

## 疑似二乗ブロックマッチングによる フラクタル画像符号化手法

伊藤 和人

埼玉大学 電気電子システム工学科

〒338-8570 埼玉県浦和市下大久保255

E-mail: kazuhito@ees.saitama-u.ac.jp

**あらまし** フラクタル画像符号化では、各画像ブロックに対して、アフィン変換後に最も相似する画像ブロックを探索するブロックマッチングを行う。ブロックの相似性は、ブロック間の画素値差二乗の積算値で評価するので、解析的に最適なアフィン変換係数を算出できるが、多数の乗算を必要とする。一方、候補係数の中から最適な係数を選択する手法では、乗算数は少なく、多数ある二乗計算は近似値を用いることで計算量を低減できる。本研究では、ブロックマッチングに疑似二乗を用いることでハードウェア量を低減したフラクタル画像符号化アーキテクチャを提案する。

**キーワード** フラクタル画像符号化、ブロック相似度、ブロックマッチング、VLSI アーキテクチャ

## An Architecture for Fractal Image Coding using Pseudo-Square Block Matching

Kazuhito Ito

Department of Electrical and Electronic Systems, Saitama University

255 Shimookubo, Urawa, Saitama 338-8570, Japan

E-mail: kazuhito@ees.saitama-u.ac.jp

**Abstract** In fractal image coding, many block matchings are performed to determine the affine transformed image block which best resembles an image block in the image to be coded. Image block resemblance is measured by the sum of the square of pixel value differences between two image blocks. For given two image blocks, the best affine coefficients can be calculated mathematically with many multiplications. The best affine coefficients can also be selected from candidate coefficients by using search method which requires less multiplications and uses pseudo-square operations in resemblance computation. In this paper, an architecture for fractal image coding using pseudo-square block matching is proposed.

**Keywords** fractal image coding, block resemblance, block matching, VLSI architecture

## 1 はじめに

フラクタル画像符号化は、画像データ圧縮手法の1つである [1]。まず、符号化しようとする画像(原画像)を互いに重なり合わないブロックに分割する。これをレンジブロックと呼ぶ。また、原画像をレンジブロックよりもサイズの大きなブロックに分割する。これを大ドメインブロックと呼ぶ。大ドメインブロック同士は重なり合いを許す。大ドメインブロックに対して、レンジブロックと同じサイズに縮小する変換を施したものをドメインブロックと呼び、ドメインブロックに対して更にアフィン変換を施す。

原画像の各レンジブロックについて、同一原画像の中のドメインブロックのうち、アフィン変換後に最もレンジブロックに相似するドメインブロックを選択する。これが、フラクタル画像符号化処理であり、ドメインブロックの画像内位置とアフィン変換係数を、各レンジブロックの符号とする。

フラクタル画像復号化では、任意の初期画像からドメインブロックを抽出し、アフィン変換を施したブロックをレンジブロックとして画像を構成する。この処理を全てのレンジブロックに対して行うことを反復することで、原画像を再現する。

フラクタル画像符号化の特徴は、復号化の際に縮小変換を用いるためブロックノイズが少ないこと、および同一符号から任意のサイズの画像の生成が可能であり、その際に画質が劣化しないことなどがあげられる。

フラクタル画像符号化処理では、符号化に大量の計算量を必要とする。これは、一般にレンジブロック数が多く、各レンジブロックに対してドメインブロックの候補が多数あり、各ドメインブロックについて最適なアフィン変換係数を求める必要があることが理由である。

最適アフィン変換係数を求めるには、2通りの手法がある。1つの手法は解析的に係数を算出する手法 [2, 3] であり、これを解析的手法と呼ぶ。いま1つの手法は、いくつかの候補の中から最適な係数を探索する手法であり、これを探索手法と呼ぶ。

解析的手法では、レンジブロック内の画素値とドメインブロック内の画素値の乗算が必要となる。一方探索手法では、レンジブロック内の画素値とドメインブロック内の画素値の差の二乗計算が必要となる。

探索手法においては、画素値の差の二乗計算はレンジブロックとアフィン変換後のドメインブロックの相似度算出のためである。しかし目的は、画素値差の二乗に基づく相似度の厳密値を計算することではなく、レンジブロックと最も相似したアフィン変換後のドメインブロックを見つけることである。したがって、最適なドメインブロックとアフィン変換係数を抽出するために相似性が正しく評価できればよく、厳密な二乗計算は不要である [4]。

そこで本論文では、二乗値に近似した値を計算し、二乗計算よりも少ないハードウェア量で実現可能な疑似二乗回路を用いた探索手法によるフラクタル画像符号化手法を提案する。最もハードウェア量が多くなると予想されるレンジブロックとドメインブロックのマッチング回路を設計し、ハードウェア量を評価する。

2節ではフラクタル画像符号化方式と最適アフィン変換係数を求める2つの手法を示す。3節では疑似二乗を用いたブロックマッチング回路を提案し、実験結果を4節に示す。

## 2 フラクタル画像符号化

フラクタル画像符号化の原理については、文献 [1, 3] に詳細に述べられており、ここでは省略する。

### 2.1 符号化方式

フラクタル画像符号化方式を図1に示す。まず、原画像を複数のレンジブロックに分割する。レンジブロックのサイズは、横  $N$  画素、縦  $N$  画素とし、互いに重なり合わない。原画像サイズが横 256 画素、縦 256 画素のとき、 $N = 8$  とすれば 1024 個のレンジブロックに分割される。また、原画像を大ドメインブロックに分割する。大ドメインブロックのサイズは、横  $2N$  画素、縦  $2N$  画素とし、座標単位を縦横とも  $N/2$  画素として大ドメインブロック同士の重なり合いを許す。原画像サイズを  $256 \times 256$  画素、 $N = 8$  とすれば、3721 個の大ドメインブロックが存在する。

各大ドメインブロックを縦横とも  $1/2$  に縮小してレンジブロックと同サイズにしたドメインブロックを集めてドメインプールを構成する。各ドメインブロックに対して90度を単位とした回転、および反転を施して得られたものもドメインブロックとしてドメインプールに含める。

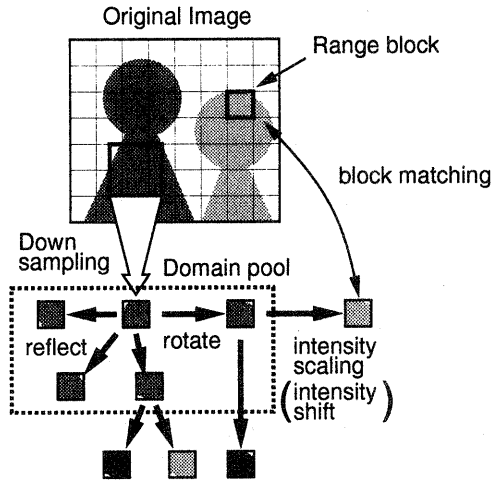


図 1. フラクタル画像符号化

ドメインプールの中から、各レンジブロックに最も相似したドメインブロックを選択するのであるが、相似度の評価の前にドメインブロックに対してアフィン変換を施す。アフィン変換の内容は、画素値に一律に係数を乗じる階調スケールリングと、画素値に一律に定数(オフセット)を加える輝度シフトである。スケールリング係数を  $s$ 、オフセット値を  $o$  とすると、ドメインブロック  $D$  の  $i$  番目の画素値  $a_i$  は以下の様に変換される。

$$a'_i = s \cdot a_i - o \quad (1)$$

ここで、 $a'_i$  は  $D$  にアフィン変換を施したドメインブロック  $D'$  の  $i$  番目の画素値を表わす。

レンジブロック  $R$  とアフィン変換後のドメインブロック  $D'$  の間の相似度は、対応位置にある画素値の差の二乗和とする。レンジブロック  $R$  の  $i$  番目の画素値を  $b_i$  とすれば、相似度  $RD(R, D')$  は以下の式によって計算できる。

$$\begin{aligned} RD(R, D') &= \sum_{i=0}^{n-1} (b_i - a'_i)^2 \\ &= \sum_{i=0}^{n-1} (sa_i + o - b_i)^2 \end{aligned} \quad (2)$$

ここで、 $n$  はブロックの画素数 ( $= N^2$ ) である。相似度  $RD(R, D')$  が小さいほど、ブロック  $R$  とブロック  $D'$  は相似性が高い。

フラクタル画像符号化においては、各レンジブロック  $R$  について、ドメインプール内の各ド

メインブロック  $D$  に対して最適なアフィン変換を施して  $RD(R, D')$  を最小化する。そしてドメインプールの中から  $RD(R, D')$  が最小のドメインブロックを選択する。すなわち、フラクタル画像符号化では、特定のドメインブロックについて最適なアフィン変換係数を求める処理と、ドメインプールの中から最適なドメインブロックを探索する処理の2つのレベルの最適化処理が行われる。

アフィン変換係数最適化には、2通りの手法が考えられる。また、最適ドメインブロック探索の計算量を低減するため、クラス化が行われる。

## 2.2 アフィン変換係数最適化

レンジブロック  $R$ 、ドメインブロック  $D$  が与えられたとき、式(2)に示すブロック相似度  $RD(R, D')$  を最小化するアフィン変換係数  $s$ 、 $o$  は、それぞれ  $RD(R, D')$  に対する微係数を0とする値である。

$$\frac{\partial RD}{\partial s} = 2 \sum_{i=0}^{n-1} a_i (sa_i + o - b_i) = 0 \quad (3)$$

$$\frac{\partial RD}{\partial o} = 2 \sum_{i=0}^{n-1} (sa_i + o - b_i) = 0 \quad (4)$$

上の2式より、 $s$ 、 $o$  は以下の様に計算できる。

$$s = \frac{n \sum a_i b_i - \sum a_i \sum b_i}{n \sum a_i^2 - (\sum a_i)^2} \quad (5)$$

$$o = \frac{\sum b_i - s \sum a_i}{n} \quad (6)$$

このとき、 $RD(R, D')$  の最小値  $RD^m(R, D')$  は以下の通りである。

$$RD^m(R, D') = \frac{n \sum b_i^2 - (\sum b_i)^2}{n} - \frac{n \sum a_i^2 - (\sum a_i)^2}{n} s^2 \quad (7)$$

スケールリング係数  $s$  は  $0 \leq s < 1.2$  の範囲であり、5ビット精度で十分とされる [2]。

レンジブロック  $R$ 、ドメインブロック  $D$  が与えられたとき、最適なアフィン変換係数を求めるには、式(5)を用いて直接算出すればよい。これを解析的手法と呼ぶ。解析的手法では、 $\sum a_i$  や  $\sum b_i$  などは各ドメインブロックとレンジブロックについて1回だけ計算し保存しておけば

よいが、 $\sum a_i b_i$  はドメインブロックとレンジブロックの組合せごとに計算する必要がある。

前述のように  $s$  の精度を 5 ビットとすると、 $s$  の値は 32 通りのみである。そこで、32 通りの  $s$  の値について式 (2) の  $RD(R, D')$  を計算し、最小値  $RD^m(R, D')$  を与える  $s$  を求めることもできる。これを探索手法と呼ぶ。

ドメインブロックとレンジブロックの画素値平均をそれぞれ  $\bar{a} = \sum a_i/n$ 、 $\bar{b} = \sum b_i/n$  とする。式 (6) より  $o = \bar{b} - s\bar{a}$  であることを考慮して、相似度は以下のように計算できる。

$$RD(R, D') = \sum_{i=0}^{n-1} (s\hat{a}_i - \hat{b}_i)^2 \quad (8)$$

ここで  $\hat{a}_i = a_i - \bar{a}$  は、ドメインブロックの画素値  $a_i$  から平均  $\bar{a}$  を減じた値である。 $\hat{b}_i$  についても同様である。

式 (8) は、探索手法によって 32 通りの  $s$  のうち最小の  $RD(R, D')$  を与える  $s$  を特定する上で  $o$  の値が不要であることを意味している。 $s$  の特定後、式 (6) に代入して  $o$  を算出する。

### 2.3 最適ドメインブロック探索

各レンジブロックに対して、ドメインプールの中から最適なドメインブロックを選び出す探索処理が行われるが、ドメインプールが膨大であり、探索に多数のブロックマッチングを要する。フラクタル画像符号化時間を削減するには、ブロックマッチング回数を削減することが有効である。

レンジブロック、ドメインブロックとも、ブロック内画素値の分布特性に基づいてクラス化する。レンジブロックに最も相似したドメインブロックは、レンジブロックと同じクラスに属すると予想されるので、ブロックマッチングを同一クラスのブロック間に限定することで、レンジブロックに最も相似したドメインブロックとのマッチングを除外することなく不要なブロックマッチングを排除できる。

クラス化は次のように行う [3]。以降では、レンジブロック、ドメインブロックを合わせて単にブロックと呼ぶ。まず、ブロックを 4 個のサブブロックに分割する。サブブロック  $k$  ( $k = 0, 1, 2, 3$ ) の画素値平均  $A_k$ 、分散  $V_k$  を求める。

$$A_k = \frac{1}{n_4} \sum_{i=0}^{n_4-1} x_i^k \quad (9)$$

$$V_k = \frac{1}{n_4} \sum_{i=0}^{n_4-1} (x_i^k - A_k)^2 \quad (10)$$

ここで、 $n_4$  はサブブロックの画素数、 $x_i^k$  はサブブロック  $k$  の  $i$  番目の画素値である。

画素値平均最大のサブブロックが左上に位置するように、90 度を単位としてブロックを回転する。この状態で、画素値平均が次に大きいサブブロックの位置により 3 通りにクラス化する。これを第 1 クラス化とする。また、画素値分散の大小関係により  $4! = 24$  通りにクラス化する。これを第 2 クラス化とする。第 1 クラス化と第 2 クラス化により、ブロックは 72 通りのクラスのいずれかに属する。なお、回転の大きさ (0 度、90 度、180 度、270 度) をブロックごとに記憶しておく。

## 3 ブロックマッチングアーキテクチャ

与えられたドメインブロック、レンジブロックの組に対して、2.2 節で述べた探索手法に基づいて最適なアフィン変換係数を求めるとともに、各レンジブロックについて最適なドメインブロックを選択するアーキテクチャを提案する。

### 3.1 スケーリング係数最適化手法

与えられたドメインブロックとレンジブロックの組に対して、探索手法では、32 通りのスケール係数の中から、相似度最小 (相似性最大) のスケール係数を求める。式 (8) の変形により、相似度  $RD(R, D')$  とスケール係数  $s$  について以下の関係が得られる。

$$RD(R, D') = \left( \sum_{i=0}^{n-1} \hat{a}_i^2 \right) s^2 - 2 \left( \sum_{i=0}^{n-1} \hat{a}_i \hat{b}_i \right) s + \sum_{i=0}^{n-1} \hat{b}_i^2 \quad (11)$$

すなわち、横軸を  $s$ 、縦軸を  $RD(R, D')$  とすると図 2 に示すように下に凸の放物線となる。

$0 \leq s < 1.2$  を満たす 32 通りの  $s$  を昇順に  $s_0, s_1, \dots, s_{31}$  とする。まず、8 通りの  $s_{4p}$  ( $p = 0, 1, 2, \dots, 7$ ) について式 (8) にしたがって相似度  $RD(R, D')$  を計算し、8 通りの  $RD(R, D')$  の内の最小値を与える  $s_{4p'}$  を得る。 $RD(R, D')$  が下に凸の放物線となることから、十分な計算精度があれば 32 通りの  $s$  に対する 32 通りの  $RD(R, D')$  の内の最小値を与える  $s$  は、 $s_{4p'+q}$  ( $q = -3, -2, -1, 0, 1, 2, 3$ ) の中に存在する。

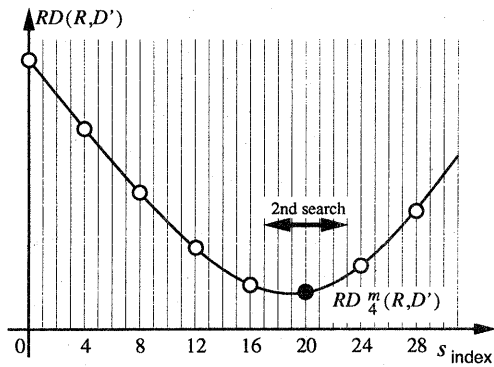


図 2. スケーリング係数と相似度の関係

そこで、次に 7 通りの  $s_{4p'+q}$  ( $q = -3, -2, \dots, 3$ ) について相似度  $RD(R, D')$  を計算し、相似度最小の  $s_{4p'+q}$  を得る。 $q = 0$  の場合の相似度はすでに計算済みであるが、ハードウェア化を想定した場合、再計算した方が都合がよい (3.3 節で述べる)。なお、 $4p'+q < 0$  の場合には相似度計算は行わない。

8 通りの  $s_{4p'}$  に対する相似度最小値を  $RD_4^m(R, D')$ 、32 通りの全ての  $s$  に対する相似度最小値を  $RD^m(R, D')$  と表わす。

### 3.2 最適ドメインブロック選択手法

レンジブロック  $R$  とクラスが同一のドメインブロックの集合 (ドメインプールの部分集合) を  $DP_R$  と表わす。

与えられたレンジブロック  $R$  について、相似性最大の最適なドメインブロックは、 $DP_R$  に属するドメインブロックのうち、最適なスケーリングを施して得られた最小相似度  $RD^m(R, D')$  が最小となるものである。これを  $D_R$  と表わす。

ここで、最適ドメインブロック選択の計算量削減のため、次の最適化手法を用いる。

まず、 $DP_R$  中の各ドメインブロック  $D$  について、8 通りの  $s$  に対する最小相似度  $RD_4^m(R, D')$  を求める。次に、全てのドメインブロック  $D \in DP_R$  について、最小の  $RD_4^m(R, D')$  を与えるドメインブロック  $\hat{D}$  を求める。この  $\hat{D}$  が最適ドメインブロック  $D_R$  であるとする。

これにより最適ドメインブロックは選択されたが、 $D_R$  に対する最適なスケーリング係数がまだ得られていない。そこで、 $\hat{D} (= D_R)$  について、7 通りの  $s_{4p'+q}$  ( $q = -3, -2, \dots, 3$ ) に対する最小相似度  $RD_4^m(R, \hat{D}')$  を求める。

上に述べた最適ドメインブロック選択手法は、

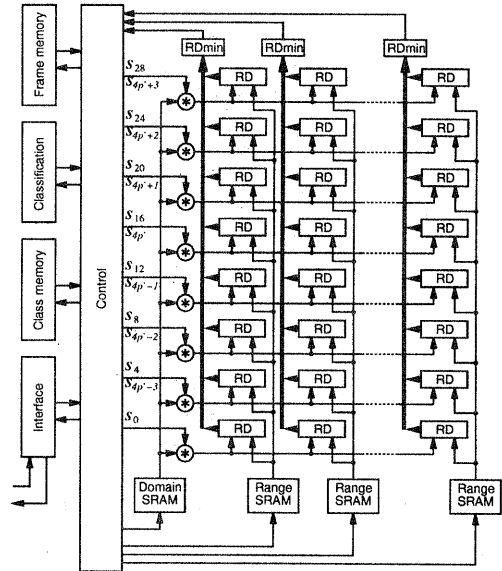


図 3. フラクタル画像符号化アーキテクチャ

同一クラスに属するレンジブロックについて並列に選択処理を行う場合に適している。すなわち、8 通りのスケーリングを施したドメインブロックデータを共通に使用し、レンジブロック  $R$  に対する  $D_R$  と  $s_{4p'}$  を並列に選択する。この処理は、同一クラスの多数のレンジブロックと多数のドメインブロックをマッチングしており、並列処理による高速化が要求され、かつ並列化の効果が大きい。

次に各レンジブロックについて順に最適な  $s_{4p'+q}$  を求める。この処理では、すでに対称ドメインブロックがただ 1 つに特定済みであり、計算量はわずかである。

### 3.3 提案アーキテクチャ

提案手法に基づいてレンジブロックとドメインブロックのマッチングを行うフラクタル画像符号化システムのアーキテクチャを図 3 に示す。ブロックマッチング以外は、文献 [3] の回路を使用する。

モジュール RD の構造を図 4 に示す。RD は、スケーリングされたドメインブロックとレンジブロックの相似度を計算する。RD には、1 クロックあたりレンジブロック中の 1 個の画素値  $\hat{d}_i$  が SRAM から供給される。同時に、スケーリングされたドメインブロック中の 1 個の画素値  $s\hat{a}_i$  も供給される。2 つの値の差を取り、絶

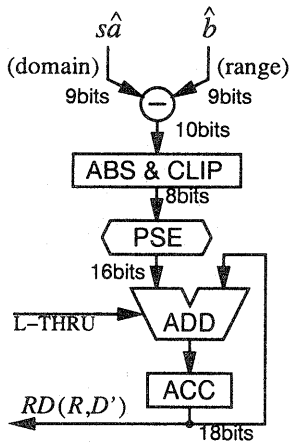


図 4. ブロック相似度計算モジュール RD

対値化および最大値 255(8 ビット) へのクリッピングを行い、疑似二乗回路 PSE に入力する。16 ビットの疑似二乗出力は、18 ビットのアキュムレータ ACC によって積算される。積算値がオーバーフローする場合には、 $2^{18}-1$  にクリッピングする。ブロックサイズが  $8 \times 8$  であるので、最初の画素値を供給してから 64 クロック後に相似度が得られる。

モジュール RD を 8 個用いて、8 通りのスケール係数  $s_{4p}$  ( $p = 0, 1, 2, \dots, 7$ ) について、相似度を並列に計算する。8 個の相似度は、モジュール RDmin によって最小値  $RD_4^m(R, D')$  を定めるとともに、最小値を与えるスケール係数のインデックス  $p'$  を出力する。

1 つのドメインブロックについて最小相似度  $RD_4^m(R, D')$  を求めたら、同一クラスの次のドメインブロックについて同様に画素値を入力して最小相似度を求め、これまでに得られている最小相似度と比較し、最小値を保存する。これを  $DP_R$  に属する全てのドメインブロックについて繰り返すことで、レンジブロック  $R$  に対する最適ドメインブロック  $\hat{D} = D_R$  が得られる。

8 個のモジュール RD およびモジュール RDmin の組を複数用いることで、同一クラスの複数のレンジブロックについて並列に最小相似度  $RD_4^m(R, D')$  を求めることができる。このとき、ドメインブロックの画素値は共通に使用できるので、スケール用乗算器を追加する必要はない(図 3)。

全てのレンジブロックについて、最小相似度  $RD_4^m(R, D')$  を与えるドメインブロック  $D_R$  お

よび 8 通りのスケール係数のインデックス  $p'$  が得られたら、次に各レンジブロックについて、 $s_{4p'}$  を含む前後 7 通りのスケール係数に対して最小相似度  $RD_4^m(R, D')$  を与えるスケール係数を定める。これは、図 3 の例えば左端の RD 群と RDmin のみを用いて、全レンジブロックを順に行う。

このとき、 $D_R$  を求めるために 8 個の乗算器に入力した 8 個のスケール係数  $s_{4p}$  ( $p = 0, 1, 2, \dots, 7$ ) の代わりに、7 個の  $s_{4p'+q}$  ( $q = -3, -2, \dots, 3$ ) を乗算器に入力するだけでよく、ブロックマッチング部分の回路構造を変更する必要はない。残りの 1 個のスケール係数は何でもよく、例えば  $s_0$  を与えればよい。

なお、スケール係数  $s_{4p'}$  に対する相似度を再計算することになるが、これを省略した場合、6 個の  $s_{4p'+q}$  ( $q = -3, -2, -1, 1, 2, 3$ ) に対する最小値と、保存しておいた  $RD_4^m(R, D')$  との比較が必要になり、かえって処理が複雑である。スケール係数  $s_{4p'}$  に対する相似度を再計算することで、最小値を求める処理が RDmin によって一律に行われるとともに  $RD_4^m(R, D')$  の保存も不要となる。

### 3.4 疑似二乗回路

ブロックマッチングにおいては、レンジブロックとドメインブロックの画素値差の二乗をブロック全体で積算した値を相似度とするが、厳密な二乗の計算はハードウェア面積あるいは時間のいずれかの面でコストが高い。そこで、コストが小さく、単なる画素値差の絶対値よりも二乗値に近似した値を出力する疑似二乗回路を用いる。

疑似二乗回路 PSE は、非負 8 ビット整数を入力し、入力値の二乗に近似した 16 ビット整数を出力する。文献 [4] で提案されている疑似二乗回路を変更し、小さな入力値に対しては正確に二乗値を出力し、大きな入力値に対しては疑似二乗値を出力する。

数値シミュレーションと回路規模の検討により、入力値が 31 以下の場合に正確な二乗値を出力する回路とする。8 ビット入力を  $x_7, x_6, \dots, x_0$ 、16 ビット出力を  $d_{15}, d_{14}, \dots, d_0$  とすると、入出力関係は以下の通りである。

$$\begin{aligned} (d_9, d_7, \dots, d_0) &= (x_4, x_3, \dots, x_0)^2 \\ d_{2j} &= x_j x_{j-1} \quad (j = 7, 6, 5) \\ d_{2j+1} &= x_j \quad (j = 7, 6, 5) \end{aligned}$$

表 1 符号化方式と PSNR[dB] の比較

	Lena	Boat	Eltro
解析的手法	25.7581	24.9874	21.3900
SQR	25.9505	25.1011	21.1940
ABS	25.7879	24.9229	21.3140
PSE	25.8716	24.9929	21.0758
PSEL	25.8616	24.9926	21.1636
PSELC	25.8616	24.9926	21.1124

## 4 実験結果

提案手法による符号化データを復号して得られた画像の画質を評価し、提案アーキテクチャの回路規模を検討する。

### 4.1 画質

再生画像の画質は、原画像との PSNR によって評価する。PSNR の定義は以下の通りである。

$$\text{PSNR} = 10 \log_{10} \frac{W \cdot 256^2}{\sum_{k=0}^{W-1} (I_k - J_k)^2} [\text{dB}] \quad (12)$$

ここで、 $I_k$  と  $J_k$  は、それぞれ再生画像  $I$  と原画像  $J$  の同じ位置にある  $k$  番目の画素値とする。画像  $I$  と  $J$  の違いが小さければ、PSNR は大きな値となる。

評価画像として、Lena、Boat、Eltoro を用いた [5]。Boat、Eltoro は画像サイズが  $512 \times 512$  であるが、 $256 \times 256$  にダウンサンプルして使用している。これらの画像はすべてモノクロ、8 ビット階調である。

0 以上 1.2 未満の条件を満足するスケーリング係数  $s_t$  として、インデックス  $t$  ( $t = 0, \dots, 31$ ) に対して  $s_t = 1.2t/32$  で与えられる値を用いた。

表 1 に符号化方式による再生画質の比較を示す。すべてブロックのクラス化を行っている。

解析的手法では、式 (5) により求めたスケーリング係数を  $s_t = 1.2t/32$  に丸めを行っている。以下の 5 つの結果は、すべて探索手法を用いている。レンジブロックとスケーリングされたドメインブロックの差に対して SQR は二乗値、ABS は絶対値、PSE は疑似二乗値を積算して相似度を求めている。PSEL は、疑似二乗値を用いるとともに 3.2 節で述べた最適ドメインブロック探索手法を用いた結果である。PSELC

は、さらに画素値差の疑似二乗値積算の際に 18 ビットへのクリッピングを行った結果である。

表 1 より、Lena、Boat においては、PSE 手法は SQR 手法と ABS 手法の中間の画質を達成することが分かる。Eltoro は全般に再生画質が低く、ABS 手法が最も良い結果となっている。PSE と PSEL の比較では、再生画質の大きな劣化は見られず、3.2 節で述べた最適ドメインブロック探索手法によって、再生画質を犠牲にせずに符号化計算量を削減できることが分かる。PSEL と PSELC の比較では、Lena、Boat において再生画質に差がない。これは、全てのレンジブロックにおいて相似度最小値  $RD_4^m(R, D')$  と  $RD^m(R, D')$  が、いずれも 18 ビット内に収まっていることを示している。16 ビット数を 64 個積算するので、本来は 22 ビットのアキュムレータが必要であるが、18 ビットにすることで再生画質を劣化することなくハードウェア量を削減できる。

以上から、提案アーキテクチャでは、解析的手法に匹敵する再生画質を達成できることが確認できる。

### 4.2 回路規模

ブロックマッチング回路を VHDL で記述し、 $0.5 \mu\text{m}$ 、2 層メタル CMOS ライブラリを用いて Synopsys Design Compiler で論理合成した結果を表 2 に示す。合成結果は、セルの面積総和を 2 入力 NAND ゲートの面積で割った値であり、配線面積は含んでいない。配線遅延を除いたクリティカルパス長は約 25nS である。

PSE は疑似二乗回路であり、43 ゲートと厳密な二乗回路と比べてわずかな回路規模である。RD は図 4 に示す相似度計算回路であり、PSE を含んでいる。MUL はドメインブロック画素値スケーリング用の乗算器であり、9 ビットのドメインブロック画素値  $\hat{a}_s$  (符号付き) と 8 ビットのスケーリング係数  $s$  (符号なし) との乗算を行って、10 ビット符号付きの積を出力する。Range SRAM は、レンジブロックの 64 個の符号付き 9 ビット画素値  $\hat{b}_s$  を記憶する。SRAM にはアドレスデコーダやセンスアンプが必要であるが、配線が規則的でコンパクトにレイアウトできることを考慮して、単純に 1 ビットセルあたり 6 トランジスタ、1.5 ゲート相当と見積もっている。

1 個のレンジブロックについて相似度計算に必要な 8 個のモジュール RD、モジュール RD-

表 2 論理合成結果

モジュール	回路規模 [gate]
PSE	43
RD	355
RDmin	740
8RD+RDmin	3580
MUL	610
Range SRAM	≈860

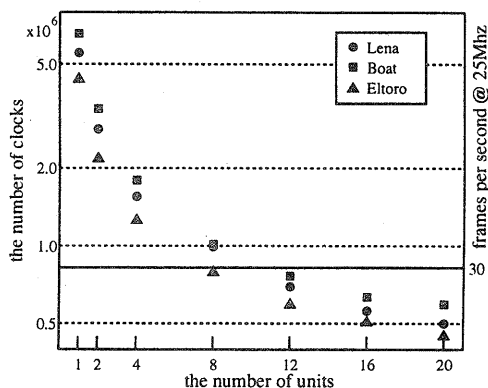


図 5. 相似度計算ユニット数と処理時間

min, SRAM を合わせて、相似度計算ユニットと呼ぶ。3.3 節に示したように、レンジブロックとドメインブロックの組に対して、最小相似度  $RD_4^m(R, D')$  の計算に 64 クロック必要である。  $DP_R$  に属するドメインブロック数を  $N_D$  とすれば、各レンジブロックについて最適ドメインブロックを得るには  $64N_D$  クロックが必要である。

レンジブロックのクラス間分布を考慮した相似度計算ユニット数と処理クロック数の関係を図 5 に示す。この処理クロック数には、レンジブロックを SRAM に転送するクロック数(レンジブロック数  $1024 \times 64$  画素)と、最適ドメインブロックが得られた後に最適スケーリング係数を求めるクロック数(レンジブロック数  $1024 \times 64$  画素)を含んでいる。

回路を 25MHz クロックで動作させたとき、1 枚の画像を 33 ミリ秒で符号化するには、757,575 クロック以内に処理を完了する必要がある。図 5 から、提案アーキテクチャでは 12 個の相似度計算ユニットが必要である。  $s_0 = 0$  とすればスケーリング用乗算器を 7 個に削減で

きることを考慮して、12 個の相似度計算ユニットからなるブロックマッチング回路の規模は約 57,550 ゲートであり、文献 [3] のブロックマッチング回路(約 64,000 ゲート)と比べて小規模である。

なお、上の評価にはクラス化のためのクロック数を含んでいない。クロック数が符号化クロック数の上限を越える場合には、  $DP_R$  の要素数が大きなクラスに対して最適ドメインブロック探索を打ち切るなど、クロック数を制限する操作が必要である。このとき、再生画質が劣化する可能性がある。

## 5 まとめ

疑似二乗ブロックマッチングを用いて、探索手法によりフラクタル画像符号化を行うアーキテクチャを提案した。いくつかの画像例について、解析手法と比べて遜色ない再生画質を達成しつつ、より小さな回路規模で実現できることを確認した。

本研究では、ドメインプール作成の際に、原画像から抽出したドメインブロックについて回転は行うが、反転は行っていない。

レンジブロックの最小相似度がしきい値以上となり、適切なドメインブロックが存在しない場合には、レンジブロックをサブブロックに分割し、各サブブロックごとに符号化を行って画質を向上する手法について検討が必要である。

また、ブロックのクラス化でサブブロックの分散を求める際に二乗計算を必要とするが、これを疑似二乗に置換できるはずである。この場合の再生画質の評価は今後の課題である。

## 参考文献

- [1] A. E. Jacquin, "Image Coding Based on a Fractal Theory of Iterated contractive Image Transformations," *IEEE Trans. Image Proc.*, vol. 1, pp. 18–30, Jan. 1992.
- [2] 松浦 and 永野 and 名古屋, "フラクタル画像圧縮の再構成可能アーキテクチャによる実現法," 信学技報, vol. CPSY98-83, pp. 15–22, 1998.
- [3] 山内 and 武内 and 今井, "フラクタル画像化プロセッサ・アーキテクチャの提案," 第 12 回回路とシステム(軽井沢) ワークショップ, pp. 319–324, 1999.
- [4] 伊藤, "画像符号化におけるブロック相似度計算手法," 情処学システム LSI 設計技術研報, vol. 99SLDM92, pp. 9–16, 1999.
- [5] <http://inls.ucsc.edu/y/Fractals/Images>.