

疑似二乗ブロック相似度による フラクタル画像符号化アーキテクチャ

伊 藤 和 人†

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、多数回実行される二乗計算を簡単な回路で近似することで計算量を低減できる。本研究では、ブロック相似度計算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。

An Architecture for Fractal Image Coding Using Pseudo-square Weighted Block Resemblance

KAZUHIITO ITO†

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. The image block resemblance is measured by the sum of the square of pixel value differences between two blocks. The best affine coefficients can be calculated mathematically but many multiplications are required. In this paper, to reduce the hardware cost, an architecture for fractal image coding is proposed where the best affine coefficients are searched among candidates by using pseudo-square operations in resemblance computation.

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)~3)}。符号化しようとする画像(原画像)をブロックに分割し、ブロックごとに符号化する。各ブロック α に対して、アフィン変換後にブロック α に相似するブロック β を同一原画像中に見出し、ブロック β の画像内位置およびアフィン変換係数をブロック α の符号とする。

フラクタル画像復号化では、符号中の位置情報に基づいて現在の画像からブロックを抽出し、これに符号中の係数によるアフィン変換を施したブロックを集めて次の画像を構成する。任意の初期画像から開始してこの処理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、一般に画像が多数のブロックに分割され、各ブロックに対して相似するブロックの候補が多数あり、さらに各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が必要なためである。

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^{4)~10)}であり、これを解析的手法と呼ぶ。もう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。

解析的手法と探索手法の計算量を比べると解析的手法の方が少なく、汎用プロセッサやDSPによるソフトウェア符号化には解析的手法が適している。しかし、クロック周波数 850 MHz の PC-UNIX 計算機による符号化時間は 1 秒程度であり、クロック周波数 200 MHz の VLIW 型 DSP ではメモリアクセスがネックとなって符号化に少なくとも 100 ミリ秒以上要するなど、毎秒 30 画像の動画の実時間符号化はソフトウェアのみでは実現できない。そこで、解析的手法に基づいて、処理高速化を目指した符号化ハードウェアアーキテクチャ^{4)~6)}や、さらにハードウェア量低減を図ったアー

† 埼玉大学工学部
Faculty of Engineering, Saitama University

キテクチャ^{7)~10)}も提案されている。

探索手法では、解析的手法に比べて乗算回数が少ないが、ブロックとアフィン変換後のブロックとの間の相似度算出のためにブロック間画素値差の二乗計算が多数必要となる。この二乗計算の目的は、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、あるブロックに最も相似するブロックを見つけることである。したがって、最適な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^{(11),(12)}。二乗計算を近似することで個々の演算回路を簡略化し、その一方で符号化処理の並列度を上げることで、解析的手法を用いた場合よりも少ないハードウェア量で高速な符号化が行えると考えられる。

ところが、これまで探索手法に基づいた符号化ハードウェアは提案されていない。そこで本稿では、探索手法に基づいて、ハードウェアによる並列処理に適したフラクタル画像符号化手法を提案する。また、この符号化手法を処理時間とハードウェア量の面で効率良く並列処理するためのアーキテクチャを提案する。フラクタル画像符号化のハードウェア化において最もハードウェア量が多くなると予想されるブロックマッチング中の相似度計算に疑似二乗回路を用いた回路を設計し、ハードウェア量を評価する。

2章では、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つの手法を紹介する。3章ではハードウェアによる並列処理に適した探索手法に基づく符号化手法およびその並列処理アーキテクチャを提案する。実験結果を4章に示す。

2. フラクタル画像符号化

既存のフラクタル画像符号化処理³⁾を説明する。符号化の原理については文献1), 2), 10)に述べられており、本稿では省略する。

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、原画像を互いに重なり合わない複数のブロックに分割する。これをレンジブロック(以下Rブロック)と呼ぶ。Rブロックのサイズは横 N 画素、縦 N 画素とする。また、原画像を縦横ともRブロックの2倍のサイズのブロックに分割する。文献1)ではこのブロックをドメインブロックと呼ぶが、本稿では大ドメインブロックと呼ぶ。大ドメインブロックのサイズは横 $2N$ 画素、縦 $2N$ 画素であり、大ドメインブロックどうしの重なり合いを許す。

各大ドメインブロックを縮小(ダウンサンプル)し

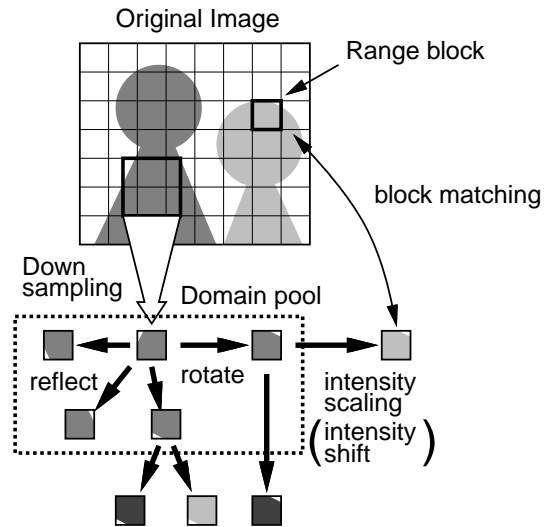


図1 フラクタル画像符号化
Fig. 1 Fractal image coding.

て、Rブロックと同サイズにしたものをドメインブロック(以下Dブロック)と呼ぶ。Dブロックを集めてドメインプールを構成する。各D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Dブロックとしてドメインプールに含める。

ドメインプールの中から、各Rブロックに最も相似したDブロックを選択するのであるが、相似度の評価の前にD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、Dブロック D の i 番目の画素値 a_i は式(1)のように変換される。

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

ここで、 a'_i は D にアフィン変換を施したDブロック D' の i 番目の画素値を表す。

Rブロック R とアフィン変換後のD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。Rブロック R の i 番目の画素値を b_i とすれば、相似度 $RD(R, D')$ は式(2)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' は相似性が高く, $RD(R, D') = 0$ ならば R と D' は完全に一致する.

R ブロック R と D ブロック D の組に対して, 相似度 $RD(R, D')$ を最小にする最適アフィン変換係数が存在する. このときの最小相似度を $RD^m(R, D')$ とする. そして, 各 R ブロック R について, $RD^m(R, D')$ を最小にする最適 D ブロックをドメインプールの中から選択する. すなわち, フラクタル画像符号化では, 各 R ブロックについて, 特定の D ブロックに対し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 D 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.

最適化処理の結果得られたアフィン変換係数および D ブロックの位置と回転の情報が各 R ブロックの符号となる.

2.2 アフィン変換係数最適化

R ブロック R , D 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 とする値であり, 式 (3), (4) で計算できる.

$$s = \frac{n \sum a_i b_i - \sum a_i \sum b_i}{n \sum a_i^2 - (\sum a_i)^2} \quad (3)$$

$$o = \frac{\sum b_i - s \sum a_i}{n} \quad (4)$$

このとき, $RD(R, D')$ の最小値 $RD^m(R, D')$ は式 (5) で与えられる.

$$RD^m(R, D') = \frac{n \sum b_i^2 - (\sum b_i)^2}{n \sum a_i^2 - (\sum a_i)^2} s^2 \quad (5)$$

スケール係数 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^{7),13)}.

R ブロック R , D ブロック D が与えられたとき, 最適なアフィン変換係数は式 (3) を用いて直接算出できる. これを本稿では解析的手法と呼ぶ. 解析的手法では, $\sum a_i$ や $\sum b_i$ などは各 D ブロックと R 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 D ブロックと R 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. これを本稿では探索手法と呼ぶ. D ブロックと R 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

(4) 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.

$$RD(R, D') = \sum_{i=0}^{n-1} (s\hat{a}_i - \hat{b}_i)^2 \quad (6)$$

ここで $\hat{a}_i = a_i - \bar{a}$ であり, D 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 (6) 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 を特定するうえで o の値が不要であることを意味している. 最適な s を特定した後に式 (4) を用いて o を算出する.

2.3 最適ドメインブロック探索

各 R ブロックに対して, ドメインプールの中から最適な D ブロックを選び出す探索処理をブロックマッチングと呼ぶ. 一般にドメインプールには多数の D ブロックが含まれるので, ブロックマッチングに多数の相似度計算を必要とする. 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に必要な相似度計算回数を削減することが有効である.

R ブロック, D 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^{10),14)} する. 以降では, R ブロック, D 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.

$$A_k = \frac{1}{n_4} \sum_{i=0}^{n_4-1} x_i^k \quad (7)$$

$$V_k = \frac{1}{n_4} \sum_{i=0}^{n_4-1} (x_i^k - A_k)^2 \quad (8)$$

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.

ブロックマッチングの際は, 注目する R ブロックと同一クラスの D ブロックのみを対象として行うことで, ブロックマッチングの組合せ数を低減する.

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.

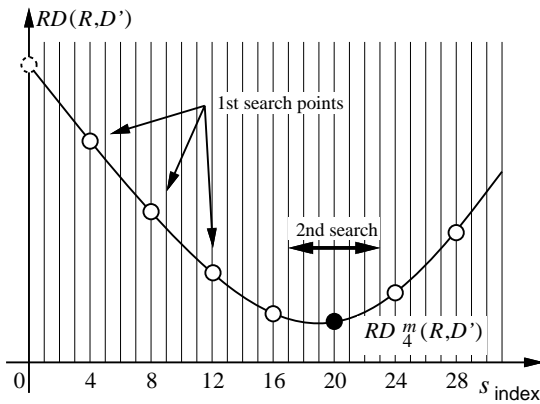


図 2 スケーリング係数 s と相似度 $RD(R, D')$ の関係
Fig. 2 The relation between the scaling factor s and resemblance $RD(R, D')$.

3. ブロックマッチングアーキテクチャ

本章では、相似度計算回数を低減するため、与えられた R ブロック、D 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手法、および、各 R ブロックについて最適な D ブロックを選択する手法を提案する。また、提案手法を処理時間とハードウェア量の面で効率良く実行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 R ブロック R と D ブロック D の組に対して、探索手法では、32 通りのスケーリング係数の中から、相似度最小（相似性最大）の最適スケーリング係数を求める。

ここで、調べるスケーリング係数の数を削減し、相似度計算回数を低減する手法を提案する。

式 (6) より、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。これら 32 通りの s は等間隔であるとする。最適スケーリング係数探索に必要な相似度計算回数を減らすため 2 段階の探索を行う。第 1 探索として、まず 8 通りの s_{4p} ($p = 0, 1, 2, \dots, 7$) について式 (6) に従って相似度 $RD(R, D')$ を計算し、8 通りの $RD(R, D')$ のうちの最小値を与える $s_{4p'}$ 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 $RD(R, D')$ のうちの最小値を与える s は、 $s_{4p'+q}$ ($q = -3, -2, -1, 0, 1, 2, 3$) の中に存在する。

そこで、次に第 2 探索として 6 通りの $s_{4p'+q}$ ($q = -3, -2, -1, 1, 2, 3$) について相似度 $RD(R, D')$ を計

算し、これらの最小値を定めるとともにその最小値を与える $s_{4p'+q}$ を得る。なお、 $4p'+q < 0$ の場合には相似度計算は行わない。第 2 探索で得た最小相似度を第 1 探索で得た最小相似度と比較し、前者が小さければ $s_{4p'+q}$ 、後者が小さければ $s_{4p'}$ が求める最適スケーリング係数である。

第 1 探索の結果得られる s_{4p} に対する相似度最小値を $RD_4^m(R, D')$ 、32 通りのすべての s に対する相似度最小値を $RD^m(R, D')$ と表す。

ところで、 $s_0 = 0$ の場合は 2.1 節で述べたように相似度は D ブロックと無関係に計算できる。 $s_0 = 0$ に対する相似度は R ブロック R に対して 1 回だけ計算すればよく、ドメインプール中の各 D ブロックに対して毎回計算する必要はない。したがって、 $s_0 = 0$ の場合には第 1 探索では 7 通りの s_{4p} ($p = 1, 2, \dots, 7$) について相似度を計算すればよい。

3.2 最適ドメインブロック選択手法

ドメインプール中で R ブロック R とクラスが同一の D ブロックの集合を DP_R と表す。

与えられた R ブロック R について、相似性最大の最適な D ブロックは、 DP_R に属する D 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 と表す。ここで、 D_R 選択の計算量削減のため、次の最適化手法を提案する。

まず、 DP_R 中の各 D ブロック D について、第 1 探索を行って最小相似度 $RD_4^m(R, D')$ を求める。次に、すべての D ブロック $D \in DP_R$ について、最小の $RD_4^m(R, D')$ を与える D ブロック \hat{D} を定める。これが最適 D ブロック D_R であるとする。また、このときのスケーリング係数 $s_{4p'}$ を定める。

これにより最適 D ブロック D_R は選択されるが、 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、 D_R について第 2 探索を行って、6 通りの $s_{4p'+q}$ ($q = -3, -2, -1, 1, 2, 3$) に対する最小相似度を計算し、相似度 $RD_4^m(R, D'_R)$ ($q = 0$ の場合に相当する) と比較して $RD_4^m(R, D'_R)$ を求める。

上に述べた最適 D ブロック選択手法は、同一クラスに属する R 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。第 1 探索においてスケーリングを施した D ブロックデータを共通に使用し、R ブロック R に対する D_R と $s_{4p'}$ の選択を並列に行う。この処理は、同一クラスの多数の R ブロックと多数の D 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、かつ実際に並列化の効果が高い。

次に各 R ブロックについて順に第 2 探索を行って

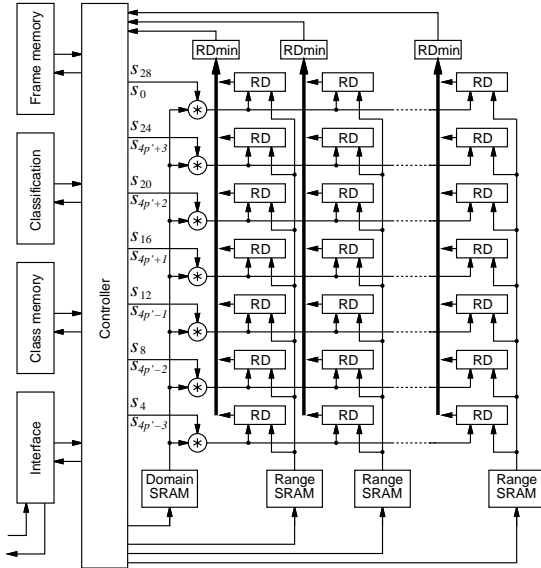


図3 フラクタル画像符号化アーキテクチャ
Fig. 3 Fractal image coding architecture.

最適な $s_{4p'+q}$ を求める．この処理では，すでに対象 D 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 R ブロックと D 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．D ブロックのスケール係数は 32 通りとし， $s_0 = 0$ とする．ブロックマッチング以外は，文献 10) の回路を使用する．

ブロックマッチング処理は以下のように行う．まず R ブロックの N^2 個の画素値 (\hat{b}_i) を SRAM に転送するとともに，D ブロックの画素値も同様に別の SRAM に転送する．これらの SRAM は 1 クロックあたり 1 画素値を出力する．第 1 探索では 7 通りのスケール係数 s_{4p} ($p = 1, 2, \dots, 7$) について，7 個の乗算器を用いて並列に D ブロックの画素値をスケールし，7 個の相似度計算モジュール RD によって並列に相似度を計算する．最小値モジュール RDmin は，7 個の RD の計算結果を入力して最小値 $RD_4^m(R, D')$ 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 D 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 D ブロックを SRAM に転送し，同様に最小相似度を定めるとともにスケール係数インデックスを出力する．コントローラは，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 に属するすべての D ブロ

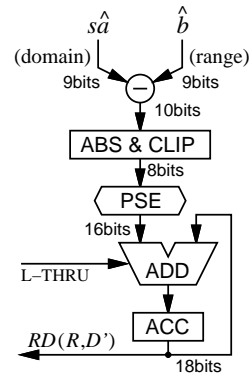


図4 ブロック相似度計算モジュール RD
Fig. 4 Block resemblance calculation module RD.

ックについて繰り返すことで，R ブロック R に対する最適 D ブロック D_R が得られる．

7 個の相似度計算モジュール 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 R 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，D 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．

すべての R ブロックについて，最小相似度 $RD_4^m(R, D')$ を与える D ブロック D_R およびスケール係数インデックス p' が得られたら，次に第 2 探索を行う．これは，図 3 の，たとえば左端の RD 群と RDmin のみを用いて，全 R ブロックを順に行う．

第 2 探索では， $s_{4p'}$ の前後 6 通りのスケール係数 $s_{4p'+q}$ ($q = -3, -2, -1, 1, 2, 3$) および $s_0 = 0$ に対して相似度を計算する．これは第 1 探索で使用した 7 個のスケール係数 s_{4p} ($p = 1, 2, \dots, 7$) の代わりに，6 個の $s_{4p'+q}$ と 0 をそれぞれの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．

第 2 探索後，コントローラはモジュール RDmin が出力する $s_{4p'+q}$ ($q = -3, -2, -1, 1, 2, 3$) と $s_0 = 0$ に対する最小相似度を，保存しておいた $RD_4^m(R, D')$ と比較し，最適スケール係数 s_i を決定する．

3.4 相似度計算モジュール

相似度計算モジュール RD の構造を図 4 に示す．RD は，R ブロックとスケールされた D ブロックの相似度を計算する．RD には，1 クロックあたり R ブロック中の 1 個の画素値 \hat{b}_i が SRAM から供給される．同時に，スケールされた D ブロック中の 1 個の画素値 $s\hat{a}_i$ も供給される．2 つの値の差をとり，絶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 $N \times N$ 画素であるので、最初の画素値を供給してから N^2 クロック後に相似度が得られる．

3.5 疑似二乗回路

ブロックマッチングにおいては、R ブロックと D 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と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．ブロックの相似度が小さい（相似性が高い）場合に誤差が大きくなることを防ぐため、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．8 ビット入力を x_7, x_6, \dots, x_0 、16 ビット出力を $d_{15}, d_{14}, \dots, d_0$ とし、入力値が $2^L - 1 (L = 1, \dots, 8)$ 以下の場合に正確な二乗値を出力するとき、入出力関係は以下のとおりである．

$$\sum_{j=0}^{2L-1} 2^j d_j = \left(\sum_{i=0}^{L-1} 2^i x_i \right)^2$$

$$d_{2j} = x_j x_{j-1} \quad (L \leq j \leq 7)$$

$$d_{2j+1} = x_j \quad (L \leq j \leq 7)$$

ここで $L = 8$ は厳密な二乗に一致し、 L が小さくなるほど回路規模は減少する代わりに二乗値との差は大きくなる．後述の画像 Lena について、3.1 節、3.2 節の符号化手法を用いた場合の PSE ゲート数と再生画質の関係を図 5 に示す．図中で N は R ブロックのサイズを表す．ゲート数を少なくするため L を小さくするほど画質は低下する．図には示していないが、例に用いた他の画像についても同様の傾向である．回路の検討により、以降では $L = 5$ を用いる． $L = 5$ の場合の PSE 入出力特性を図 6 に示す．図中で SQR は厳密な二乗値 x^2 を表す．

4. 絶対値回路の簡略化

疑似二乗回路の入力データは非負整数でなければならないが、入力データの絶対値化を行う．2 の補数表現を用いている場合の絶対値化は、データの符号を調べ、負ならば全ビットを反転して最下位ビットに 1 を加えるが、ここで加算器を必要とする．

ところで、前述のようにブロックマッチングは相似

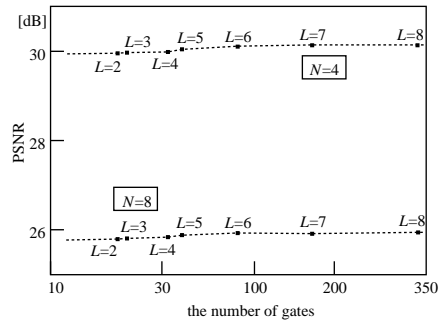


図 5 疑似二乗回路 PSE のゲート数と再生画質の関係
Fig. 5 The number of gates of the pseudo-square circuit PSE vs. the decoded picture quality.

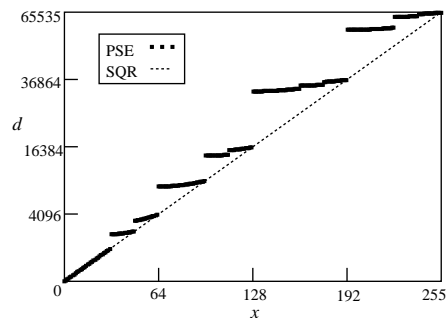


図 6 疑似二乗回路 PSE の入出力特性 ($L = 5$)
Fig. 6 Input-output characteristics of the pseudo-square circuit PSE ($L = 5$).

度を厳密に求めることではないので、絶対値化も厳密に行う必要はないと考えられる．そこで疑似二乗回路の入力データが負の場合に単に全ビットの反転だけを行う．入力データが負の場合に真の絶対値より 1 だけ小さな値が疑似二乗回路に入力されることになるが、1 を加えるための加算回路を省略できる．

5. 実験結果と考察

提案手法による符号化結果を復号して得られる再生画質を評価して提案符号化手法の有効性を調べる．次に、提案する符号化アーキテクチャのハードウェア量を調べる．

5.1 画 質

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する．再生画像の画質は、原画像との PSNR によって評価する．PSNR の定義を式 (9) に示す．

$$\text{PSNR} = 10 \log_{10} \frac{W \cdot 256^2}{W-1 \sum_{k=0}^{W-1} (I_k - J_k)^2} [\text{dB}] \quad (9)$$

表 1 最適化方式と PSNR[dB] の比較 ($N = 8$)Table 1 Comparison of PSNR [dB] over optimization method ($N = 8$).

最適化方式	Lena	Boat	Eltro
解析的手法 (実数係数)	25.7575 26.2742	25.0667 25.3325	22.3737 22.6253
SQR	26.0007	25.1298	22.3773
SQRL	25.9869	25.1088	22.3959
ABS	25.8008	24.9730	22.1810
ABSL	25.8081	24.9367	22.1759
PSE	25.9042	25.0052	22.2733
PSEL	25.8968	24.9791	22.2340
PSELC	25.8968	24.9791	22.1710
PSEpm	25.9325	25.0326	22.2920
PSELpm	25.9092	25.0373	22.2361
PSELCpm	25.9092	25.0373	22.2361

表 2 最適化方式と PSNR[dB] の比較 ($N = 4$)Table 2 Comparison of PSNR [dB] over optimization method ($N = 4$).

最適化方式	Lena	Boat	Eltro
解析的手法 (実数係数)	29.8157 31.0183	29.7946 30.7777	26.2568 27.5642
SQR	30.1790	30.1323	26.6812
SQRL	30.1512	29.9761	26.6670
ABS	29.9037	29.8248	26.3984
ABSL	29.8271	29.7378	26.3504
PSE	30.0898	30.0954	26.5379
PSEL	30.0439	29.9206	26.5295
PSELC	30.0439	29.9206	26.5295
PSEpm	30.1010	30.0836	26.5471
PSELpm	30.0387	30.0483	26.5309
PSELCpm	30.0387	30.0483	26.5309

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, Eltro を用いた¹⁵⁾。Boat, Eltro は画像サイズが 512×512 であるが、 256×256 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, 2 に符号化方式による再生画質の比較を示す。表 1 は $N = 8$, 表 2 は $N = 4$ の場合の結果である。すべてブロックのクラスタ化を行っている。

解析的手法では、式 (3) により求めたスケーリング係数を $s_t = 1.2t/32$ に丸めている。解析的手法の PSNR がその他の探索手法の結果よりも低い原因は、探索手法があらかじめ量子化された係数の中から PSNR 最大になるものを選択するのに対して、探索手法は式 (3)、

(4) により求めた最適実数アフィン変換係数を量子化する際に誤差が加わることである。比較としてアフィン変換係数を実数とし、復号化も実数計算を行った場合 (実数係数) には、PSNR は最大値を達成する。

以降はすべて探索手法を用いている。R ブロックとスケーリングされた D 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。SQRL, ABSL, PSEL は、それぞれ SQR, ABS, PSE に 3.2 節で述べた最適 D 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。PSEpm, PSELpm, PSELCpm は、それぞれ PSE, PSEL, PSELC に対して絶対値回路の簡略化を行った場合の結果である。

表 1, 2 より、すべての画像において PSE は SQR と ABS の中間の画質を達成することが分かる。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 D 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く、Eltro の場合も差はわずかである。これは、すべての R 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。 $N = 8$ の場合は 16 ビット数を $N^2 = 64$ 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。

さらに、PSELC と PSELCpm を比較すると、再生画質が向上する場合と劣化する場合があるが、再生画質の大きな劣化なく絶対値回路を簡略できることが分かる。ここで PSELCpm の方が再生画質が高い場合を調べると、画像中の R ブロックごとの画質が PSELC に比べて高いブロックが低いブロックよりも多くなっている。これは、あらかじめ D ブロック, R ブロックの画素値から減じる平均画素値の丸めと、負の画素値差の絶対値が 1 だけ増加して相似度を計算することが相乗して、PSELC と比べてより適切なスケーリング係数が得られる R ブロックが画像中に多かったためと考えられる。しかし、逆に PSELCpm では再生画質が劣化する場合もあり、つねに PSELCpm が PSELC よりも優れているとはいえない。

以上から、提案アーキテクチャでは、解析的手法に匹敵し、完全な探索を行った場合の探索手法から大きく再生画質を損なうことなく符号化できることが確認できる。

表 3 論理合成結果
Table 3 Synthesis results.

モジュール	回路規模 [gate]
PSE	43
RD	348
RDpm	318
RDmin	680
7RDpm+RDmin	2906
MUL	610
SRAM	≒860

5.2 回路規模

提案アーキテクチャの回路規模を検討する．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 3 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 25 nS である．

疑似二乗回路 PSE のゲート数は 43 ゲートであり，厳密な二乗回路 (337 ゲート)，8 ビット \times 8 ビット乗算器で二乗を計算する場合 (507 ゲート) 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．RDpm は RD から絶対値を簡略化したモジュールである．MUL は D ブロック画素値スケール用の乗算器であり，9 ビットの D ブロック画素値 \hat{a}_i (符号付き) と 8 ビットのスケール係数 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．SRAM は， $N = 8$ の場合にブロックの 64 個の符号付き 9 ビット画素値 \hat{b}_i 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． $N < 8$ の場合の符号化しか行わないのであれば，SRAM のビット数は相応して減少できる．

1 個の R ブロックについて相似度計算に必要な 7 個のモジュール RD，モジュール RDmin，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，R ブロックサイズ $N = 8$ のとき，R ブロックと D 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 クロックを要する． DP_R に属する D ブロック数を N_D とすれば，各 R ブロックについて最適 D ブロックを得るには $64N_D$ クロックが必要である．

R 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 7 に示す．この処理クロック数には，R ブロックを SRAM に転送するクロック数 (R ブロック数 1024×64 画素) と，最

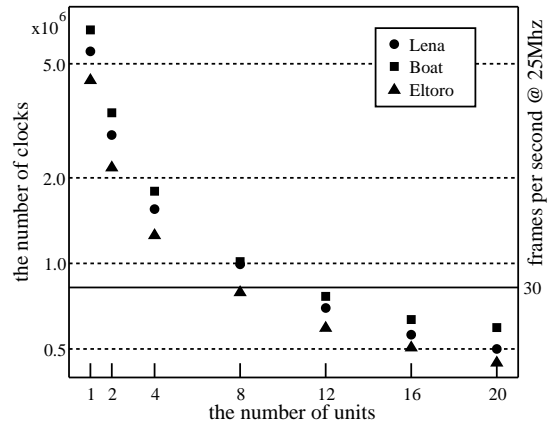


図 7 相似度計算ユニット数と処理時間

Fig. 7 The relation between the number of resemblance calculation units and processing time.

表 4 ブロックマッチング回路の比較 ($N = 8$)

Table 4 Comparison of block-matching units ($N = 8$).

アーキテクチャ	回路規模 [gate]
文献 10)	約 64,000
提案	約 50,400

適 D ブロックが得られた後に最適スケール係数を求めるクロック数 (R ブロック数 1024×64 画素) を含んでいる．

回路を 25 MHz 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 7 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． $s_0 = 0$ とすればスケール用乗算器を 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 50,400 ゲートである．表 4 に示すように，文献 10) のブロックマッチング回路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超える場合には， DP_R の要素数が大きなクラスに対して最適 D 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．

提案アーキテクチャでは，スケール係数 s が 32 通りであることに基づいて相似度計算ユニットあたりの RD モジュール数が 7 個に最適化されている．画像符号フォーマットを変更してより多くの s 候補値が利用できる場合に，解析的手法では式 (3) で求めた s の量子化を変更するだけで済む．一方探索手法に基づく提案アーキテクチャでは，処理速度を維持するた

めには RD モジュール数を増やす必要がある。そのため、 s の候補数によっては探索手法に基づくアーキテクチャに比べて提案アーキテクチャのハードウェア量が大きくなることもある。

6. ま と め

ブロックマッチングの際の相似度計算に疑似二乗を用いて、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、相似度計算に厳密な二乗計算を用いた場合や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符号化が実現できることを確認した。

従来のアーキテクチャと比べて提案アーキテクチャでは、構成単位である相似度計算ユニットが比較的小規模であり、これの使用個数の増減により処理速度とハードウェア量のトレードオフを柔軟に行うことができる。また、回路は変更せずに相似度計算モジュールを制御して画素値差疑似二乗値の積算回数を変更するだけでより小さなブロックに対応できる。そのため、レンジブロックの最小相似度がしきい値以上となって適切なメインブロックが存在しない場合に、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階層符号化¹⁾にも容易に対応可能である。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。

謝辞 本研究を進めるにあたり貴重な助言をいただき、また評価にご協力いただいた埼玉大学末次亮氏、伊藤仁氏に感謝する。

参 考 文 献

- 1) Jacquin, A.E.: Image Coding Based on a Fractal Theory of Iterated contractive Image Transformations, *IEEE Trans. Image Proc.*, Vol.1, No.1, pp.18–30 (1992).
- 2) Jacquin, A.E.: Fractal Image Coding: A Review, *Proc. IEEE*, Vol.81, No.10, pp.1452–1465 (1993).
- 3) Wohlberg, B. and Jager, G.: A Review of the Fractal Image Coding Literature, *IEEE Trans. Image Proc.*, Vol.8, No.12, pp.1716–1729 (1999).
- 4) He, Z.L., Liou, M.L. and Fu, K.W.: VLSI Architecture for Real-Time Fractal Image Coding, *Proc. ISCAS'96*, pp.332–336 (1996).
- 5) Acken, K.P., Kim, H.N., Irwin, M.J. and

Owens, R.: An Architectural Design for Parallel Fractal Compression, *Proc. ASAP'96*, pp.3–11 (1996).

- 6) 李 信行, 阿曾弘具: フラクタル画像圧縮の VLSI アーキテクチャ, *信学技報*, Vol.CPSY98-82, pp.9–22 (1998).
- 7) 松浦昭洋, 永野秀尚, 名古屋彰: 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).
- 8) Nagano, H., Matsuura, A. and Nagoya, A.: An Efficient Implementation Method of a Metric Computation Accelerator for Fractal Image Compression Using Reconfigurable Hardware, *IEICE Trans. Fundamentals*, Vol.E84-A, No.1, pp.372–377 (2001).
- 9) 山内英樹, 武内良典, 今井正治: 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).
- 10) Yamauchi, H., Takeuchi, Y. and Imai, M.: VLSI Architecture for Real-Time Fractal Image Coding Processors, *IEICE Trans. Fundamentals*, Vol.E83-A, No.3, pp.452–458 (2000).
- 11) 伊藤和人: 画像符号化におけるブロック相似度計算手法, *情報処理学会システム LSI 設計技術研究報告*, Vol.99SLDM92, pp.9–16 (1999).
- 12) 伊藤和人: 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, *情報処理学会システム LSI 設計技術研究報告*, Vol.00SLDM97, pp.39–46 (2000).
- 13) Fisher, Y.: *Fractal Image Compression: Theory and Application*, Springer Verlag, New York (1995).
- 14) Wein, C.J. and Blake, I.F.: On the Performance of Fractal Compression with Clustering, *IEEE Trans. Image Proc.*, Vol.5, No.3, pp.522–526 (1996).
- 15) <http://inls.ucsc.edu/y/Fractals/Images>.

(平成 13 年 9 月 25 日受付)

(平成 14 年 3 月 14 日採録)



伊藤 和人(正会員)

昭和 39 年生。平成 4 年東京工業大学大学院理工学研究科電気・電子工学専攻博士課程修了。博士(工学)。平成 4 年より東京工業大学工学部助手。平成 7 年より埼玉大学工学部助教授。VLSI 上流設計自動化, VLSI 信号処理の研究に従事。IEEE, 電子情報通信学会各会員。