大阪学院大学　人文自然論叢 ＜論　説＞
第81-82号　2021年３月

フレームメモリ容量削減を目的とした視覚的ロスのない画像圧縮手法

山　口　雅　之

A Visually Lossless Image Compression Method for Frame Memory Reduction

Masayuki Yamaguchi

あらまし

フレームメモリ容量や画像データアクセス時の通信量を削減するひとつの方法として，フレームメモリに格納するときに画像データを圧縮し，画像処理を行う前に伸張する方法がある．本稿では，このときに用いるフレームメモリ向き画像圧縮手法に求められる要件について整理し，新たな手法を提案する．提案手法は，フレームメモリ圧縮手法に必要な要件を備え，圧縮率1/3の場合に一般的に視覚的に画質の劣化が認められないと言われている PSNR 指標で35dB 以上の性能を保持することが可能である．本稿では，いくつかの画像に提案手法を適用して評価実験を行い，その有効性を実証する．

キーワード：フレームメモリ容量削減，画像圧縮，視覚的ロスレス

1．はじめに

近年，モバイル機器やデジタル TV などの映像機器の急速な高解像化に伴い，処理対象となるデジタル画像のデータ量が増大している．画像処理には画像データを格納するためのフレームメモリが必要な処理も多く，外付けメモリの増加によるコストアップや，画像処理時のデータアクセスにかかるデータ通信量の増大に起因する消費電力増加などの問題が生じている．メモリ容量やデータ通信量を削減するひとつの方法として，画像圧縮を用いる方法がある．すなわち，フレームメモリに格納するときに画像データを圧縮し，メモリから読み出して画像処理を行う前に伸張する方法である．フレームメモリ向きの画像圧

縮には，圧縮によって視覚的にロスを生じない準可逆な性質（視覚的ロスレス）のほか，メモリアクセスやハードウェア回路実装に起因するいくつかの要件が必要である．本稿では，このようなフレームメモリ向き画像圧縮手法に求められる要件について整理し，圧縮率1/3〜1/4をターゲットとした準可逆な画像圧縮手法を提案する．

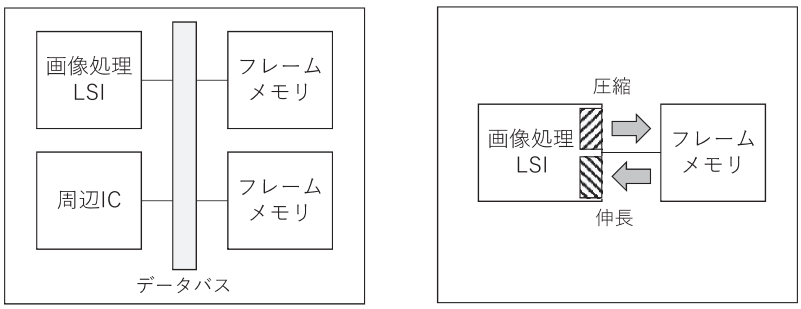
本稿の構成は以下の通りである．第２節ではフレームメモリ向き画像圧縮に対する要件について考察し，第３節で既存の画像圧縮手法に関する従来研究についてフレームメモリ向き画像圧縮の観点から論じる．第４節では提案手法のアルゴリズムについて詳細を説明する．第５節で提案手法の評価実験の結果を示し，最後に第６節で結論を述べる．

2. フレームメモリ向き画像圧縮の要件

フレームメモリは，通常，数枚分の画像のフレームデータを格納する外部メモリとして画像処理 LSI とデータバスなどを介して接続される（図１(a)）．画像処理 LSI は，データが必要な画像処理ごとにフレームメモリにアクセスして画像データを取得し，必要に応じて書き戻す．フレームメモリ容量削減を目的とした画像圧縮回路は，通常画像処理 LSI に内蔵され，画像データをフレームメモリに格納するときに圧縮し，画像処理を行う前に伸張する（図１(b)）．そのために用いる圧縮手法には以下の要件が必要となる．

a） 高画質性（準可逆性，視覚的ロスレス性）
画像処理の多くは補正処理など画質向上を目的としているためメモリアクセス時の画質劣化が補正効果を損なうことは望ましくない．そのため，画像圧縮は他の要件が許す範囲で画質劣化を抑える必要がある．

b） 圧縮率の保証
ハードウェアとして実装されるフレームメモリ容量は限られているため，どのような種類の画像であっても最悪のターゲット圧縮率が保証されている必要がある．



(a) 画像処理LSIとフレームメモリの構成　　(b)　フレームメモリ用画像圧縮

図１　画像圧縮を用いたフレームメモリ容量削減
Figure1　Frame memory reduction by image compression

c） リアルタイム性
メモリアクセスごとに伸長処理と圧縮処理にかかる時間がオーバーヘッドとして付加されるため，できるだけ圧縮処理や伸長処理に必要な演算量（クロックレイテンシ）が小さいことが望ましい．

d） アルゴリズムの単純性（ハードウェアのローコスト性）
フレームメモリのコスト削減が目的であるため，データ圧縮伸長回路の回路コストがそれを上回るようでは意味がない．そのためにはアルゴリズムを回路実装したときのハードウエアコストが小さい単純なアルゴリズムが必要である．

e） 圧縮処理のローカル性（ランダムアクセス性）
表示個所の一部更新など画面の一部を処理するときに，フレームデータ全体を参照するのではなく，画像の指定された画素の復元のためにメモリのどの範囲の圧縮データを参照すればよいか対応が取れることが必要である．また，このアクセス範囲はできるだけ必要最低限であることが望ましい．

f） 画像データアクセス時のスキャン方向対応性
スマートフォンなどの画像フレームのデータアクセスは，端末の回転や使用するアプリケーションよりスキャン方向が縦横に変化する．このような画像データ読み書き時のアクセス順序の変化にも対応できることが望ましい．

3. 既存の画像圧縮手法

画像データ圧縮手法には，大別して，圧縮前のデータと圧縮・伸張後のデータが完全に一致する可逆圧縮手法（ロスレス圧縮手法）と，圧縮前のデータと圧縮・伸張後のデータが完全には一致しない非可逆圧縮手法がある．

画像の可逆圧縮手法には，国際標準である JPEG-LS[1]や JPEG 2000[2]のロスレスモードなど，圧縮効率の良いさまざまなアルゴリズムが考案されている．しかし，これらのアルゴリズムは可逆と言う性質上，シャノンの情報源符号化定理によって定まる下限未満の固定値を保証することは不可能である．また，通常演算量が多くリアルタイム処理が困難であるため，フレームメモリ容量の削減を目的とした圧縮手法としては適さない．

一方，画像の非可逆圧縮手法にも多くのアルゴリズムが考案されている[3]-[14]．非可逆圧縮手法では圧縮・伸長処理による画質劣化の少なさがひとつの性能指標となる．画像の客観的な評価尺度には，圧縮前の原画像データと，圧縮して伸長した後の画像データの誤差をノイズと見なしてピーク SN 比を求めた PSNR（Peak Signal to Noise Ratio）指標がよく用いられる．PSNR 値は大きいほど圧縮による劣化が少なく，PSNR が35dB 以上の自然画であれば人間は視覚的に画質劣化を感じないと言われている．このような条件を満たす視覚的に劣化がない非可逆圧縮は視覚的にロスがない（Visually Lossless）可逆もしく

は準可逆（Near-Lossless）の画像圧縮と呼ばれる．

一般に，非可逆圧縮では圧縮率と画質の間にトレードオフが存在する．圧縮によってより多くの情報を失う可能性がある高い圧縮率をターゲットとする圧縮アルゴリズムでは，もとの画質を保持するのがより困難だからである．また，それをできるだけ補おうとすると圧縮のための処理量が増大し，処理のリアルタイム性やハードウェアのローコスト性を満たせなくなる．DCT（Discrete Cosine Transform）を使った国際標準の JPEG[3]や JPEG 2000の圧縮率は1/10～1/100と言われているが，このような高い圧縮率をターゲットとした圧縮アルゴリズムは処理量も多く，画質的にもブロックノイズやモスキートノイズと呼ばれるノイズが発生し視覚的にも劣化が発生するため，フレームメモリ用圧縮アルゴリズムには適さない．

対象とするアプリケーションにも依存するが，通常，フレームメモリ容量削減を対象とした準可逆圧縮手法は1/2～1/4の圧縮率をターゲットとして設定される．予測符号化を用いた画像圧縮の代表格である差分パルス符号変調 DPCM（Differential Pulse Code Modulation）と量子化を用いたピクセル単位の固定長圧縮[4][5][6]は1/2圧縮などの低い圧縮率のフレームメモリ容量削減アルゴリズムとしては良い結果を示す．筆者らが以前に提案した DPCM ベースの固定長圧縮[5][6]は簡単なアルゴリズムでリアルタイム性を持った準可逆な圧縮手法である．しかし，ピクセル単位に1/2より大きい圧縮率をターゲットにする場合は差分値の量子化係数の増大により急激に画質劣化が無視できないレベルに大きくなる．このようなとき，可変長符号化の導入によって圧縮伸長後の画質の向上が期待できる[7]．可変長符号化は圧縮が得意な画像領域のビット配分を減らすことによって，圧縮が不得意な画像領域により多くのビットを配分し，全体での画質を底上げできるからである．しかし，可変長符号化では原画像の画素位置とメモリ上の圧縮データの格納位置の関係がデータによって変化する．可変長符号化の適用範囲を小さくすると上記メリットの恩恵を受けない反面，画像フレーム全体やライン単位など広い範囲で適用すると圧縮処理のローカル性が損なわれ，１画素のランダムアクセス時にもフレーム全体やライン全体を復号しないといけなくなる．

上記の経緯を踏まえ，圧縮率1/3～1/4をターゲットとするフレームメモリ圧縮は小さいブロック単位の固定長符号化による圧縮を採用することが多い[8]-[14]．ブロック単位の画像圧縮は，①近傍に似た色の画素が統計的に多く存在する傾向がある，②狭い領域内に色の大きく異なる画素が多く存在する場合には圧縮誤差が大きくても視覚的に目立ちにくい，という画像特有のエントロピーの偏りと視覚的認知の性質を利用したデータ圧縮手法である．ブロック単位の固定長符号化は，圧縮後の符号化データがブロックごとに決まったメモリ位置に格納されるため，圧縮処理のローカル性（ランダムアクセス性）や画像データアクセス時のスキャン方向の変化にも対応できる．

BTC（Block Truncation Coding）は画像を小ブロックに分割し各ブロックを２つの値で

近似する古くから使われる画像圧縮手法である．1970年代末に Delp と Mitchell により白黒濃淡画像の圧縮手法として開発され[8]，カラー画像への対応[9]などいくつもの改良が試みられている． 例えば文献[10]は４×４ブロック単位での BTC をベースにベクトル量子化を組み合わせた手法である．しかしながら，BTC も1/2以上の圧縮率をターゲットとする場合には良い結果が得られていない．

文献[11]は４×４ブロック単位で可変長符号化を適用した固定長圧縮手法である．選択したブロック内画素のスキャン順序に従って DPCM を用いて得られた画素差分をブロックごとに目標圧縮率を満たすように量子化係数を調整しながら可変長符号化を繰り返す．

文献[12]は DXTC（DirectX Texture Compression）は S3 Graphics 社の開発した画像圧縮技術である．S3TC とも呼ばれる．Microsoft 社の DirectX 6.0にも採用され，さまざまな家庭用ゲーム機や３G グラフィックスカードで使われる．DXTC は４×４ブロック内の16画素から２色の代表色を選び，代表色ペアを両端点とする色空間での直線上の内分点を求めて中間色を作り，ブロック内の画素を代表色と中間色に縮退して表現する．DXTC の基本アルゴリズムでは代表色と中間色で表せる色しか表現できないため，アニメや CG 画像などによく現れるような，ブロック内に多くの色を含み，選択された代表色では近似できない色味の画素が含まれていた場合にはどうしてもその画素は全く異なる色にマッピングされてしまうという欠点がある．

HDMI，MIPI，DisplayPort などの表示インターフェース規格の標準化団体である VESA（Video Electronics Standards Association）が策定した規格には準可逆の画像圧縮が採用されている．2014年に最初に発表された DSC は改良され DSCv1.2が HDMI2.1や MIPI などに採用されている[13]．2018年に VDC-M が発表され今後の MIPI 規格に採用予定である[14]．RGB 8bit 画像の場合，DSC は1/3，VDC-M は1/4を圧縮ターゲットとしている．これらの画像圧縮は表示デバイスのインターフェース部に実装され表示データ量を圧縮することを目的として開発された画像圧縮である．圧縮率ターゲットが1/3～1/4の視覚的ロスレスであることや回路化して表示制御用 LSI に実装することを前提としている点など，フレームメモリ向き画像圧縮と共通の特徴があり，これらの手法をフレームメモリ圧縮にも流用できる．しかし，スライスと呼ばれる大きなまとまりを単位とした可変長圧縮の仕組みを取り入れることによって圧縮性能を向上しており，ランダムアクセス時にオーバーヘッドが予想される点は弱点である．

4. 提案手法

4.1 基本アイデア

前述した観点から，本稿では，前記要件を満たす４×４ブロック単位で圧縮率1/3～1/4をターゲットとしたフレームメモリ向け画像圧縮手法を提案する．この目的を満たすため，

提案手法ではブロック内の16画素を内包するRGB直方体空間を算出し，その色空間のRGB階調をサンプリングして得られた候補色の集合に対して，各画素の色をそれぞれ最も色味の近い候補色にマッピングして縮退することによりデータ量を削減する．本手法は上記色空間領域指定とインデックスに割り当てるビット数の配分の調整によって本質的にはスケーラブルであるが，本稿ではRGB 8bit画像を1/3圧縮する場合に基づいて説明する．４×４画素ブロックのRGB 8bit画像は圧縮をしない場合 $8(bit) \times 3(color) \times 16(pixel) = 384$ bit で表現できる．1/3圧縮では提案手法ではこれを図２に示す $384/3 = 128$ bit の固定長パケットに変換する．

図２において，$Mode$ は後述する圧縮モード指定のために使用する．$(R_{max5}, G_{max5}, B_{max5})$ と $(R_{min5}, G_{min5}, B_{min5})$ はそれぞれ16画素のRGB階調値の最大値と最小値を5bitに量子化したものである．この２点により16画素の色を内包するRGB直方体空間を生成でき，これが対象となる色空間領域となる．$P1$~$P16$ は原画像の16画素に対する候補色へのマッピングのインデックスである．インデックスのビット数が6bitの場合，それぞれ $2^6 = 64$ 個の候補色へのマッピングを選択できる．

これまでの基本アイデアの説明から明らかなように，ブロック内の画素の各色の階調値のMAX-MIN範囲が色空間領域を決定し，インデックスのビット数がマッピングできる

$Mode$	R_{max5}	G_{max5}	B_{max5}	R_{min5}	G_{min5}	B_{min5}	$P1$	$P2$	…	$P16$
$2bit$	$5bit \times 3$			$5bit \times 3$			$6bit \times 16$			

$2bit + 5bit \times 3 + 5bit \times 3 + 6bit \times 16 = 128bit$

図２　固定長パケットデータ

Figure2　Packet data of fixed length

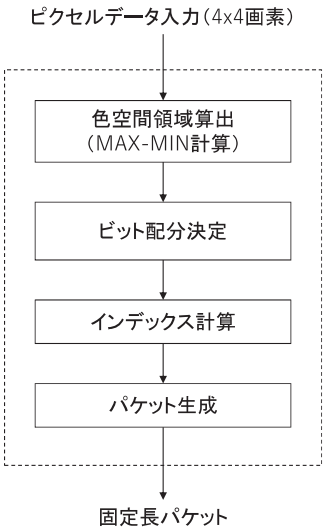


図３　提案手法の基本圧縮処理フロー

Figure3　Basic compression flow of proposed method

候補色数を決定する．従って，各色の MAX-MIN 範囲が狭ければ狭いほど，もしくは，インデックスのビット数を大きく取れるほど, サンプリングを細かくでき, 候補色へのマッピングにより生じる圧縮誤差を小さくできる可能性がある．そのため，提案手法では，対象ブロック内画素の色分布の性質を解析し，基本アルゴリズムの枠組みを利用して圧縮誤差を小さくする追加処理を行う圧縮モードを導入して，画質劣化を抑える．

提案手法の基本アルゴリズムの大まかな符号化フローは図３のようになる．次節以降では，最初に基本的な処理フローでの各処理ステップの詳細について述べ，次に圧縮誤差を小さくする圧縮モードの導入について述べる．

4.2 色空間領域の算出

ブロック内の16画素について RGB 成分の各階調値をスキャンし，RGB 成分それぞれについて階調値の最大値と最小値を求める．階調値は必要に応じてパケットに格納するためのビット長に量子化する．図２の例のように8bit 階調の RGB 成分を持つ画素のそれぞれの RGB 値 R_8, G_8, B_8 に対して5bit の値 R_5, G_5, B_5 をパケット内に格納する場合には $(R_{max5}, G_{max5}, B_{max5})$ と $(R_{min5}, G_{min5}, B_{min5})$ の値はそれぞれ $X_5 = X_8 \gg 3\ (X = R\ or\ G\ or\ B)$ によって算出する．このとき，例えば8bit 色空間上での16画素を内包する RGB 直方体空間 U は

$$U = \{(R_8, G_8, B_8) | (R_{min5} \ll 3) \le R_8 \le \big((R_{max5} + 1) \ll 3\big) - 1, (G_{min5} \ll 3) \le G_8 \le \big((G_{max5} + 1) \ll 3\big) - 1, (B_{min5} \ll 3) \le B_8 \le \big((B_{max5} + 1) \ll 3\big) - 1\}$$

で定義できる．但し，本ステップで算出する色空間領域は16画素をマッピングする64の候補色の境界を決定するためだけのものであるので，必ずしも上記式に正確に従う必要も16画素の色を完全に内包する必要もない．

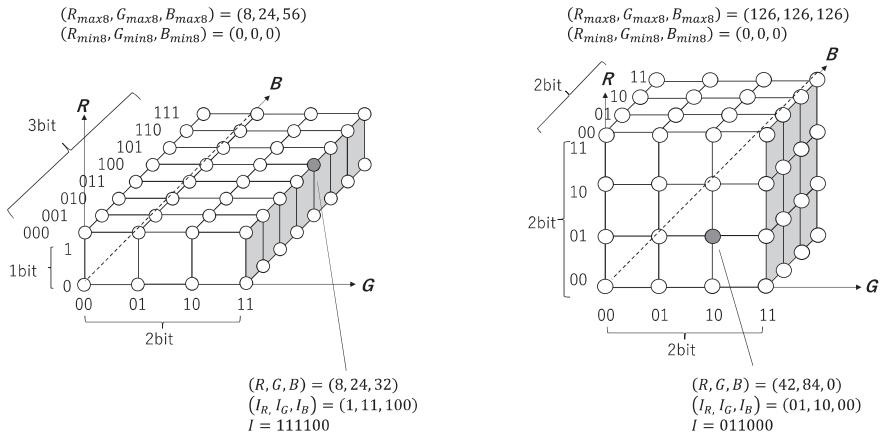
4.3 ビット配分による候補色生成

このステップでは，前ステップで求めた RGB 直方体空間の各 RGB 座標をサンプリングしてインデックスのビット長で表現できる数の候補色を生成する．このとき，原画像のブロック内の16画素の色分布に関する情報がない状態では，色空間領域上のどの領域に画素色が存在するかはわからないため，候補点の色空間上の間隔ができるだけ等しくなるように配置する．

例えば，色空間領域が $(R_{min8}, G_{min8}, B_{min8}) = (0,0,0)$，$(R_{max8}, G_{max8}, B_{max8}) = (8,24,56)$ で表されるとき，RGB をそれぞれ８階調間隔でサンプリングすると，R={0, 8}, G={0, 8, 16, 24}, B={0, 8, 16, 24, …, 56} が階調値の候補値となる．このとき，RGB の階調値の候補数はそれぞれ，２値，４値，８値であるので，RGB をそれぞれ1bit, 2bit, 3bit のインデックスを用いて6bit 長で表現でき，表現できる候補色は，$R \times G \times B = 2 \times 4 \times 8 = 2^6 = 64$ 通

りとなる. もし, 例えば色空間領域が $(R_{min8}, G_{min8}, B_{min8}) = (0,0,0), (R_{max8}, G_{max8}, B_{max8}) = (126,126,126)$ の場合のように RGB 範囲がほぼ等しい場合には RGB のすべてに2bit のインデックスを用いて表現すれば各 RGB 画素成分を $\{X_{min}, (2 \times X_{min} + X_{max})/3,\ (X_{min} + 2 \times X_{max})/3,\ X_{max}\}\ (X = R\ or\ G\ or\ B)$ の 4 つをそれぞれ RGB 成分とした $R \times G \times B = 4 \times 4 \times 4 = 2^6 = 64$ 通りの候補色にマッピングできる. 上記それぞれの色空間イメージは図 4 のようになる.

このように, 本ステップでは色空間領域の RGB 各色の範囲 $R_{range} = R_{max} - R_{min}$, $G_{range} = G_{max} - G_{min}$, $B_{range} = B_{max} - B_{min}$ の比によってインデックスの RGB のビット配分を決定する. RGB の各色の範囲の比を $r_1, r_2, r_3\ (r_1 \leq r_2 \leq r_3)$ として bit 配分 $b_1, b_2, (b_1 \leq b_2 \leq b_3)$ を求める. インデックスの bit 長が6bit $(b_1 + b_2 + b_3 = 6)$ の場合, b_1, b_2, b_3 の配分は 7 通り考えられる. 1 つの bit 値で表現できる最大範囲が RGB でできるだけ等



(a) $(R_{range}, G_{range}, B_{range})$の比 $= (1,3,7)$の場合の例　　(b) $(R_{range}, G_{range}, B_{range})$の比 $= (1,1,1)$の場合の例

図４　色空間領域と候補色

Figure4　Color space region and candidate colors

bit	range比			bit配分		
	r1	r2	r3	b1	b2	b3
6bit	1	1	64	0	0	6
	1	2	32	0	1	5
	1	4	16	0	2	4
	1	8	8	0	3	3
	2	2	16	1	1	4
	2	4	8	1	2	3
	4	4	4	2	2	2

図５　各色の範囲比とビット割り当て（6bit インデックスの場合）

Figure5　Color range ratio and bit allocation (in case of 6bit index)

しくなるように決定すると，例えば，各色の範囲比（の閾値）とビット配分との対応は図５の通りとなる．

4.4 インデックス計算

このステップでは，前ステップで決定したビット配分に従い，各画素の階調値からRGB のそれぞれの成分についてインデックスを計算する．今，ある色に対して色空間での RGB 画素の階調値の範囲が $X_{min} \le X \le X_{max}$ $(X = R\ or\ G\ or\ B)$ で，かつ，この色に対するインデックスのビット配分結果が M bit であったとする．このとき，M bit のインデックスで表現できるのは 2^M 通りである．そこで区間 (X_{min}, X_{max}) を $(2^M - 1)$ 個の範囲に内分した階調値

$$X_n = (n \times X_{max} + (2^M - 1 - n) \times X_{min}) \,/\, (2^M - 1)\ \ (0 \le n \le 2^M - 1)$$

をその色の候補色の階調値とするとちょうど両端点を含め 2^M 個の点でサンプリングできる．n を２進表現したものが対象とする候補色の RGB 成分のインデックスになる．計算対象である画素の色成分の階調値が X のとき，X が最も近い階調値を表すインデックス m は，上記 $X_0, X_1, X_2, \cdots, X_{2^M-1}$ によって分割された $(2^M - 1)$ 個の区間のそれぞれをさらに中点で２分割してできる $2(2^M - 1)$ 個の何番目の区間に X が含まれるかを算出してそこから最も近いインデックスの番号に変換すれば以下の計算式で直接求められる．図６に $M = 3$ のときのインデックス計算のイメージを示す．

$$m = \left\lfloor \left(\left((X - X_{min}) \times 2(2^M - 1) \right) / (X_{max} - X_{min}) + 1 \right) / 2 \right\rfloor$$

この処理は各色独立で行えるため，対象画素それぞれの RGB 各色に対するインデックスの２進表現 I_R, I_G, I_B を求め，これらの２進表現を連接することにより対象画素に対するインデックス I は以下の式の通り求められる．

$$I = CONCATENATE(I_R,\ I_G,\ I_B)$$

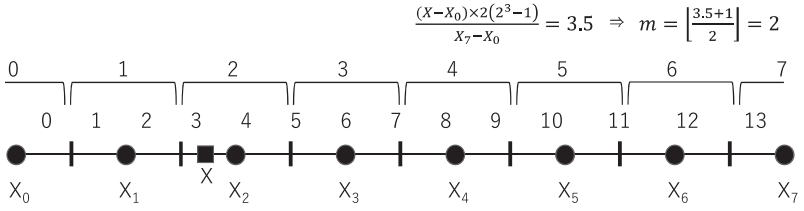


図６　最も近いインデックスの計算（M=3）

Figure6　Calculation of the nearest index（M=3）

4.5 パケット生成

8bitRGB 画像の1/3圧縮では４×４画素ブロックに対するパケット長は128bit であり，図２で示した構成を持つ．前節までの基本処理で得られた MAX-MIN による RGB 色空間領域定義（②）とブロック内の16画素に対するインデックス（③）に格納される．基本アルゴリズムでは①の情報は使わないため，次節以降で説明を加える．

①圧縮モード（2bit）

②色空間領域定義（5bit ×３×２）

③インデックス（6bit ×16）

提案手法はスケーラビリティを備えた手法である．ターゲットとする圧縮率や元画像の画素のビット数が異なる場合，サンプルデータを用いた評価を行い，パケットのターゲットビット長に収まるように色空間やインデックスのビット長を調整することによって，スケーラブルにアルゴリズムを構築できる．

4.6 圧縮モード選択（１） ー YUV モードー

前節までは，RGB 8bit 画像の例に基づいて基本処理フローを説明してきたが，これまでの説明で明らかな通り，基本処理フローは RGB 色空間にのみ適用できるのではなく，汎用的にどのような色空間にも適用できる．符号化対象となる４×４ブロックが，もし RGB 色空間では MAX-MIN 範囲が広くなる場合に領域を狭くできる特徴を持つ色空間表現を見つけることができれば，サンプリング間隔を小さくでき，画質劣化をより抑えると期待できる．

今回の提案手法では，このような一例として，JPEG2000の可逆圧縮モードで採用されている RGB-YUV 変換式（図７）を用いて，前処理で YUV 色空間に変換してから同じアルゴリズムを用いて色空間領域とインデックスを生成し，処理モードヘッダにどちらのモードで符号化を行ったかを記録する．なお，YUV 色空間では Y 成分（輝度成分）が視覚的により重要な働きをするため, ビット配分において Y 成分に重みを与え, UV 成分（色

$$\begin{pmatrix} Y_r \\ U_r \\ V_r \end{pmatrix} = \begin{pmatrix} \left\lfloor \dfrac{R+2G+B}{4} \right\rfloor \\ R-G \\ B-G \end{pmatrix}$$

$$\begin{pmatrix} G \\ R \\ B \end{pmatrix} = \begin{pmatrix} Y_r - \left\lfloor \dfrac{U_r+V_r}{4} \right\rfloor \\ U_r+G \\ V_r+G \end{pmatrix}$$

図７　RGB-YUV 変換式
Figure7　RGB-YUV conversion formula

差成分）より細かいサンプリングを行える工夫をしている．

4.7 圧縮モード選択（2） －グラデーションモード（GRAD）－

4×4の局所的なブロックでは，グラデーションなど，ブロック内の16画素すべての画素が類似の色味で変動する場合も多い．そのような性質のブロックの場合にはRGBに対するインデックスを3色別々に指定してマッピングするのではなく，1つのインデックスで連動させてマッピングする方が通常モードより細かい階調表現が期待できる．提案手法では，本目的のためにグラデーション（GRAD）モードをサポートする．GRADモードではパケットに色指定フィールド$color_sel$=（b, g, r）(3bit）を追加し，指定した（該当bitが1である）色を同一のインデックスで連動させて変化させ，他の（該当bitが0である）色は固定する．ある色を変化させるか固定させるかは閾値$GRAD_TH$を用いて判定し，MAX-MIN範囲が$GRAD_TH$以上である色を変化させる．通常モードと異なり，GRADモードの色空間領域は2つの色$(C1, C2)$を結ぶ色線分となる．例えば，$color_{sel} = (001)_2$のときブロック内の全画素を$(R_1, G, B) - (R_2, G, B)$色線分上の32階調の候補色で表現する（インデックスのビット長が5bitの場合）．同様に，例えば，$color_{sel} = (011)_2$のときにはブロック内の全画素を$(R_1, G_1, B) - (R_2, G_2, B)$色線分上の32階調の候補色で表現し，$color_{sel} = (111)_2$のときにはブロックの全画素を$(R_1, G_1, B_1) - (R_2, G_2, B_2)$色線分上の32階調の候補色で表現する．但し，すべての色成分でMAX-MIN範囲が閾値$GRAD_TH$より小さくなった場合には最も視認度が高いGを優先し$color_{sel} = (010)_2$とする．YUVモードと併用するときは，RGBの代わりにYUVを同様に連動させ，すべての色成分でMAX-MIN範囲が閾値$GRAD_TH$より小さくなった場合には最も視認度が高いYを優先する．

GRADモードを圧縮誤差が小さく効果的に使用できる対象ブロックは限定的であるが，GRADモードを適用できるブロックに関しては，インデックスを1つに共通化して連動させることによって同じビット長のインデックスで通常モードより細かい階調表現が可能となり，大幅な画質向上が期待できる．また，基本的な処理は通常モードとはそれほど変わらないため，通常モードのハードウェアとの回路共有ができ，モード追加による回路規模増加はそれほど大きくない．

4.8 圧縮モードの選択（3） －空間マッピング（SP）－

3番目の工夫は空間方向の縮約によるインデックスビットの拡張である．一般に画像では隣接する画素に似た色が多い傾向があるため，4×4ブロックのうち半分の8画素を通常のインデックスによるマッピングを行う代わりに，隣接する画素の色を参照することによって生成し表現する．例えば，ある画素につき4通りの参照パターンから選ぶ場合には2bitで参照パターンを表すことができるため，空間マッピング対象の8画素のインデックスを6bit→2bitに節約することができる．このため，通常マッピングする残りの8画素の

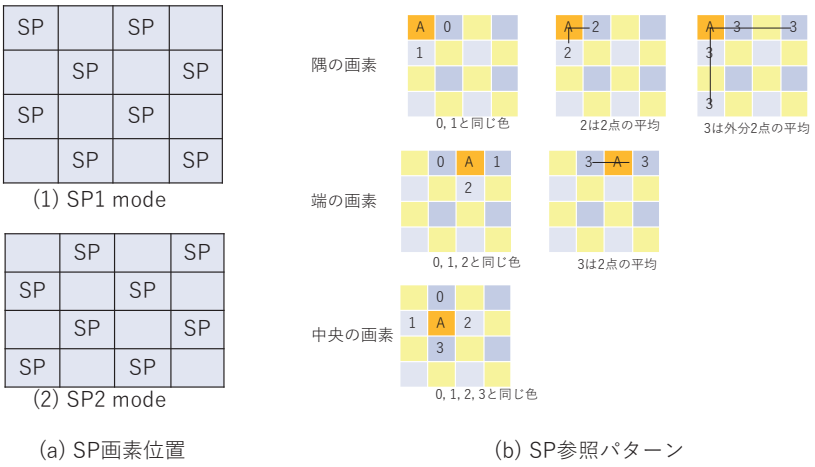


図８　空間マッピング（SP mode）
Figure8　Space mapping（SP mode）

インデックスを6bit →10bit に増やすことができ，MAX-MIN 範囲が広くなるブロックに対してサンプリング間隔が大きくなることに起因する圧縮誤差の抑制に効果的であると考えられる．提案手法では，空間マッピングする画素位置に周囲の参照画素では表せない特徴的な色が存在することを避けるために，２通りの空間マッピングモード（SP1と SP2）を採用する．

図８(a)に２通りの SP マッピングモードの画素位置を示す．参照画素の４通りの SP 参照パターンは画素のブロック内位置が隅にある場合，端にある場合，中央にある場合で切り替える．図８(b)に提案手法で実装した SP 参照パターンを示す．選択した SP 参照パターンによっては隣接画素と単純に同じ色で表現するのではなく，グラデーションなどを考慮して２つ以上の画素の色の平均などの値を取る．

あるブロックの圧縮に空間マッピングを採用したかどうか，SP1と SP2のどちらのマッピングモードを使うか，は圧縮モード（*mode*）により指定する．空間マッピングは基本アルゴリズムの単純な拡張ではないため，アルゴリズムが複雑となり回路コストも増加する．マッピングモードや参照パターンは種類が多いほど各ブロックに適したものが見つかる可能性があるが，マッピングモード指定のため圧縮モードのビットフィールドが長くなること，空間マッピングによる回路規模増加とのトレードオフのため，今回の提案手法では最低限のパターン数を採用する．

4.9 提案手法のパケットと処理フロー

オプションモードを含めた提案手法全体の固定長パケットの bit 割当ては図９の通りとなる．通常モード（Normal），SP モード（SP1/SP2），GRAD モード（GRAD）は *Mode* フィールド2bit の（カッコ内に記載した）値によって判定され，その他の箇所も連動して

異なる bit 割り当てとなる．dum は128bit 固定長に調整するためのダミー bit である．SP モードでは，画素 $P1$~$P8$ が SP 参照画素となり，画素 $P9$~$P16$ が通常モードと同じインデックスで表現される．GRAD モードのときは $COL(color\_sel)$ のビットが 1 である数（連動させる色数）で Grad1, Grad2, Grad 3 のモードに分類され，固定色の階調値は 2 番目の色フィールドは省略される．$C1, C2, C3$ がどの色に対応するかは COL 値に依存して決定する．基本アルゴリズムでは C_1 と C_2 の大小関係は $C1_1 \geq C1_2 (MAX \geq MIN)$ 固定であったが，最終的なパケットでは図 9 で(*)で示されている場合には $C1_1 \geq C1_2$ のとき RGB モード，$C1_1 < C1_2$ のとき YUV モードに制御する．$C1_1 = C1_2$ かつ YUV モードでの圧縮が最適な場合には C_1 の MAX-MIN 範囲を $C1_1 = C1_2$ でないように変更してから $C1_1 < C1_2$ となる

モード	*パケットデータのbit割当て*															
	$Mode$	COL	YUV	$C1_1$	$C2_1$	$C3_1$	$C1_2$	$C2_2$	$C3_2$	$P1$	…	$P8$	$P9$	…	$P16$	dum
Normal	2(01)	-	(*)	5	5	5	5	5	5	6	…	6	6	…	6	0
SP1	2(10)	-	(*)	5	5	5	5	5	5	2	…	2	10	…	10	0
SP2	2(11)	-	(*)	5	5	5	5	5	5	2	…	2	10	…	10	0
Grad1	2(00)	3	1	8	8	8	8	-	-	5	…	5	5	…	5	10
Grad2	2(00)	3	1	8	8	8	8	8	-	5	…	5	5	…	5	2
Grad3	2(00)	3	1	7	7	7	7	7	7	5	…	5	5	…	5	0

図９　提案手法のパケットデータ

Figure9　Packet data of proposed method

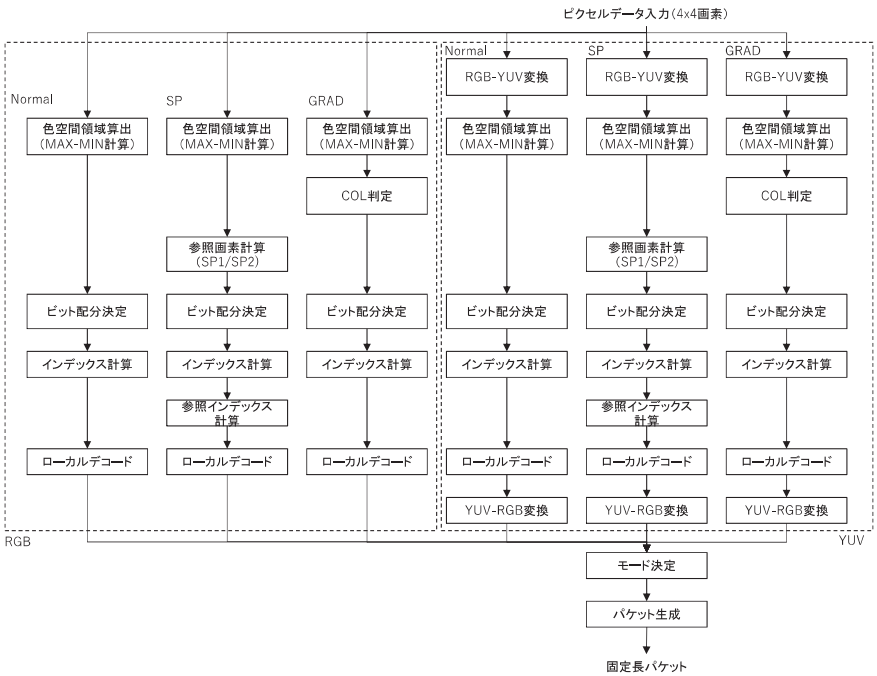


図10　提案手法の圧縮処理フローチャート

Figure10　Compression flowchart of proposed method

ようにパケットに格納する.

提案手法の処理フローを図10に示す．前節までで説明したすべての圧縮モード処理は，できるだけ並列に処理を行い，ローカルデコードによって得られた伸長後の階調値と元の画素の階調値とからPSNR値を計算し，もっとも結果が良いモードの結果を選択する．ローカルデコードした階調値はいずれの処理モードでもMAX，MIN，インデックスを用いた単純な内分点計算で求められる．図10ではSP1とSP2はまとめて記載しているが実際には別々に処理をするため，全体で８つのモードの処理を行うことになる．共通する処理ステップのアルゴリズムや計算回路は類似しているため共有し，これらのステージをモードごとにパイプライン処理で行うよう実装できる．モード決定のためのPSNR計算も同様に同一回路で共有できる．各モードのYUVモードは前処理と後処理にそれぞれRGB-YUV変換とRGB-YUV変換を追加するだけで実現可能である．GRADモードも色空間領域指定に対するマッピングを連動させる仕組みを追加することで基本アルゴリズムや処理回路を共有化することが可能である．SPモードはSP参照画素に関して参照モードを決定し画素を計算する処理と参照インデックスを生成する処理や回路実装が追加で必要となる．

5. 評価実験による提案手法の評価

5.1 提案手法の評価

本節では，いくつかの評価データを用いて提案手法の評価を行う．これらの画像について圧縮前の画像と提案手法による圧縮伸長後の画像のデータを比較し，PSNR値を計算した．まず，提案手法のアルゴリズムをC言語プログラムにより実装し，性能確認用の動作モデルを作成した．プログラムはデバッグ用や解析用のコードを含めても約2,000行程度のコード量で記述できた.

評価画像には，画像処理用標準画像データベースSIDBA（Standard Image Data-Base）[15]からよく知られている女性画像（Lena, LENA），マンドリル画像（Mandrill, MAND）に加え，解像度チャート（RESO），15枚の自然画像（breakfast，catなど），CGフラクタル画像（FRAC），をインターネット上のフリー画像素材より選択して使用した．自然画像はさまざまな特徴の画像を評価できるよう，できるだけ高周波成分や低周波成分がバランス良く多く含まれた画像を選択した．図11に使用した評価画像の一覧を挙げる.

表１に実験結果を示す．左欄は提案手法のPSNR評価結果である．(a)は基本アルゴリズム（RGBモード）のみの場合，(b)-(d)は(b)YUVモード，(c)GRADモード，(d)SPマッピングを順に重ねて追加したときの結果である．中央欄は(d)の場合に，ブロックごとに最適として採用された処理モードの適用率（%）を示す．右欄に(a)のときと(d)のときでのそれぞれの元画像と圧縮伸長画像のRGB画素の階調の平均と最大の差分の比較を示す.

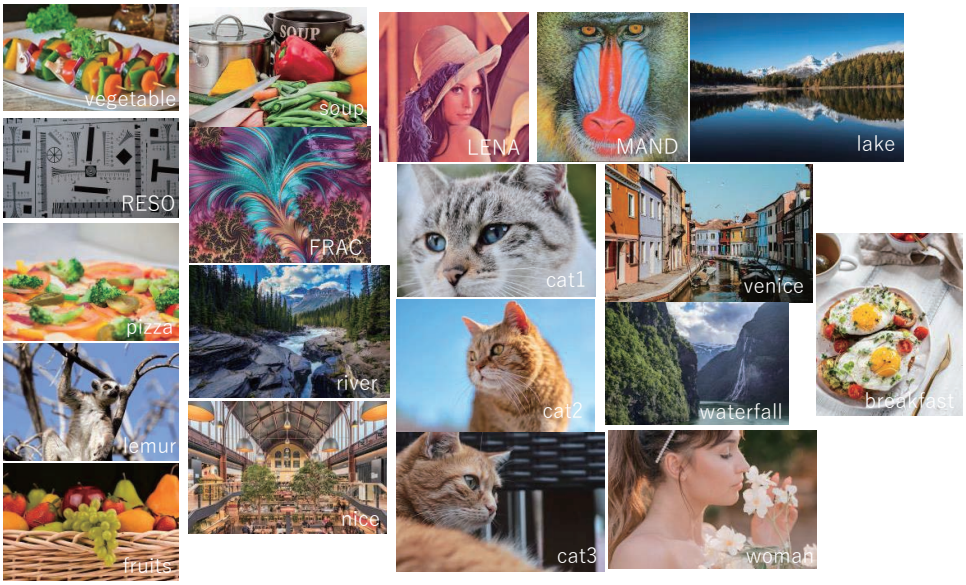


図11　評価画像
Figure11　Evaluation Images

表１の結果の通り，(a)の場合には一部の画像の PSNR 指標は35dB を達成できなかったが，すべての圧縮モードをサポートする(d)の場合ではすべての画像で35dB を達成し，平均で40dB に迫る結果が得られた．また，目視による主観評価でも視覚的に画質の劣化が認められなかった．

実験では，いずれの画像も(b)YUV モードを追加サポートしたときに大幅な PSNR 値

表１　提案手法の評価結果
Table1　Evaluation results of proposed method

画像データ		圧縮前後のPSNR値(dB)				処理モードごとの適用率(%) (d)の場合								圧縮前後のRGB階調の差分			
	モード	(a)	(b)	(c)	(d)									(a)		(d)	
	RGB	○	○	○	○												
	YUV		○	○	○												
	GRAD			○	○	RGB				YUV							
	SP				○	Normal	GRAD	SP1	SP2	Normal	GRAD	SP1	SP2	平均	最大	平均	最大
FRAC		29.29	34.75	34.76	35.12	9.46	0.69	10.23	9.35	61.99	0.44	4.16	3.67	6.18	41	3.29	51
LENA		37.43	38.84	38.84	39.36	64.74	0.01	7.35	7.24	14.89	0.00	2.51	3.25	2.22	35	1.83	52
MAND		32.87	37.00	37.00	37.76	6.18	0.07	17.62	17.01	52.54	0.04	3.01	3.51	4.24	37	2.43	34
RESO		35.94	42.27	42.99	43.28	16.04	33.19	6.92	6.94	30.35	2.43	2.06	2.08	2.20	27	1.11	18
breakfast		39.71	41.70	42.22	43.14	21.66	16.21	19.34	19.96	14.19	5.71	1.39	1.54	1.74	41	1.13	49
cat1		33.06	39.71	40.12	40.85	5.50	10.23	16.31	14.50	48.14	3.84	0.81	0.66	3.62	41	1.59	25
cat2		37.07	39.17	41.60	42.54	7.27	17.95	14.49	11.82	11.62	34.45	1.45	0.95	2.44	33	1.02	26
cat3		35.73	38.80	39.87	40.90	6.35	19.92	22.14	14.97	20.31	14.50	1.03	0.79	2.74	36	1.38	27
fruits		35.17	37.84	38.58	40.07	11.51	16.18	19.13	19.35	14.08	6.91	6.14	6.69	2.91	40	1.55	31
lake		36.35	39.25	41.51	42.05	7.50	29.77	10.50	7.93	17.61	24.48	1.20	1.01	2.55	41	1.13	37
lemur		35.19	38.57	39.42	40.43	12.05	11.71	20.68	16.72	19.92	15.94	1.44	1.55	2.82	41	1.45	32
nice		29.39	35.45	35.47	35.93	8.68	1.40	12.23	11.48	55.57	0.51	5.18	4.95	6.01	41	2.99	42
pizza		37.34	39.19	40.35	41.43	14.61	15.50	19.53	17.26	9.69	16.19	3.65	3.56	2.30	35	1.25	43
river		31.08	35.98	36.00	36.41	11.86	1.50	12.30	12.12	59.38	0.27	1.33	1.25	5.09	41	2.89	58
soup		33.06	38.18	38.45	39.25	9.71	10.03	19.37	15.90	31.70	5.62	3.67	4.00	3.43	41	1.75	44
vegetable		35.53	38.35	38.73	39.84	16.76	7.39	21.97	19.73	17.04	7.68	4.62	4.82	2.73	39	1.66	52
venice		29.87	36.02	36.33	36.81	9.91	10.64	9.80	9.67	42.56	9.54	4.06	3.82	5.15	41	2.43	45
waterfall		32.54	37.58	37.63	38.04	11.91	2.39	21.30	22.14	39.10	1.60	0.90	0.67	3.82	35	2.18	31
woman		37.72	39.79	41.05	42.37	12.04	19.75	23.47	18.93	11.12	12.37	1.18	1.13	2.18	35	1.20	25
平均		34.44	38.34	39.00	39.77	13.88	11.82	16.04	14.37	30.09	8.55	2.62	2.63	3.39	38	1.80	38

の改善が見られ，(c)や(d)での改善度は画像の性質に依存して高い場合と低い場合があった．しかし，(d)の場合の処理モードごとの適用率を見る限り，すべての処理モードが性能に貢献していることが分かる．GRAD モードの適用はその性質上限定的ではあるが，アルゴリズム検討時の予想通り，RESO，lake，cat2，woman など低周波成分の多い画像領域を多く含む画像で最良なモードとして適用される割合が大きくなることが判明し，追加した圧縮モードの有効性が確認できた．この結果から GRAD モードはアニメ画像や余白の多いドキュメント画像にも効果的であることが予想できる．なお，各処理モードの適用率などの統計情報や RGB 階調の差分が大きいブロックでの適用モードを確認して閾値 $GRAD\_TH$ やパケットのビット配分を調整すれば，結果の改善が期待できる．

圧縮前後の RGB 階調値の差分の平均値は概ね PSNR 評価値と相関性が高いが，最大値は画像によっては基本アルゴリズムのみの場合の方が良くなる場合が存在する．この結果は各 4 × 4 ブロックの結果では PSNR 値が良くなるが，あるブロック内の画素に注目すると誤差が大きくなる箇所が存在することを示している．これは SP モードのマッピングモードが対象ブロックの色のばらつきに十分に適合できていないことを意味している．単純にマッピングモードを増やすことにより改善は期待できるが，前述の通り SP モード選択にさらに多くのビットが必要となりトレードオフになる．さらなる圧縮品質の向上を目指す場合には，このような個所を抑えてゆくことも今後の改善課題であると思われる．

各々の画像に対する結果解析の一つとして，Vegetable の解析結果を示す．図12に示した画像はそれぞれ，(a)元画像，(b)圧縮伸長画像，(c)画素ごとの圧縮誤差階調（強調表示），(d)GRAD モード適用ブロック（白色部分），(e)YUV モード適用ブロック（白色部分），(f)SP マッピング適用ブロック（SP1：白色，SP2：中間色）を示している．

図12(c)では明るい画素ほど程度に応じて元画像と圧縮伸長画像の誤差が大きい．これ

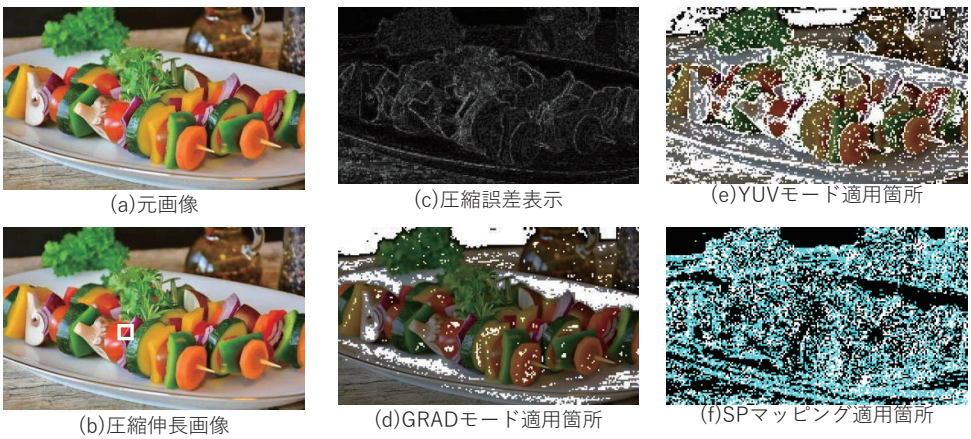
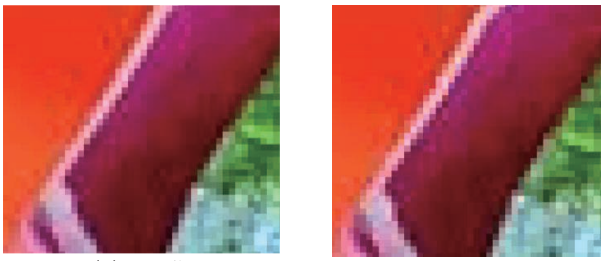


図12　画像 Vegetable の解析
Figure12　Analysis of Image “Vegetable”



(a)元画像　　(b)圧縮伸長画像

図13　Vegetable の拡大画像
Figure13　Zoomed image of “Vegetable”

によると，色の境界が密集し高周波成分が多い部分に誤差が集中している．(d)と合わせて観察すると GRAD モードが適用された平滑な部分はいずれも誤差が小さく抑えられている．RGB と YUV の選択に大きな傾向はみられないが RGB と YUV はバランスよく採用され，特に RGB モードではどちらかの SP マッピングが採用されている傾向が高く，結果的に SP マッピングを採用した方が PSNR による性能が上がることが確認できた．

図12(b)の白い四角枠が圧縮誤差最大の画素がある箇所である．当該ブロックは RGB モードで SP1空間マッピングが採用されて圧縮されている．図13に該当箇所を含む領域を拡大した画像を示す．図では四角く見える各領域が画素に対応している．該当箇所は１，２画素の幅で色が大きく変化した結果，インデックス表現できるサンプリング間隔が荒くなり(b)の圧縮伸長画像では(a)の元画像の滑らかさが損なわれている．しかし，前述した通りこのような周囲に大きな色の変化がある箇所では視覚的感度も鈍くなり，もともとの画素サイズで見ると誤差が画像に及ぼす影響はほとんど知覚できないレベルとなっている．

5.2 他の従来手法との比較

本節では，前節と同じ評価画像データを用いて他の従来手法との PSNR 性能比較を行う．比較対象としてはリファレンスモデルが入手できる DPCM [5]と，VESA の表示インターフェース用画像圧縮である DSC（v1.2）[13]，VDC-M [14]を選択した．

DPCM はフレームメモリ用画像圧縮回路として実際にいくつかの商用製品の LSI に搭載されている非常に単純なアルゴリズムで，10Kgates 未満のコンパクトな回路で1/2圧縮をリアルタイムで実現する手法である．DSC も標準規格である HDMI2.1や MIPI などに採用され回路化されており，圧縮回路の規模は約500Kgates と言われている．今回の提案手法は正確な回路規模は不明だが，見積りでは DSC 同等の回路規模であると予想している．VDC-M は次期 MIPI 規格のために回路化が進められているが，多くのアルゴリズムを実装してベストな結果を採用する構成のため，すべての機能を実装すると DSC よりははるかに大きな回路規模になると予想される．

各手法はターゲット圧縮率や回路規模（アルゴリズムの複雑度）も全く異なるため，一

律の基準では比較できない．しかし，DSC はターゲット圧縮率が今回の提案手法と同じ1/3であり，想定している回路規模もそれほど変わらないと予測できるため，DSC を中心に比較を行った．DSC は表示インターフェース用を想定した画像圧縮のため，フレームメモリ向き画像圧縮と異なり，ランダムアクセス性やスキャン方向対応性の要件を重要視していない．表示パネルが決まるとドライバによるスキャン方向は決まり，通常，画面全体で表示を更新するからである．そのため，DSC では複数ブロックをまとめたスライスという概念を導入し，スライス毎に可変長パラメータを変化させて圧縮性能を優先している．スライスの概念を持たない提案手法は４×４画素ごとに独立に縦横ランダムアクセスが可能であるが，DSC でランダムアクセスをするときにはスライスサイズ単位でのアクセスしかできないという制限が加わる．DSC との比較ではこのスライスサイズのパラメータを変化させて性能を比較した．

表２に PSNR 値の比較結果を示す．DPCM は1/2圧縮であるが，画素単位の圧縮でアルゴリズムも単純なため提案手法に比べ PSNR 性能的には全体的に劣っている結果が出た．VDC-M は1/4圧縮にも関わらず全体的に提案手法より良い結果が出ているが，回路規模とのトレードオフになっていると考えられる．実験では DSC は４通りにスライスの大きさを変更して結果を比較した．DSC は画面全体やライン全体×４を１つのスライスとする場合は提案手法より総じて良い PSNR 値が得られ，スライスサイズを縮小すると PSNR 性能が落ちてくる．512×４のスライス設定では標準画像 LENA，MAND では提案手法の

表２　従来手法と提案手法の比較

Table2　Comparison with conventional methods

画像データ		提案手法	DPCM	DSC1.2	DSC1.2	DSC1.2	DSC1.2	VDC-M
	圧縮率	1/3	1/2	1/3	1/3	1/3	1/3	1/4
	回路規模	中	小	中	中	中	中	大
	スライス	(4x4画素)	-	画面全体	LINEx4	512×4	256×4 (*)	Default
FRAC		35.12	31.64	39.78	38.64	37.83	35.20	46.00
LENA		39.36	32.99	42.59	39.08	39.08	37.71	48.16
MAND		37.76	32.09	44.98	38.79	38.79	38.55	45.82
RESO		43.28	38.91	53.41	50.70	47.12	44.12	51.87
breakfast		43.14	37.81	52.48	51.94	46.69	43.11	51.65
cat1		40.85	34.35	52.51	47.35	46.03	41.87	49.91
cat2		42.54	37.57	52.88	50.52	49.43	43.75	51.87
cat3		40.90	36.35	53.49	49.41	47.06	43.39	48.93
fruits		40.07	37.33	48.39	44.57	44.17	41.24	49.07
lake		42.05	37.49	51.64	49.21	46.73	43.43	52.03
lemur		40.43	35.43	51.17	47.18	45.71	41.95	50.42
nice		35.93	31.76	43.25	39.73	39.88	37.33	46.96
pizza		41.43	37.54	48.58	44.59	44.30	41.42	52.69
river		36.41	31.88	43.27	40.17	40.48	37.93	45.87
soup		39.25	35.76	49.06	44.77	44.39	41.08	50.27
vegetable		39.84	35.96	48.74	44.66	43.73	40.71	48.45
venice		36.81	32.97	45.22	41.09	40.86	37.91	50.16
waterfall		38.04	33.81	48.21	42.96	41.60	39.20	48.45
woman		42.37	36.80	53.57	49.06	47.89	43.38	48.99
平均		39.77	35.18	48.59	44.97	43.78	40.70	49.35

(*)デフォルトの動作モードでは動かないためFIRST_LINE_BPG_OFFSET=4に変更

方が優れた結果が出た．これを256×４にするとさらにbreakfastやniceも提案手法よりPSNR性能が低くなり，256×４より小さいスライス設定にすると可変長圧縮による制約のために簡易的なパラメータの変更だけでは圧縮結果を得ることができなくなった．しかし，例えば，４×４画素をアクセスするために256×４画素をアクセスし復号する必要があるのであれば，せっかくのデータ圧縮効果が伝送データ量に反映されない．以上から，フレームメモリ用の画像圧縮として適用する場合，ある程度大きな単位でのアクセスのみしか行わない画像処理だけに限定すればDSCは良い性能を得ることができるが，より汎用的には，柔軟なランダムアクセスが求められる画像処理が含まれる可能性を考慮すると，４×４ブロックごとに独立に圧縮伸長を行うことができる我々の提案手法の方がより適していると考えられる．

6. あとがき

本稿では，フレームメモリ削減を目的とした視覚的にロスのない画像圧縮手法に関して求められる要件を考察した．考察結果にもとづき，圧縮率1/3をターゲットとした準可逆な画像圧縮手法を提案した．提案手法は，４×４画素ブロック単位で色空間領域を算出し，色空間領域の代表点として得られる候補色の集合に対して，各画素をそれぞれ最も近い候補色にマッピングして縮退することによりデータ量を削減することを基本アルゴリズムとし，品質向上のための圧縮モード追加を行ったものである．本稿では，いくつかの評価画像を用いてPSNR指標により評価して考察を行うとともに，他の従来手法との比較を行うことによって，提案手法の有効性を実証した．

今後の研究の方向性としては，提案手法のハードウェア実装を行い正確な回路規模を見極めるとともに，GRADモードやSPモードの改善によるPSNR性能の向上，圧縮誤差の最大値の抑制による視覚的な画像ロスレス品質の向上を予定している．

参考文献

［1］ M. J. Weinberger and G. Seroussi:“The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS,”IEEE Trans. on Image Process., vol.9, no.8, pp.1309-1324（2000）.

［2］ C. Christopoulos, A. Skodras, and T. Ebrahimi:“The JPEG2000 still image coding system: an overview, IEEE Trans. on Consumer Electronics, vol.46, no.4, pp.1103-1127（Nov. 2000）.

［3］ ISO/ISE 10918-1,“Digital compression coding of continuous-tone still images,”Draft International Standard,（Jul. 1992）.

［4］ 加藤裕視, 山田晃久, 尾上孝雄：“フレームメモリの削減を目的とした画像圧縮手法,” 電子情報通信学会技術研究報告，vol.107, no.476, pp.31-36（Jan. 2008）.

［5］ 山田晃久，前田真一，山口雅之：“画像符号化装置，画像符号化方法，画像符復号化システム,” 日本特許 登録 No.4612716（Sep. 2008）.

［6］ M. Yamaguchi, A. Yamada, and S. Maeda,“Image encoding device, image encoding method, image encoding/decoding system,” U.S. Patent（CN, EP, US, WO, DE, GB, NL）No.8340445,（Sep. 2008）.

［7］ 中前貴司，山田晃久，山口雅之，尾上孝雄：“フレームメモリ容量削減のための準可逆画像圧縮手法,” 電子情報通信学会技術研究報告，MSS111-105, VLD111-103, CAS111-102, SIP111-104, pp.163-168（June 2011）.

［8］ E. J. Delp and O. R. Mitchell,“Image compression using block truncation coding,” IEEE Trans. on Communication, COM-27, 1335-1342（1979）.

［9］ M. D. Lema and O. R. Mitchell,“Absolute moment block truncation coding and its application to color images,” IEEE Trans. Communication, COM-32, No.10, pp.1148-1157（1984）.

［10］V. R. Udpikar and J. P. Raina,“BTC image coding using vector quantization,” IEEE Trans. on Communication, COM-35, No.3, pp.352-356（1987）.

［11］J. W. Han, M. C. Hwang, S. G. Kim, T. H. You, and S. J. Ko,“Vector quantizer based block truncation coding for color image compression in LCD overdrive,” IEEE Trans. on Consumer Electronics, vol.54, no.4, pp.1839-1845（Nov.2008）.

［12］K. I. Iourcha, K. S. Nayak, Z. Hong,“System and method for fixed-rate block-based image compression with inferred pixel values,” US. Patent No.5956431（2006）.

［13］VESA Display Stream Compression（DSC）Standard v1.2（Jan. 2016）.

［14］VESA Display Compression-M（VDC-M）Standard v1.2（July 2019）.

［15］坂内正夫，大沢裕，曽根光男，尾上守夫：“画像処理研究用標準画像データベース SIDBA の運用について,” テレビジョン学会技術報告 8（38）, 7-12（1984）.