フラクタル符号化の高速化

富樫慎司[†] 池原雅章[††] 野寺隆[†]

近年，画像情報圧縮の一手法としてフラクタル符号化が注目されている．フラクタル符号化は，原画像と，原画像に縮小変換を施した縮小画像との距離が最小になる変換パラメータを符号として使っている．この変換を求める際に画像を分割した小ブロック間を総当たりで比較するために，符号化時間が膨大になるという欠点を持っている．本稿では，フラクタル符号化のための3種類の高速符号化法を提案する．第一の方法として，符号の一部である対称変換をブロックの重心を使って決定することで，画質の劣化を最小限にとどめながら符号化時間を大幅に削減する手法を提案する．第二の方法では，縮小性を保つための条件を用いて探索ブロックを削減する．第三の方法は，適応的にブロックの形状を分割する方式を使った符号化に関する高速化手法である．原画像上のブロック間の相関を求める前に，原画像を縮小した縮小画像で試験的に相関を計算し，その結果によって相関を求めるブロックをあらかじめ選別したうえで最終的な原画像を変換するという手法を提案する．最後に，3つの手法を組み合わせることで，画質にほとんど影響を与えずに全探索による手法の約7%の時間で符号化できることを示す．

Fast Encoding Methods of Fractal Coding

SHINJI TOGASHI,[†] MASAAKI IKEHARA[††] and TAKASHI NODERA[†]

Recently fractal coding is paid attention as an image compression method. In fractal coding, the code should express a contractive transformation. The transformation is set up as the distance between the original image and the transformed image is minimum. The encoding time is huge to compare lots of blocks on the original image. In this paper, three fast encoding methods are proposed. In the first method, the coder decides a symmetrical transformation using the center of gravity of blocks. As a result the encoding time is reduced drastically with a little loss of image quality. In the second method, the number of searched blocks is reduced using the condition keeping contractive transformation. The third method is applied for the coding methods dividing blocks adaptively. The coder first computes a correlation between smaller blocks which requires less computation time, then selects blocks that are used for computation on the original image. In the last of this paper, the above three methods are combined. As experimentally shown, their combined methods are remarkably improved. Encoding time is comparatively reduced by 93 %, giving decoded image whose drop in PSNR was only 0.13 dB.

1. はじめに

フラクタル符号化はJacquin[1]によって提案された画像圧縮の手法のひとつである．この手法は，DCTによるJPEG等の周波数領域における統計的性質に基づく変換符号化と比較するとその性質を異にする画像符号化法である[2)~4)]．その原理はフラクタル画像を生成するIFS (Iterated Function Systems) 理論に基づいており，任意の画像に対して，縮小写像を反復的に施すと唯一の画像に収束することを一般の濃淡画像に拡張したものである．従来の手法と圧縮率において同程度の性能を持ち[4)]，符号となる縮小写像を反復的に施すことで画像の復元が高速に行える利点を持っている．また，最近になってウェーブレット変換とベクトル量子化にフラクタル符号化の符号化アルゴリズムを採り入れることで，従来の標準の手法を上回る性能が得られることが報告されている[5)]．この手法は符号化時間が膨大になるという大きな欠点を持っており，この問題を解決するための高速化手法も提案されている[2),6)]．しかし，その高速化は十分ではなく，また，ある程度の画質の劣化をともなうものであった．本稿では，フラクタル符号化のための3種類の高速符号化法

† 慶應義塾大学理工学部数理科学科
Department of Mathematics, Faculty of Science and Technology, Keio University
†† 慶應義塾大学理工学部電気工学科
Department of Electrical Engineering, Faculty of Science and Technology, Keio University

を提案する．第一の方法として，符号の一部である対称変換をブロックの重心を使って決定することで，画質の劣化を最小限にとどめながら符号化時間を大幅に削減する手法を提案する．第二の方法では，縮小性を保つための条件を用いて探索ブロックを削減する．第三の方法は，適応的にブロックの形状を分割する方式を使った符号化に関する高速化手法である．原画像上のブロック間の相関を求める前に，原画像を縮小した縮小画像で試験的に相関を計算し，その結果によって相関を求めるブロックをあらかじめ選別したうえで最終的な原画像を変換するという手法を提案する．最後に，3つの手法を組み合わせて使うことにより大幅に符号化時間が短縮できることを示す．

2. フラクタル符号化のアルゴリズム

フラクタル符号化について概説する．最初に，原画像を互いに重なり合わない $R \times R$ のブロックに分割する（レンジブロック）．次に，同様に原画像中から $2R \times 2R$ 画素のブロック d（ドメインブロック）を取り出す．このように定めた各レンジブロック r に対して**図1**に示すようなアルゴリズムで符号となる変換とドメインブロックを決定する．各ドメインブロック d は，縮小変換 S，対称変換 I，輝度スケーリング α，輝度シフト o を施され，これとレンジブロック r との距離が求められる．縮小変換 S は，$2R \times 2R$ 画素のブロックを 2×2 画素ごとに平均化して，$R \times R$ 画素のブロックに縮小する．対称変換 I は，ブロックを0度，90度，180度，270度の回転変換と対角線に関する対称変換を組み合わせた8種類の変換の1つを選択して，ドメインブロックに施される．輝度スケーリング α はブロック内の全画素値を α 倍すること，輝度シフト o はブロック内の全画素値に o を加算することを意味する．輝度シフト，輝度スケーリングは最小二乗法によってレンジブロックとドメインブロックの二乗誤差を最小にする値が選ばれる．

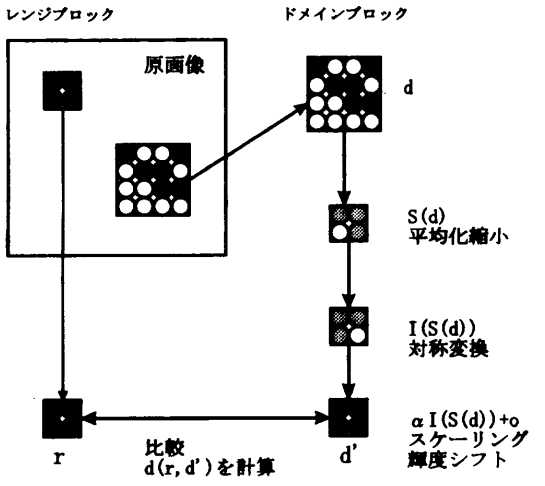


図1 符号化原理
Fig. 1 Coding principle.

変換されたドメインブロックのうち，r との距離が最小なドメインブロックの位置情報と変換パラメータを符号とし，情報圧縮を行う．

この符号化法は，すべてのレンジブロックに対して距離が最小なドメインブロックと変換を全探索によって求めているため，符号化時間が膨大になるという問題点を持っている．

具体的な符号決定のアルゴリズムを考えると，符号となる変換を決定するためにドメインブロックとレンジブロック間で必要な計算は相関である．その他に必要な値（ブロックの輝度値の二乗総和，総和）は，各レンジブロック，ドメインブロックごとに1回ずつ計算して記憶しておくことにより，繰り返し使用できるため計算時間に与える影響は少なく，符号化時間の大部分はブロック間の相関を求めることに費やされている．したがって，符号化時間短縮のためには，ドメインとレンジブロックの相関を求める回数を復元画像の画質に影響を与えずに削減する必要がある．すなわち，1つのレンジブロックに対して，事前に対称変換やドメインブロックの候補をどれだけ絞り込めるかという問題に帰着される．

これまでの高速化法としては，ブロックを分散などにより分類し，同じクラスに含まれるものどうしだけを比較する手法[1]やブロックを平均化，間引きなどにより縮小し，計算量を削減する手法[6]がある．これらの手法は，ある程度の画質の劣化をともなっている．本論文で提示する手法は，フラクタル符号化の変換パラメータの性質に基づき探索ブロックを絞り込むことにより，高速化を実現しながら画質の劣化をきわめて少なくするものである．

3. 実験方法

以降の実験で各レンジブロックの変換は，**表1**のビット数で量子化されている．本論文の実験では，固定長符号化をしているが，可変長符号化を使用することでさらに圧縮率が向上する[4]．しかし，本論文の主旨である符号化時間の短縮には影響がないと考えられるので，ここでは固定長符号化を用いた．

圧縮率は1画素あたりのビット数，ビットレートを使って表す．

表1 変換のビット表現
Table 1 Information of bits of transformation.

	ビット数
輝度スケーリング	4
輝度シフト	8
対称変換	3
ブロックの位置情報	12
ブロックの大きさ	2

$$\text{ビットレート} = \frac{\text{総符号量}}{\text{原画像の総画素数}}[\text{bits/pixel}]$$

また，画質の評価としてPSNR（Peak to peak Signal to Noise Ratio）を使うものとする．

$$\mathbf{PSNR} = 10\log_{10}\frac{(\text{画像のダイナミックレンジ})^2}{\text{平均二乗誤差}}[\text{dB}]$$

ここで，平均二乗誤差は原画像と再生画像を比較したものを示す．

すべての実験で高速化の対象となるフラクタル符号化法は，レンジブロックとして 16×16，8×8，4×4 画素の3種類のブロックを適応的に用いている[2]．ブロック分割の条件は二乗誤差で1画素あたり49以上で分割している．また，ドメインブロックはあらかじめ画像全体を縦1/2，横1/2に縮小し，その中からレンジブロックと同じ大きさのブロックを取り出して使っている[3]．輝度スケーリングは，正の値に限定し0から0.9375まで0.0625刻みの数値の中で最適なものを選んでいる．輝度シフトは，レンジブロックの平均値を使うことにより再生画像の収束を早めている[4]．

4. ブロックの重心を利用した対称変換決定の高速化

従来のフラクタル符号化では，符号の一部である対称変換 I を決定するために，1つのドメインブロックに対して8種類の対称変換をそれぞれ施し，それぞれ最適パラメータを作成し，二乗距離を求め，レンジブロックに最もよく近似しているものを選んでいた．本章では，この対称変換の決定過程を短縮する手法を提案する．

対称変換 I は，**図2** に示すように x 方向の対称変換を1ビット目，y 方向の対称変換を2ビット目，$x=y$ 方向の対称変換を3ビット目と決めて，3ビット目から順にビットが1ならば対応する対称変換を行い，0ならば行わないとする．この方法により8種類の対称変換を3ビットの二進表現とそれに対応する3つの対称変換によって表現することができる．

4.1 手法の説明

$N\times N$ のブロック b を考える．その画素値を左上から右に $b_{0,0}, b_{1,0}, \ldots, b_{N-1,0}$，左上から下に

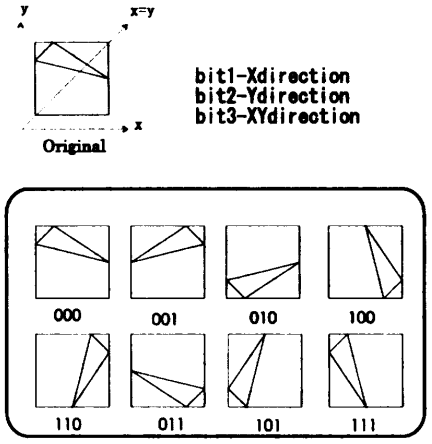


図2 対称変換
Fig. 2 Symmetrical transformations.

$b_{0,0}, b_{0,1}, \ldots, b_{0,N-1}$，右下を $b_{N-1,N-1}$ のように二次元のインデックスで表す．このブロックの中心を起点とした重心の位置ベクトル (g_x, g_y) を

$$g_x = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\frac{x}{N}\frac{b_{x,y}}{M} - \frac{N-1}{2N}$$

$$g_y = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\frac{y}{N}\frac{b_{x,y}}{M} - \frac{N-1}{2N}$$

$$\text{ただし，} M = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} b_{xy}$$

と定義する．

この量 g_x，g_y を使って，レンジブロックとドメインブロック間の対称変換を1つに決定する．そのために，この位置ベクトルを8種類に分類して，それぞれ二進表現（000, 001, ..., 111）で表す．分類の方法は，**表2** に従うものとする．この二進表現は，**図3** のように重心のある区域を表している．

本手法では，従来法のようにドメインブロックに8種類の対称変換を行わずに，まずレンジブロック，ドメインブロックからそれぞれの重心を計算して3ビットの二進表現を求め，図3に示した8つの区域に分類する．次に重心のある区域が重なるように対称変換を決定する．具体的には，比較するレンジブロック r とドメインブロック d の重心の位置ベクトル $\mathbf{g}^r, \mathbf{g}^d$ をそれぞれ計算し，表2に従って二進表現 $e_3^r e_2^r e_1^r$，$e_3^d e_2^d e_1^d$ に変換する．この2つの二進表現 $e_3^r e_2^r e_1^r$，$e_3^d e_2^d e_1^d$ から**表3** に従ってドメインブロックの対称変換を表す二進表現 $e_3 e_2 e_1$ を作成する．$e_3 e_2 e_1$ は，図2に示した変換を表す二進表現になっている（1ビット目が x 方

表2 重心の位置と二進表現
Table 2 The position of the center of gravity of blocks and binary expression.

	1	0
ビット1	$g_x < 0$	$g_x \geq 0$
ビット2	$g_y < 0$	$g_y \geq 0$
ビット3	$\|g_y\| > \|g_x\|$	$\|g_y\| \leq \|g_x\|$

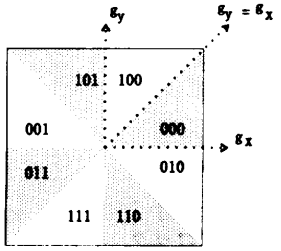


図3 重心の位置と二進表現
Fig. 3 The position of the center of gravity of blocks and its binary expression.

表3 対称変換の決定
Table 3 Determination of symmetry transformation.

e_3^r xor e_3^d	e_3	e_2	e_1
0	0	e_2^r xor e_2^d	e_1^r xor e_1^d
1	1	e_2^r xor e_1^d	e_1^r xor e_2^d

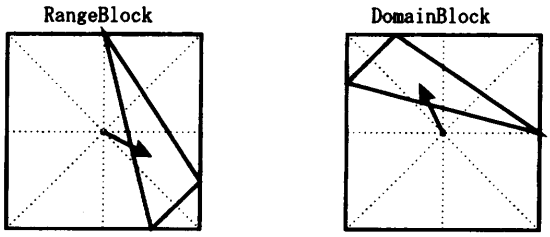
向，2ビット目が y 方向，3ビット目が $x = y$ に関する対称変換)．この変換は，図3に示した重心の含まれる区域を重ねるように作られている．その計算量は，ブロックに8種類の対称変換を行って，それぞれ比較する方法に比べて十分に少なく，大幅な符号化時間の短縮が期待できる．

簡単な例として，図4に示す2つのブロック上の方向の違う相似な三角形を例に，対称変換の決定方法を示す．それぞれの重心の位置から，表2，図3に従って，二進表現 010 ($e_3^r e_2^r e_1^r$)，101 ($e_3^d e_2^d e_1^d$) が決まり，この2つの二進表現から表3に従って変換となる二進表現 $e_3 e_2 e_1$ を求めると 100 となる．すなわち，図2上の変換 100 ($x = y$ に関する対称変換) に相当しており，2つの三角形が重なるように対称変換が決まっているのが分かる．

この手法によって，対称変換を決定したブロックには，さらに輝度スケーリング，輝度シフトを施すことになる．この際，輝度スケーリングを正の値に限定して最小二乗法により最適なパラメータを決定すれば，重心の分類に影響を及ぼさないことに注意されたい．

4.2 実験結果

表4は，512 × 512 画素，256 階調 (8 bits/pixel) の濃淡画像 lenna を使っての，本手法と全探索による



重心の二進表現 010　　重心の二進表現 101

変換となる二進表現 100

図4 変換の決定例
Fig. 4 Example of deciding transformation.

表4 lennaによる実験結果
Table 4 The experimental result of image lenna.

	圧縮率 (bits/pixel)	PSNR (dB)	符号化時間 (second)
提案法	0.5403	34.0498	531
全探索	0.5331	34.1565	3858

表5 全探索によるブロックの対称変換と重心により決定した対称変換とが一致する割合
Table 5 The ratio of the number of same symmetrical trasformations of proposed method and full-search method to the number of all blocks.

ブロックの大きさ	一致ブロック数/総ブロック数	割合
16 × 16	452/582	0.777
8 × 8	703/895	0.785
4 × 4	2842/3492	0.814

手法の実験結果である．PSNR は二乗誤差に基づく画質を表す尺度で，大きければ大きいほど原画像に近いことを表す．また，輝度スケーリングは重心の性質を保つため正の値のみを使っている．

表4から本手法に要する符号化時間は，全探索の場合に比べ大幅に減少していることが分かる．しかも，画質は PSNR で 0.11 dB 程度の劣化にとどまっており，圧縮率も 0.01 bits/pixel しか落ちていない．

また，視覚的には本手法と全探索による手法の復元画像は違いを見つけるのが困難であり，本手法を使ったことによる影響はほとんどないと考えてよい．

表5は，lenna を符号化した際に本手法によって決定した対称変換と，8種類の対称変換すべてを施して，その中で二乗誤差が最小である対称変換とが一致する割合をブロックごとに表したものである．約8割のブロックで本手法と全探索による手法が同じ対称変換を選んでいることが分かり，本手法による対称変換の決定方法が重心の位置を一致させると同時に，ブロック間の二乗誤差を少なくする方向を選択するものであることが分かる．

5. 縮小性を満たす条件による探索ドメインブロックの削減

フラクタル符号化は，縮小写像のアトラクタを復元画像とする．したがって，符号化時に決まる変換は縮小写像でなければならない．しかし，最小二乗法によって決められる最適な輝度スケーリングの値 α は，必ずしも $|\alpha|<1$ とならない．$\alpha>1$ となっているものは，1以内の値に量子化される．このとき，$\alpha>1$ となるレンジブロックとドメインブロックの距離は最適なものから外れてしまう．このことから，$|\alpha|<1$ を満たす条件がブロック間の相関を求める前に計算可能なら，その条件を使って符号化時間を短縮できると考えられる．

ここで扱うフラクタル符号化は平均値分離方式[4]を仮定している．この方式は，復号化を高速に行うために，輝度シフトとしてレンジブロックの平均値を用いている．このため，レンジブロックから直流成分を除いたものを近似する変換を探索することになり，ドメインブロックからも直流成分を取り除いて使い，ブロック間の二乗誤差を小さくしている．平均値分離方式を導入することにより，前述の輝度スケーリングの縮小性を保つための条件が簡単に表され，本章ではこの条件を使った高速符号化法を提案する．

5.1 手法の説明

符号化の際，画質を制御するためにあらかじめブロックの誤差の許容範囲を設定する．その値を ϵ とする．$N\times N$ 画素のブロック b に対してユークリッドノルムを

$$\|b\|=\sqrt{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}b_{x,y}^2}$$

と定義する．

平均値分離方式の符号化ではレンジブロック r と符号となるドメインブロック d との間には輝度スケーリングを α とすると，

$$\|(r-\mu)-\alpha(d-\gamma)\|<\epsilon$$

μ，γ は，それぞれ r，d の平均値

という関係が満たされていなければならない．輝度スケーリングを正に限定すると，三角不等式より，

$$\|r-\mu\|-\alpha\|d-\gamma\|<\epsilon$$

が成り立つ．よって，

$$\frac{\|r-\mu\|-\epsilon}{\|d-\gamma\|}<\alpha$$

が成立していれば，レンジブロックとドメインブロックの誤差は許容範囲内にあると考えられる．ここで，縮小性を満たす条件 $\alpha<1$ をさらに考えると，この条件を満たすには，

$$\frac{\|r-\mu\|-\epsilon}{\|d-\gamma\|}<1$$

つまり，

$$\|r-\mu\|-\epsilon<\|d-\gamma\| \qquad (1)$$

が十分条件である．

式(1)は，レンジブロック，ドメインブロックの相関を必要としていない．よって，別々に $\|r-\mu\|$，$\|d-\gamma\|$ を求めるだけで式(1)が成り立つかどうかを判定することができる．一般的にこの符号化で使われる距離関数はユークリッドの距離なので，レンジブロック，ドメインブロックのユークリッドノルムは計算の効率化のために探索前に求めることができる．よって，式(1)は符号化時間のほとんどを占めるブロック間の相関を計算する前に評価することが可能である．したがって，本手法は式(1)を満たさないブロック間では相関を計算せず，式(1)を満たすブロック間でのみ相関を計算すればよく，探索範囲を大幅に減少できるため符号化時間の短縮が可能となる．

5.2 実験結果

レンジブロックとして 16×16，8×8，4×4 画素の3段階のブロックを大きな順に適用し，あらかじめ決められた閾値を満たさなければ4分割していく方法[2]を使って，本手法を実装する場合の実験結果について述べる．式(1)の ϵ には分割条件の閾値をそのまま使用している．

表6 は，本手法と全探索による手法を実行した結果である．提案法の符号化時間は，全探索による手法の約66%となっている．しかも，圧縮率に変化はなく，全探索と比べた場合の画質の劣化はPSNRで約0.03 dBときわめて少ない．これより，評価関数として式(1)が適切なものであることが分かる．本方法は，縮小変換の性質に基づいた条件を使っているため，高速化を達成しつつも画質の劣化はきわめて少ない手法であるといえる．

表6 lennaによる実験結果
Table 6 The experimental result of image lenna.

	圧縮率 (bits/pixel)	PSNR (dB)	符号化時間 (second)
縮小性保存	0.5331	34.1308	2534
全探索	0.5331	34.1565	3858

6. 縮小画像間の前探索による探索ブロックの制限による高速化

本章で提案する手法を使うフラクタル符号化は4分

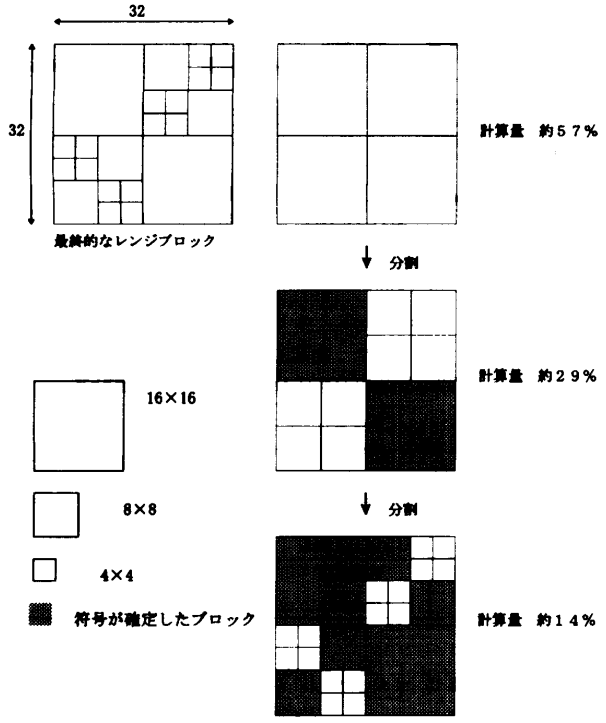


図5 4分割法の計算量
Fig. 5 The computational quantity of quad-tree method.

割法[2]を仮定している．つまり，大きなブロックで符号化を開始して近似精度が良くない場合，さらにブロックを4分割して再符号化して行くというトップダウン方式を仮定する．

図5は，32×32 画素の画像を 16×16，8×8，4×4 画素の3種類の大きさのブロックで符号化する計算過程を示したものである．最終的なレンジブロックの様子は，16×16 画素のブロックが2個，8×8 画素のブロックが4個，4×4 画素のブロックが16個となっている．しかし，トップダウン方式の符号化では，8×8 画素のレンジブロックの探索は，そのブロックを含む 16×16 画素のレンジブロックの探索を行った後に行われる．よって，16×16 画素のブロックは，最終的に符号となる2個だけでなく，分割されたものと合わせて4個探索が行われている．同様に，8×8 画素のブロックは，4個ではなく8個のブロックの探索が行われている．

また，各レンジブロックと相関を求めるドメインブロックの数がブロックの大きさによらず同数だとすると，相関の計算量はブロックの画素数に比例する．たとえば，16×16 画素のブロックどうしの相関を求めるのに乗算256回，加算255回かかるのに対して，8×8 画素のブロックどうしでは乗算64回，加算63回と 16×16 画素のブロックの約1/4となっている．

この考えに基づいて，図5のレンジブロックで符号化を行う場合のブロックの大きさごとの計算量の割合を求めると，16×16 画素のブロックで57%，8×8 画素のブロックで29%，4×4 画素のブロックで14%となることが分かる．たとえば，16×16 での計算量のうち，半分の計算結果は符号に活かされずにブロックを分割するかどうかの判断を行うために使われており，全体の計算量の28%に達している．8×8 の場合も同様に 4×4 に分割するための判断のためにその計算量の半分を使っている．本章で提案する高速化手法は，この最終的な符号を求めるため以外の部分での計算量を削減するためのものである．

6.1 手法の説明

ここまでは，レンジブロックとドメインブロックを比較するときに，レンジブロックを $R \times R$ 画素，ドメインブロックを $2R \times 2R$ 画素として，ドメインブロックに空間的縮小 S を施して $R \times R$ 画素に変換してレンジブロックと比較すると説明してきた．しかし，実装する際には，効率化のために画像全体を空間的に縮小して，ドメインブロックとして $R \times R$ 画素のブロックをそこから取り出している[3]．このようにするとドメインブロックを複数のレンジブロックと比較するときに原画像から逐次 $2R \times 2R$ 画素のブロックを取り出して同じ空間的縮小を何度も繰り返すことを避けることができる．これ以降では，ドメインブロックという場合は，空間的縮小が施されて，レンジブロックと同じ画素数になったもののことをいう．

以下，説明を簡単にするために画像の高さ，幅を2の累乗に限定する．このことにより手法の一般性を失うことはない．

$2^N \times 2^N$ 画素の画像を原画像 ImageR として符号化を行う．ImageR を縦 1/2，横 1/2 に縮小した $2^{N-1} \times 2^{N-1}$ 画素の画像 ImageD とし，ここからドメインブロックを取り出すことにする．また，4分割法におけるレンジブロックの最も大きなものを $2^{\max} \times 2^{\max}$ 画素とする．

通常の手法では，ImageR 上のレンジブロックと ImageD 上のドメインブロックを総当たりに比較するわけであるが，ここで提案する手法では，その他に ImageR を縦 1/2，横 1/2 に平均化により縮小した $2^{N-1} \times 2^{N-1}$ 画素の画像 ImageR′，ImageD を縦 1/2，横 1/2 に平均化により縮小した $2^{N-2} \times 2^{N-2}$ 画素の画像 ImageD′を用意する（**図6**）．

ImageR 上の $2^R \times 2^R$ 画素のレンジブロック r の符号化を考える．ここで，ブロック間の近似精度の1画素あたりの閾値として定数 ϵ をあらかじめ定めておく．

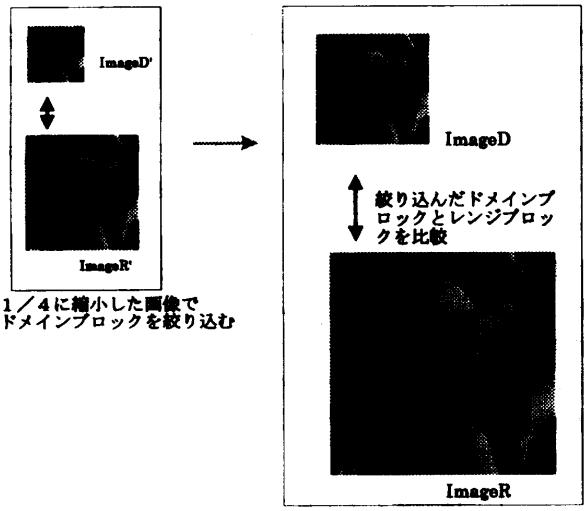


図6 1/4画像の探索による計算量の削減
Fig. 6 Reduction of computational quantity searching 1/4 image.

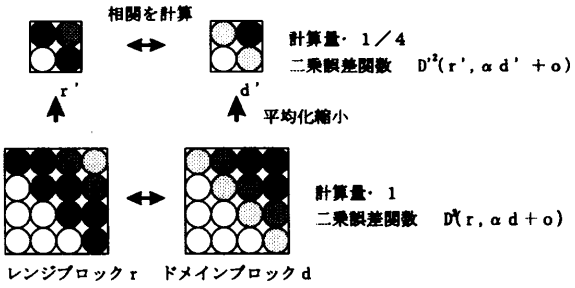
提案する手法では，まず r と d の相関を求める代わりにImageR'上の $2^{R-1} \times 2^{R-1}$ 画素のImageR上での r に対応する位置のブロック r' とImageD'上の $2^{R-1} \times 2^{R-1}$ 画素のImageD上での d に対応する位置のブロック d' の相関を求める．対応する位置というのは，r'，d' がそれぞれ r，d を縦1/2，横1/2に平均化により縮小したものになっている位置という意味である．

r'，d' 間の符号を定め距離 $D'^2(r', \alpha' d' + o')$ を求める．ただし，α'，o' はそれぞれ最小二乗法による最適な輝度スケーリング，輝度シフトとする．この距離が1画素あたりで閾値 ϵ 以下ならば，ImageR，ImageD上の r と d の相関を求め，ブロック間の距離 $D^2(r, \alpha d + o)$ を求める．ただし，α，o はそれぞれ最小二乗法による最適な輝度スケーリング，輝度シフトとする．もし，r'，d' 間の距離が1画素あたりで閾値 ϵ を超えたなら，r と d の相関を求めずに次のドメインブロックの探索に移るようにする．分割の最小ブロックではこの手法を行わない．つまり，ここで提案する手法は，前もって計算量が少なくて済む小さなブロック間で相関を計算し，実際に相関を計算するブロックを選択することにより符号化時間の短縮するものである．

本手法では，$D'^2(r', \alpha' d' + o')$ を計算して，$D^2(r, \alpha d + o)$ の計算を行うかどうかを決定している．ただし，D'^2，D^2 は，それぞれ対応する大きさのブロック間の二乗誤差関数とする．その根拠は，r，d から平均化によって r'，d' を作った場合，

$$D'^2(r', \alpha d' + o) < D^2(r, \alpha d + o)/4 \qquad (2)$$

が成立していることによる．この不等式から，



$$D'^2(r', \alpha d' + o) < D^2(r, \alpha d + o)/4$$

図7 提案法の原理
Fig. 7 The principle of the proposed method.

表7 lennaを使った実験結果
Table 7 The experimental result of image lenna.

アルゴリズム	圧縮率 (bits/pixel)	PSNR (dB)	符号化時間 (second)
小画像前探索	**0.5334**	34.156	2444
全探索	0.5331	34.1565	3858

$$\begin{aligned} D'^2(r', \alpha' d' + o') &< D^2(r, \alpha' d + o')/4 \\ &\quad \text{(不等式 (2) より)} \\ &< D^2(r, \alpha d + o)/4 \\ &\quad (\alpha, o \text{ は } D^2 \text{における最適値}) \end{aligned}$$

となり，

$$\begin{aligned} D'^2(r', \alpha' d' + o') &> \epsilon \times 2^{R-1} \times 2^{R-1} \\ \Longrightarrow D^2(r, \alpha d + o) &> \epsilon \times 2^R \times 2^R \end{aligned}$$

が成立することが示される．よって，同じ閾値 ϵ を使った場合，従来の手法と本手法によって出力される符号は同じものになるはずである．

図7 に示したように，$D'^2(r', \alpha' d' + o')$ を求める計算量は，$D^2(r, \alpha d + o)$ を求める計算量の約1/4であるから，$D'^2(r', \alpha' d' + o') \leq \epsilon \times 2^{R-1} \times 2^{R-1}$ が成立しないブロック，つまり $D^2(r, \alpha d + o)$ を計算しないで済むブロックがある割合を超える場合に，この手法は通常の方法に比べて高速になる．

6.2 実験結果

提案法を 16×16，8×8，4×4 の3段階の4分割法に実装した．

表7 は，画像lennaによる実験結果である．全探索による手法の63%の符号化時間で画質の劣化がPSNRで0.0005 dB，圧縮率の変化も0.0003 bits/pixelにすぎない．これらは，D' の計算時に生じる数値誤差のために最適なブロックが選択されない場合が稀にあるためである．

表8 は，lennaのブロックごとの計算量を概算してその割合を全探索による手法の計算時間を100として示したものである．表から分かるように，従来のトップダウン方式の手法が，ブロックが大きいほど多くの

表8 計算量のブロックごとの割合，全探索の総計算量を100とする
Table 8 The ratio of computational quantity of each block, all computational quantity of full-search method: 100.

ブロックの大きさ	全探索の割合 (%)	提案法の割合 (%)
16 × 16	54	22
8 × 8	27	9
4 × 4	19	19
全体	100	50

表9 lennaを使った実験結果
Table 9 The experimental result of image lenna.

	アルゴリズム	圧縮率 (bits/pixel)	PSNR (dB)	符号化時間 (second)
1	重心	0.5403	34.0498	531
2	縮小性保存	0.5331	34.1308	2534
3	小画像探索	0.5334	34.156	2444
4	2 + 3	0.5334	34.1309	**1632**
5	1 + 2 + 3	0.5406	34.0281	**251**
6	全探索	0.5331	34.1565	**3858**

計算を必要としていたことが分かる．提案法によって，大きさ 16 × 16 画素のブロックでは約 1/2，大きさ 8 × 8 画素のブロックでは約 1/3 になり，符号化時間短縮に効果をあげていることが分かる．

7. 三手法の組合せ

ここまで，フラクタル符号化の高速化手法として3つの手法を提案した．本章では，これらの手法を組み合わせて実行した場合，符号化時間，画質，圧縮率がどのように変化するかを実験によって確かめる．

7.1 実験結果

表9 は，本論文で提案した3つの手法，およびその組合せを画像 lenna を使って行った実験結果である．各手法に番号をつけて，重心による分類は1，縮小性保存は2，小画像探索を3とし，その組合せは + で結ぶことにする．

縮小性保存は，画像によらず符号化時間の短縮が期待できる．小画像探索は，画像によっては符号化時間が増加する可能性がある．しかし，符号化時間が増加するような画像は分割するブロックが少ない画像と考えられ，その符号化時間は全探索による手法によっても短いものとなることが予想される．したがって，この手法を用いることによる影響は少ないと考えられる．2 + 3 (縮小性保存と縮小画像前探索の組合せ) は，全探索による手法と比べると圧縮率が約 0.003 bits/pixel，画質が PSNR で約 0.03 dB 劣化しただけである．この値は，2 + 3 によって作られた符号が全探索による手法によって作られた符号とほぼ同じものであるといえるほど，わずかな違いでしかない．しかも，全探索による手法の約 42%の時間で符号化を終了する．あらゆる条件において，全探索による手法を使うならば 2 + 3 を使うべきである．また，縮小性保存の計算時間，小画像前探索の計算時間はそれぞれ全探索による手法の計算時間の 66%，63%であり，これを掛け合わせると 0.66 × 0.63 = 0.42 となり 2 + 3 の符号化時間の全探索による符号化時間の割合と一致する．これは，縮小性保存によって削減される探索ブロックの分布と小画像前探索によって削減される探索ブロックの分布が無相関であるためだと考えられる．

すべての手法を組み合わせると画質は PSNR で 0.13 dB，圧縮率は 0.007 bits/pixel 劣化するが符号

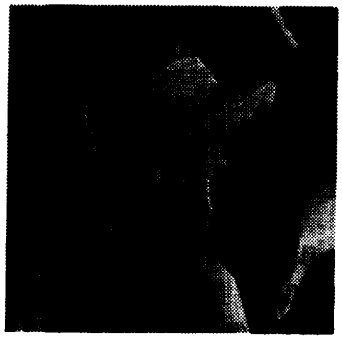


図8 原画像 lenna
Fig. 8 The original image lenna.

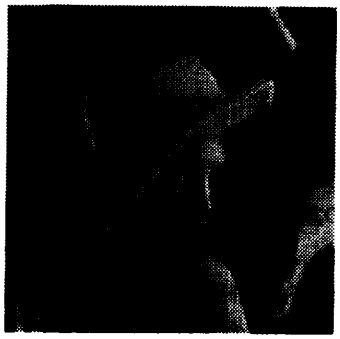


図9 全探索法による復元画像
Fig. 9 The decoded image of full-search method.

図10 提案法による復元画像
Fig. 10 The decoded image of proposed method.

表10 baboonを使った実験結果
Table 10 The experimental result of image baboon.

アルゴリズム	圧縮率 (bits/pixel)	PSNR (dB)	符号化時間 (second)
三手法	1.42579	25.7237	482
全探索	1.42141	26.1801	7206

化時間は251秒，全探索の7%と大幅な短縮を実現している．図8，図9，図10にlennaの原画像，全探索よる手法の復元画像，三手法を組み合わせた手法による復元画像を示す．

また，高周波成分を多く含む画像baboonの実験結果を表10に示す．この場合も符号化時間は全探索の7%を実現しており，PSNRは減少するものの視覚的には差異はなかった．

8. まとめ

ブロックの重心を使って，対称変換を探索なしに決定することで，符号化の質を落とさずに全探索による手法の15%の符号化時間を達成できることを示した．この手法は，その他のブロック形状を使ったフラクタル符号化に適用することも可能であり，対称変換を探索なしに決定するのに有効である．また，提案した3つの手法のうち，縮小性を保存する条件を使ったものと縮小画像間の前探索を行うものは画質，圧縮率をほとんど劣化させないことが理論的にも実験的にも示されており，その考え方はあらゆる種類のフラクタル符号化に採り入れても符号化時間の短縮を達成できるものである．提案した3つの手法を組み合わせて使うことで，従来の高速化手法に比べて少ない画質の劣化で符号化時間を全探索による手法の7%にまで短縮することができた．

参考文献

1) Jacquin, A.E.: A Novel Fractal Block-coding Technique for Digital Images, *Proc. ICASSP-90 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol.4, pp.2225–2228 (1990).
2) Fisher, Y.: *Fractal Image Compression Theory and Application*, Spriger-Verlag (1995).
3) Barnsley, M. and Hurd, L.: *Fractal Image Compression*, AK Peters (1993).
4) 井田，駄竹：反復変換符号化による画像圧縮，電子情報通信学会第5回回路とシステム軽井沢ワークショップ，pp.137–142 (1992).
5) Rinaldo, R. and Calvagno, G: *Image Coding by Block Prediction of Multiresolution Subimages*, IEEE Trans. Image Processing, Vol.4, No.7, pp.909–920 (1995).
6) 川又，長久，樋口：多重解像度木探索による反復変換理論符号化の高速符号化アルゴリズム，電子情報通信学会論文誌A，Vol.J78-A, No.2, pp.253–260 (1995).
7) 高木，下田（監修）：画像解析ハンドブック，東京大学出版会 (1991).

(平成8年 2 月26日受付)
(平成8年10月 1 日採録)

富樫 慎司（正会員）
昭和46年生．平成4年慶應義塾大学数理科学科卒業．平成8年同大学大学院数理科学専攻修士課程修了．同年日本テキサスインスツルメンツ（株）入社．現在同社ASIC製品部所属．

池原 雅章
昭和59年慶應義塾大学工学部電気工学科卒業．平成元年同大学大学院博士課程修了．平成元年長崎大学工学部講師．平成4年慶應義塾大学理工学部電気工学科講師．回路理論，ディジタル信号処理に関する研究に従事．工学博士．電子情報通信学会会員．

野寺 隆（正会員）
昭和57年慶應義塾大学大学院工学研究科博士課程（数理工学専攻）修了．同年，同大学数理科学科助手，平成元年講師となり現在に至る．その間，昭和61年より1年間米国スタンフォード大学客員教授．大規模な行列計算の算法の研究開発に従事．ハイパフォーマンス・コンピューティングや文書処理に興味を持つ．著書に『楽々LaTeX』などがある．工学博士．エッセイスト．SIAM，日本応用数理学会会員．