかによって文脈表を選択することにする。ここでは、平均値を整数に丸める事で文脈表の範囲を決定する。

また、平均値の基となるラベルには、近傍がかつて注目画素であった時に付けられたラベルを使用する。注目画素に付されるラベルは同色の近傍と同じラベルなので、近傍の近傍も含めた領域の画素配置を参照しているとも考えられる。

アルゴリズムを記載する。

近傍の数を n ($n > 0$) とし、 R_n を優先順位 n の

画素がかつて注目画素であった時付されたラベルとする。また、 L_n を優先順位 n の画素のラベルとする。

アルゴリズム:

1. $i = 1$ とし、0 次の文脈表に着目する。
2. 注目画素の n 近傍にラベル付けを行う。
3. R_0 から R_n までの平均値を算出する (図 4a)。また、小数点第 1 位を四捨五入して整数に丸める。
4. 平均値を基に 1 次の表へと移動する (図 4b)。平均値に対応する文脈が無い場合 7. へ移る。
5. L_i を参照して、対応する表を決定し、高次表に移動する。対応する文脈が無い場合 7. へ移る。
6. i をインクリメントして、指定された回数まで 5. を繰り返す。
7. 現在の注目画素の色と近傍を比較して、一致した近傍の L と現在注目している文脈表を使って符号化する。一致しなかったら ESC 記号を出して、0 次で符号化する。
8. 次の注目画素を参照し、表を 0 次に戻して 1. に戻る。

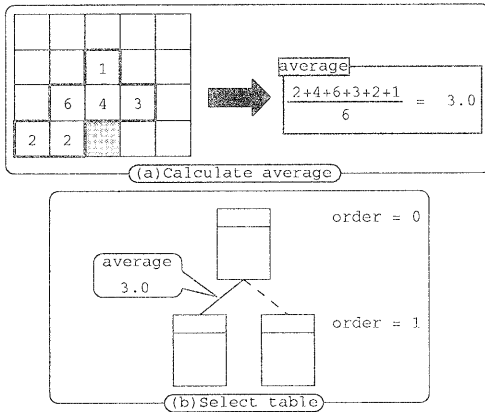


図 4: 6 近傍時の平均値の算出例

実験結果を 6 章の図 8,9 に記載する。

5.2.2 平均値を利用したモデル (2)

平均値モデル 1 では、近傍がかつて注目画素だった時のラベルを用いて平均値を算出した。しかし、近傍に注目画素と同色の画素が存在しないと、注目画素のラベルは ESC (ラベルの最大値 = 参照画素数) となり、近傍内にただ一つでも ESC ラベルが付くと、急激に平均値が増加してしまう。

そこで、このモデルでは現在の近傍のラベルの平均値を計算して、文脈の決定に利用する。この平均値は、注目画素の近傍の色の変化度合を表すため、平均値モデル 1 の平均値よりも局所的な情報となる。

4 章の手順で近傍にラベルが付されているものとする。このラベルの平均値を算出し、0 次から 1 次へと表を選択する際に用いる。また、アルゴリズムは 5.2.1 の 3. 「 R_1 から R_n 」を「 L_1 から L_n 」にのみ変更したものである。

6 実験と結果

実験に使用した GIF 画像は、イラスト画像、CG 画像、自然画像である (図 5、6、7)。また、それぞれの属性を表 1 に示す。

画像	種類	サイズ	byte	色数
phoenix	イラスト	640x480	307200	16
mac	CG	640x480	307200	256
bird	自然	640x480	307200	209

ベースプログラムとして、文献 [2] に掲載されている算術符号プログラム「arith-n.c」を利用し、Sun Ultra 5 (UltraSPARC-III 270MHz) 上で実験を行った。

上述の近傍を拡大したモデル (以下、拡張ラベリングモデルと呼ぶ) と平均値モデル 1、および 2 について、いずれも 3 次の文脈を用いて圧縮を行った結果を表 2 に示す。また、従来のラベリングモデルの 3 次 (3 近傍) の結果と、比較のために、汎用的に用いられている gzip(best) の結果も記載する。図 8 は 3 次 20 近傍の場合の比較のグラフである。また、画像 3 に対する 20 近傍時の実行時間のグラフを図 9 に示し、6 次および 8 次の 20 近傍での圧縮率比較を表 3 に示す。

これらの実験では、画像を構成する色番号列のみを符号化している。パレットやヘッダ情報は圧縮率の計算の対象としていない。これは gzip についても同様である。また、圧縮率は圧縮前データに対する色番号列を符号化した出力データの百分率で、小さければ小さい程高圧縮である。



図 5: 原画像 1:phoenix

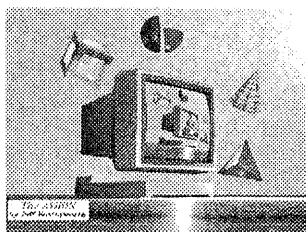


図 6: 原画像 2:mac



図 7: 原画像 3:bird

表 2: 3 次各手法での圧縮率の比較

手法	近傍数	圧縮率		
		Phoenix	Mac	Bird
ラベリング	3	22.2	50.8	58.3
拡張	10	20.3	45.6	52.8
ラベリング	20	19.5	43.5	51.2
平均値	10	19.1	45.3	52.5
モデル (1)	20	18.7	43.0	52.3
平均値	10	19.8	44.8	51.9
モデル (2)	20	19.0	42.4	49.9
gzip	-	21.6	50.9	61.4

表 3: 6 次および 8 次の 20 近傍での圧縮率の比較

手法	次数	圧縮率		
		Phoenix	Mac	Bird
ラベリング	4	18.3	49.4	55.1
	6	16.9	46.0	53.1
	8	16.1	45.6	54.4
拡張	4	17.4	43.2	50.4
	6	16.4	42.4	50.3
ラベリング	8	15.9	44.2	53.4
平均値	4	17.0	42.8	50.2
	6	16.3	44.8	53.3
モデル (1)	8	16.4	53.4	65.5
平均値	4	17.2	42.5	49.8
	6	16.3	43.3	51.5
モデル (2)	8	16.1	49.5	59.3

7 考察

7.1 圧縮能力

拡張ラベリングモデルはラベリングモデルの近傍数を大きく取れる様に改良したものである。図 8 の結果

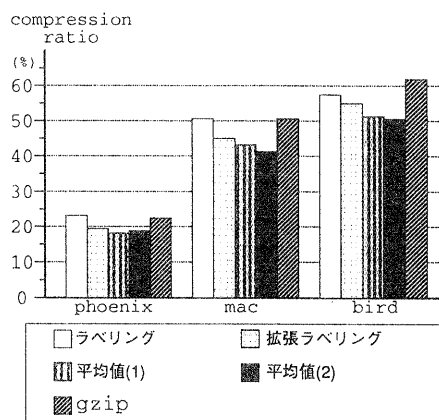


図 8: 3 次 20 近傍での圧縮率比較

をみれば、拡張ラベリングモデルの方が圧縮能力が高い事が言える。文脈を構成するための画素の領域よりも広い近傍を参照することによって、出現する画素値の推定がより正確に行われることがわかる。

平均値モデルは、拡張ラベリングモデルの 0 次から 1 次までの表の決定に平均値を用いるように改良したものである。平均値モデル 1、2 と拡張ラベリングモデルを比較すると、ほとんどの画像において平均値モデルの圧縮率の方が優れている。これは 0 次から 1 次への表を選択する際に平均値というパラメータを用いる事が有効であることを示している。

さらに、平均値モデルの 1 と 2 を比較すると、ほとんどの画像で 2 の方が良い結果が得られた。これはモデル 1 の問題点であった、ESC 記号によって無意味に増大した平均値で表を選択する事が解消されたものと考えられる。

表 3 では次数を上げた時の圧縮率比較を示してい

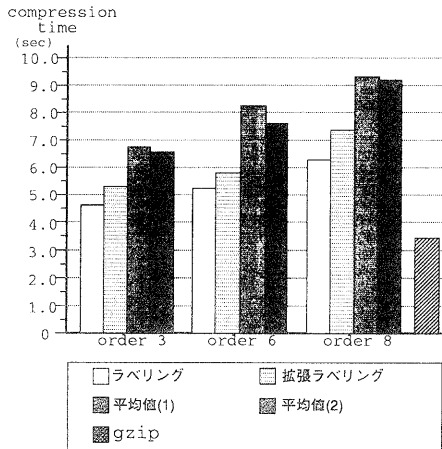


図 9: 20 近傍での画像 3 の実行時間比較

る。画像によって多少の差はあるが、提案モデルは 4 次程度が最も良い圧縮能力を示している事がわかる。それ以上の次数で符号化しても、圧縮率はむしろ下がる。これは、文脈の数が多くなり、個々の文脈の保持する頻度情報が十分更新されずに注目画素を正しく推定できなくなるためであろう。

上記の 3 手法と gzip などの汎用圧縮手法と比較しても良い圧縮率を示している。これは 2 次元的に画素を参照する事が有効である事を示している。

7.2 実行時間

図 9 で示す様に、全ての画像において gzip が最も速い事がわかる。これは、gzip が LZSS 符号と Shannon-Fano 符号を組み合わせた符号化方式 [3] であるのに対して、他の手法は算術符号である事が主な原因であると思われる。数字的には gzip の実行時間は他手法の半分未満であるので、実行時間に関しては gzip は実用的な手法であると言える。

ラベリング、拡張ラベリング、平均値それぞれの実行時間を比較すると、ラベリングモデルが最も高速である事がわかる。これは、近傍数が少ない分だけ画素参照やラベル付けの作業が早く行われる事が原因と思われる。平均値モデルは注目画素が変更される度に近傍の平均値を再計算するため、他手法よりも時間がかかるものと思われる。

また、平均値モデルは 1 よりも 2 の方がわずかでは

あるが少ない時間で処理できる。これは、モデル 1 でそれぞれの注目画素のラベルを配列に保存し参照していくために、単にラベルを参照するだけのモデル 2 との差が出たものと思われる。

7.3 近傍の画素数

表 2 の結果より、近傍を多く設定する事は圧縮率向上において有効な手段である事が示された。しかし、同時に実行時間も比例して増加している事がわかる。さまざまな適用実験の結果から、近傍数をさらに 10 増やしたとしても、圧縮率向上は 1% 程度に過ぎないことが分かっている。これらの結果から、10 近傍あたりにとどめておくのが、トレードオフの関係上、最も効率的であると考えられる。

8 おわりに

今回提案したモデルはラベリングモデル近傍を拡張し、文脈表の決定手順に平均値パラメータを与えたものである。これにより、一応の圧縮率の向上は見取れたものの、劇的に性能が向上した訳ではない。ラベルは単に画素の色の配置に着目した考え方で、色情報そのものは含まれていない。そこで今後は、平均値の代わりに画素番号の情報をモデルに反映させ、さらに細かく注目画素を推定する方法について検討を進める予定である。

参考文献

- [1] 萩原, 高瀬, 金田: “インターバル符号による限定色画像の圧縮について” 情報理論とその応用シンポジウム (SITA'99)(1999).
- [2] M.Nelson: (訳 萩原, 山口): “データ圧縮ハンドブック”, トッパン (1994).
- [3] 植松友彦: “文章データ圧縮アルゴリズム入門”, C-Q 出版社 (1994).
- [4] <http://candy.alacarte.co.jp/~kojif/mu/comp/interval.htm>