

画像符号化におけるブロック相似度計算手法

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 多くの静止画像や動画の符号化では、画像を複数の画像ブロックに分割し、ある1つのブロックに最も相似したブロックを探索するブロックマッチングが行われる。ブロックの相似度を計算するため、2つのブロックの各画素値の差の絶対値あるいは二乗値の累計を用いるが、二乗値計算は時間とハードウェアの面で不利であり、絶対値による相似度は復号画像の画質が劣る点で不利である。ブロックマッチングの目的は最も良く似たブロックを探索することであり、画素値差の二乗を正確に計算することが目的ではない。本研究では、良質な符号化を達成し、計算コストの小さいブロック相似度の計算手法を提案する。

キーワード 画像符号化、フラクタル符号化、MPEG、ブロック相似度、ブロックマッチング

A Method to Compute Image Block Resemblance for Image and Video Coding

Kazuhito Ito

Department of Electrical and Electronic Systems, Saitama University

255 Shimookubo, Urawa, Saiama 338-8570, Japan

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

Abstract In most image and video coding algorithms, an image is divided into many blocks. Then block matching is performed to search the block which best resembles a block to be coded. The block resemblance is measured usually by an accumulation of either the simple absolute or the square of the difference of pixel values. The disadvantage of the square is high cost on computation time and the amount of hardware. The disadvantage of the absolute is low quality of the decoded image. In this paper a block resemblance computation method is proposed to achieve as good coding as the square resemblance without much increase of cost compared to the absolute resemblance.

Keywords image coding, fractal coding, MPEG, block resemblance, block matching

1 まえがき

静止画像や動画は一般にデータ量が多く、保存に際しては大きな記憶容量、通信に際しては高速な通信路を必要とする。画像を圧縮してデータ量を減らし、記憶容量や通信速度を低減するため、静止画像や動画の符号化手法が提案されている [1, 2, 3, 4]。その多くの手法では、画像を複数の画像ブロック (以下単にブロックと呼ぶ) に分割し、各ブロックを符号化する。

ブロックの符号化では、現在注目するブロックが、予め用意された一定個数のブロック集合の中のどのブロックに最も似ているかを検索することがある。これをブロックマッチングと呼ぶこととする。例えば静止画像フラクタル符号化では、符号化しようとする画像自身のブロックをブロック集合としてブロックマッチングを行う。また MPEG2 や H.263 動画画像符号化では、動きベクトル検出のために過去あるいは未来の画像のブロックをブロック集合としてブロックマッチングを行う。

ブロックマッチングでは、符号化しようとするブロック $block$ と、ブロック集合 BLK 中の比較対象ブロックとの間に相似度を定義する。相似度は任意のブロックのペアに対して非負の値をとり、相似度の値が小さいほど2つのブロックが似ていると考える。ブロック集合 BLK 中の比較対象ブロック数を N とする。 BLK 中の N 個の比較対象ブロック blk_m ($m = 0, 1, \dots, N-1$) のうち、ブロック $block$ との相似度が最小となる m を定めることでブロックマッチングを行う。

ブロックの相似度として、以下の式によって計算される値が使用されている。

$$\sum_{k=0}^{B-1} diff(\alpha_k, \beta_k) \quad (1)$$

ここで、 B はブロックの画素数、 α と β は相似度を計算しようとする2つのブロックである。 α_k と β_k はそれぞれ α と β の k 番目の画素値であり、2つのブロックの相等しい位置にある画素の輝度を表す。関数 $diff(x, y)$ として、一般的には

$$diff(x, y) = (x - y)^2 \quad (2)$$

または

$$diff(x, y) = |x - y| \quad (3)$$

が用いられている。式 (2) を用いる相似度を二乗相似度、式 (3) を用いる相似度を絶対値相似度

と呼ぶこととする。ブロックマッチングとその結果を用いた符号化実験結果を比較すると、符号化の品質の点で二乗相似度は絶対値相似度よりも優れていることがわかる。しかし、二乗相似度では、2つの画素値の差の二乗を行うために乗算器または ROM を用いた表引きを必要とし、計算に要するコストが絶対値相似度よりも大きい。そのため、計算コストを優先し、符号化の質を犠牲にして絶対値相似度を使用している場合がある。

ところで、ブロックマッチングの目的は最も良く似たブロックを探索することであり、二乗相似度や絶対値相似度の最小値を正確に計算することが目的ではない。乗算器や ROM よりも簡単な回路によって絶対値相似度よりも優れた相似度を計算できれば、計算コストを大きく増加することなく良質の符号化が行えると考えられる。本研究では、良質な符号化を達成し、計算コストの小さいブロック相似度の計算手法を提案する。

2 静止画像および動画の符号化手法

ブロックマッチングを行う静止画像符号化および動画画像符号化を簡単に紹介する。

2.1 ベクトル量子化による画像符号化

まず、符号化しようとする原画像 I に合わせて最適化されたコードブック CB_I を用意する。コードブックは、比較的少数のブロックの集合であり、そのブロック数を N とする。符号化では、原画像 I を互いに重ならない複数のブロック分割し、各ブロック $block$ に最も似たブロック cb_m ($m = 0, 1, \dots, N-1$) を CB_I 中で検索する。検索の結果、 $block$ に最も似たブロック cb_m のインデックス m をブロックの $block$ の符号とする。復号化では、符号化に使用したコードブックと同じコードブック CB_I を共有し、符号 m が指すブロック $cb_m \in CB_I$ を並べることで画像を再生する。このように、ベクトル量子化による画像符号化方式では、極めて簡単な処理で画像を復号できる利点がある。

再生画像の画質は、コードブックの品質とブロックマッチング結果に依存する。コードブックは LBG アルゴリズム [1] を用いて作成する。LBG アルゴリズムでは、トレーニングブロック集合と初期コードブックを用い、トレーニングブ

ロック集合とコードブックとの相似度に基づいてトレーニングブロック集合のクラスタ化およびコードブック更新を繰り返すことで、トレーニングブロック集合に特化して最適化されたコードブックを得る。

原画像の各ブロックに対して、コードブック中から最も似たブロックを検索する計算量も大きいですが、LBG アルゴリズムを用いて原画像について最適なコードブックを作成する計算量も大きなものとなる。

2.2 フラクタル画像符号化

フラクタル符号化 [2] では、まず符号化しようとする原画像 I をレンジブロックと呼ぶ互いに重ならない正方形のブロックに分割する。また、原画像 I の中に縦横ともレンジブロックの 2 倍の大きさの正方形のブロックを定義し、隣接する 4 画素の平均値を求めることでダウンサンプリングしたブロックをドメインブロックと呼ぶ。ドメインブロックの大きさはレンジブロックと同じであるが、ドメインブロック同士は重なりを許す。ドメインブロックの集合をドメインプールと呼ぶ。

レンジブロックは、ブロック中の画素値の分布状態によってシェードブロック、ミッドレンジブロック、エッジブロックの 3 種類に分類される。

シェードブロックは、画素値に大きな分散がなく、すべて同一の画素値の画素を並べることで近似する。この結果、シェードブロックは 1 つの画素値によって符号化される。

ミッドレンジブロックは、画素値に分散があるが、その偏りに明確な方向性のないブロックである。ミッドレンジブロックは、当該レンジブロックを最も良く近似するドメインブロックの座標、レンジブロックとドメインブロックの平均画素値の差、画素値の分散のスケーリング係数の組によって符号化される。ドメインブロックの座標およびスケーリング係数は、レンジブロックとドメインプールとのブロックマッチングによって検索する。

エッジブロックは、画素値に分散があり、その偏りに方向性のあるブロックである。ドメインプール中の各ドメインブロックについて 0 度、90 度、180 度、270 度の回転およびそれぞれを左右反転した 8 通りのドメインブロックについて、レンジブロックとブロックマッチングを行う。その結果、エッジブロックは、レンジブロックを最も良く近似するドメインブロックの座標およ

び回転度数と反転の有無、レンジブロックとドメインブロックの平均画素値の差、画素値の分散のスケーリング係数によって符号化される。

復号化では、 I と同サイズの任意の画像を初期画像 $\tilde{I}(0)$ とする。画像 $\tilde{I}(p)$ に基づいて符号にしたがって画像 $\tilde{I}(p+1)$ 中のレンジブロックを復号する。 $p = 0, 1, 2, \dots$ として復号処理を繰り返し施すと、符号化が適切ならばやがて画像 $\tilde{I}(p)$ は原画像 I に近い画像 \tilde{I} に収束する。

再生画像の画質は、レンジブロックとドメインプールのブロックマッチング結果に依存する。

フラクタル画像符号化では、再生画像にレンジブロックの境界があまり目立たないこと、原画像を拡大した画像の再生が可能であり、その際にノイズが発生しないなどの利点がある。しかし、ドメインブロックに重なりを許すためドメインブロック数が多く、さらにスケーリング係数の多様性とエッジブロックの場合には回転と反転も考慮するため、符号化におけるブロックマッチングの計算量が膨大になる。

2.3 動画画像符号化

動画画像は、静止画像がある時間間隔で連なったものである。H.263 動画画像符号化方式 [4] では、動画画像を構成する静止画像を INTRA 画像と INTER 画像に分類する。

INTRA 画像の符号化では、まず画像を縦 16 画素、横 16 画素の大きさで互いに重ならないマクロブロックに分割する。各マクロブロックは、縦 8 画素、横 8 画素の 4 個のブロックに分割される。各ブロックの 64 個の画素値データに対して二次元離散コサイン変換 (DCT) を施し、64 個の空間周波数成分を求める。周波数成分を量子化した結果をジグザグスキャンして 64 個の一次元データに整列する。この一次元データの中で、連続するゼロデータの個数と非ゼロデータのデータ値の組をエントロピー符号化することで、ブロックの符号を得る。

INTER 画像の符号化では、各マクロブロックについて、すでに復号済みの参照画像の中で最も似た縦 16 画素、横 16 画素の領域 R を検索する。この領域の座標とマクロブロックの座標の差を動きベクトルと呼び、動きベクトルを検索する処理を動き予測と呼ぶ。マクロブロック中の各画素値から、 R 中の対応する位置の画素値を減じる。これを動き補償と呼ぶ。この後の処理は INTRA 画像と同じである。INTER 画像は、動きベクトルのエントロピー符号および周波数

成分のエントロピー符号によって符号化される。

低速な通信回線による比較的low解像度の動画通信を主目的とする H.263 と、比較的高解像度の動画を対象とする MPEG(MPEG2) 符号化方式では、符号化処理の細部は異なるが、基本的な符号化の考えは同じである。

符号ビット数と再生画像の画質は量子化ステップに依存する。量子化ステップが大きければ、周波数成分の精度が低下することで再生画像の画質が劣化するが符号ビット数は減少する。量子化ステップが小さければ、周波数成分の精度低下が小さく、再生画像の画質は向上するが符号ビット数は増加する。

INTRA 画像の画質は、周波数成分の量子化ステップのみに依存する。一方、INTER 画像の画質は、周波数成分の量子化ステップおよび動き補償結果に依存する。良好な動き補償の結果、ブロック中の多くの画素値が 0 となれば、DCT によって算出した周波数成分の非ゼロデータ数が少なくなる。そのため、符号ビット数を増加せずに小さな量子化ステップを用いることが可能となり、その結果再生画像の画質を向上する。

H.263 符号化では、動きベクトルの範囲は、半画素を単位として上下および左右にそれぞれ -16.0 画素以上+15.5 画素以内と規定されている。動き予測は、符号化しようとする画像中の各マクロブロックと再生済みの参照画像中の $64^2 = 4096$ 個 (あるいはその一部) の領域とのブロックマッチングによって行う。動画では、各 INTER 画像の符号化を画像間時間以内に完了させなければならないため、動き予測の高速化が重要である。

3 ブロック相似度

まず、原画像と再生画像の近似度の評価尺度である PSNR を定義し、PSNR が良好であり、かつ計算コストの小さいブロック相似度計算手法を検討する。

3.1 PSNR

画像を構成する画素数を W とする。画像 I と画像 J の Peak Signal-to-Noise Ratio (PSNR) を以下のように定義する。

$$\text{PSNR} = 10 \log_{10} \frac{W \cdot 256^2}{\sum_{k=0}^{W-1} (I_k - J_k)^2} [\text{dB}] \quad (4)$$

表 1 相似度計算手法と PSNR

画像	符号化に用いた相似度		
	ABS	SQR	CUBE
SIZUE-vq	30.250	30.401	29.485
LENA-f	26.766	26.956	26.793

表 2 ベクトル量子符号化仕様

ブロックサイズ:	4×4 画素
コードブロックサイズ:	64 ブロック
トレーニングブロック集合:	全ブロック

ここで、 I_k と J_k は、それぞれの画像において k 番目の同じ位置にある画素値とする。画像 I と画像 J の違いが小さければ、PSNR は大きな値となる。

3.2 最適ブロック相似度

式 (1) のブロック相似度計算において、2 つのブロック内で同位置にある画素値の相似度は、式 (2) あるいは式 (3) によって計算する 경우가多い。まず、原画像と再生画像の PSNR を最大化するためには、ブロックマッチングにおいてどのような相似度を用いるべきであるか考察する。

表 1 は、図 1(a) に示す動画画像 SUZIE の第 0 画像についてベクトル量子化による画像符号化を行う場合、ならびに図 2 に示す画像 LENA についてフラクタル画像符号化を行う場合に、原画像と復号によって得られた再生画像との PSNR を表している。ベクトル量子符号化で行うブロックとコードブロックとの間のブロックマッチング、およびフラクタル画像符号化で行うレンジブロックとドメインプールとの間のブロックマッチングにおいて、相似度計算には 3 通りの計算式を用いた。表中で、ABS は式 (3) による絶対値相似度、SQR は式 (2) による二乗相似度、CUBE は以下の式

$$\text{diff}(x, y) = |x - y|^3 \quad (5)$$

による三乗相似度をそれぞれ用いた場合を意味する。

ベクトル量子符号化の仕様とフラクタル符号化の仕様は、それぞれ表 2 と表 3 に示す。

表 1 より、ベクトル量子符号化とフラクタル符号化の両方で、絶対値相似度よりも二乗相似度を用いた方が PSNR の点で良好な再生画像が得られることが分かる。画素値の差が大きいほどより強力にブロック相似度を悪化させるため、



(a) (b)

図 1. 動画像 SUZIE(画像サイズ 176 × 144)
(a) 第 0 画像. (b) 第 60 画像.



図 2. 静止画像 LENA(画像サイズ 256 × 256).

表 3 フラクタル符号化仕様

レンジブロックサイズ:	8 × 8 画素
ドメインブロック元サイズ:	16 × 16 画素
ドメインブロックの座標ステッ	4 画素
ブ数:	
スケール係数: ミッドレンジブロックの場合	0.7, 0.8, 0.9, 1.0 から選択
エッジブロックの場合	0.2, 0.3, ..., 0.9 から選択

より良好なブロックマッチングが行えると予想された三乗相似度は、実際には両方の符号化方式において二乗相似度よりも PSNR は低い。三乗計算は、乗算器を用いると 2 回乗算が必要であり、ROM を用いた表引きを行ったとしても三乗の結果のビット数が長くなるためブロック中の全ての画素に対する加算の際に不利である。また、画素値の差の 1.5 乗や 2.5 乗などの計算は、絶対値計算や二乗計算に比べてはるかに複雑である。

以上のことから、ブロックマッチングには二乗相似度が適していることがわかる。

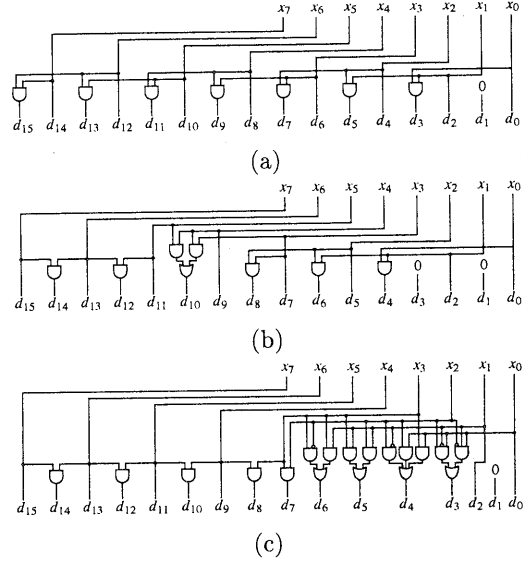


図 3. 疑似二乗計算回路. (a) PSE(1).
(b) PSE(2). (c) PSE.

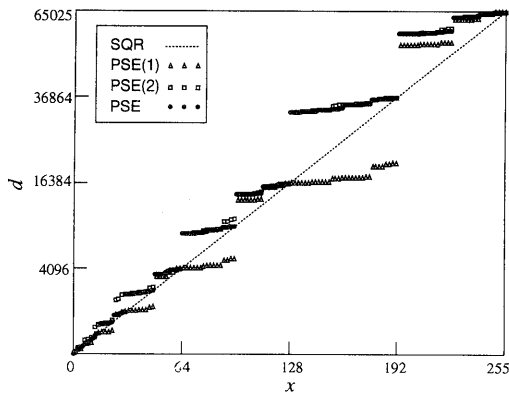
3.3 疑似二乗計算回路

二乗相似度の計算では、画素値の差 x に対してその二乗値 x^2 を求める必要がある。二乗計算は、乗算器を用いて乗数と被乗数ともに x を入力して積として得る方法と、アドレス x にデータとして x^2 を記憶する ROM を参照する方法が考えられるが、半導体チップ上ではどちらも大きな面積を必要とする。

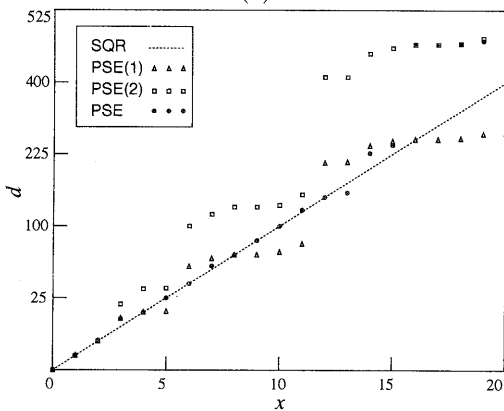
ブロックマッチングでは、正確に二乗相似度を計算することが目的ではなく、二乗相似度を最小とするブロックを検索することが目的である。したがって、ある程度二乗相似度に似た値が計算できれば、二乗相似度を用いた場合と同等に良好なブロックマッチングが行えると考えられる。

図 3 に、8 ビットの非負の整数 x を入力して、 x^2 に近い 16 ビットの整数 d を出力する 3 通りの論理回路を示す。これらの回路の入出力関係を図 4 に示す。図 4(b) は、図 4(a) のうち $x \leq 20$ について拡大表示したものである。図中の破線は、 $d = x^2$ を表す。

図 3(a) は PSE(1) と名付ける回路であり、3 つの回路の中では最も簡単な構成をしている。入力 x の値によっては出力 d が x^2 を大きく下回る。図 3(b) は PSE(2) と名付ける回路であり、PSE(1) に若干の変更を加えている。どの x にお



(a)



(b)

図 4. 疑似二乗計算特性. (a) 全体図. (b) 拡大図.

いても $d \geq x^2$ を満足するように工夫されているが、小さな x (10 以下) において x^2 との差が大きい。図 3(c) は PSE と名付ける回路であり、特に $x < 16$ において出力 d が x^2 とほぼ等しくなるように工夫されている。いずれの回路も論理段数は 3 段程度であって遅延時間は小さく、厳密に二乗値を出力する回路よりもハードウェア量は少ないと予想される。

提案する疑似二乗計算回路の有効性を調べるため、ベクトル量子符号化により実験を行った結果を表 4 に示す。表には、原画像と再生画像の間の PSNR の dB 値が示されている。ベクトル量子符号化実験に用いた画像は図 1(a)、符号化の諸元は表 2 の通りである。

表 4 からは、縦方向には LBG アルゴリズムによってコードブックを作成する際に用いた相類似度計算手法の違いによる PSNR の変化、横方向には同じコードブックを用いた場合に符号化の

際に用いた相類似度計算手法の違いによる PSNR の変化を読み取ることができる。表中で SQR は厳密な二乗値を用いて相類似度を計算、ABS は絶対値を用いて相類似度を計算、PSE(1)、PSE(2)、PSE はそれぞれの疑似二乗計算回路を用いて相類似度を計算することを意味する。

表 4 より、PSE(2) は PSE(1) よりも若干回路が複雑であるにもかかわらず、PSE(1) に対して PSNR は小さいか、わずかに大きい程度である。PSE は、すべてのコードブックについて SQR に次ぐ良好な PSNR を達成している。小さな画素値差 x に対して、疑似二乗出力 d が x^2 に近いことが有効であることが分かる。

3.4 疑似二乗計算回路と絶対値について

図 3 に示す疑似二乗計算回路が二乗に近い値を出力するためには、回路入力 x は非負整数でなければならない。したがって、画素値差 x が負である場合には、2 の補数回路を通して $-x$ が疑似二乗回路に入力されるようにする。2 の補数回路は、入力データの全ビットの論理を反転し、最下位ビットに 1 を加えるので、加算器と同程度の遅延時間を要する。

ここで、 x が負の場合に厳密に 2 の補数によって値 $-x$ を求めるのではなく、1 の補数すなわち全ビットの単なる論理反転のみで近似する疑似絶対値手法を検討する。この実験結果を表 5、表 6 に示す。表 5 は、図 1(a) に対するベクトル量子符号化の結果である。表の縦方向はコードブック作成の際に用いる相類似度計算手法、表の横方向は符号化の際に用いる相類似度計算手法である。PSEpm は、疑似絶対値手法を PSE に組み合わせたことを意味する。表 6 は、3 つの画像 LENA (図 2)、SUZIE (図 1(a))、CLAIRE (図 5(a)) について、ブロックマッチングに ABS、SQR、PSE、PSEpm を用いてフラクタル符号化を行った結果を示している。表中の値は、原画像と再生画像の間の PSNR の dB 値である。

ベクトル量子符号化では、コードブック作成、符号化とも疑似絶対値を用いることで再生画像の PSNR が低下している。一方フラクタル符号化では再生画像の PSNR はほとんど変化がない。疑似絶対値は、符号化方式によって有効である場合と無効である場合がある。

表 4 疑似二乗計算回路を用いた相類似度計算手法と PSNR

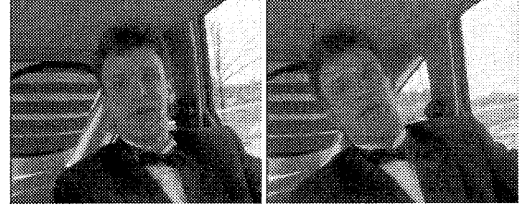
LBG	符号化に用いた相類似度				
	SQR	PSE(1)	PSE(2)	PSE	ABS
SQR	30.401	30.350	30.327	30.382	30.250
PSE	30.475	30.429	30.427	30.464	30.355
PSE(1)	30.462	30.420	30.423	30.423	30.305



(a)

(b)

図 5. 動画像 CLAIRES(画像サイズ 176 × 144) (a) 第 0 画像. (b) 第 130 画像.



(a)

(b)

図 6. 動画像 CARPHONE(画像サイズ 176 × 144) (a) 第 0 画像. (b) 第 130 画像.

表 5 疑似絶対値と PSNR(ベクトル量子符号化)

LBG	符号化に用いた相類似度		
	SQR	PSE	PSEpm
SQR	30.401	30.382	30.374
PSE	30.475	30.464	30.453
PSEpm	29.213	30.411	30.402

表 6 相類似度計算方式および疑似絶対値と PSNR の関係 (フラクタル符号化)

画像	符号化に用いた相類似度			
	ABS	SQR	PSE	PSEpm
LENA	26.766	26.956	26.815	26.847
SUZIE	30.057	30.328	30.095	30.095
CLAIRE	32.929	33.158	33.782	33.782

4 動画像符号化への応用

H.263 動画像符号化における動き予測は、ブロックマッチングによって実現されている。そこで、このブロックマッチングに提案する疑似二乗相類似度を応用した結果を示す。

実験に用いた動画像は、いずれも画像サイズ 176 × 144(QCIF) の 4:2:0 カラー動画像であり、それぞれ SUZIE(図 1)、CLAIRE(図 5)、CARPHONE(図 6) と名付ける。すべての動画像について、先頭から 150 枚の画像を H.263 符号化方

式によって符号化した。SUZIE は、画面全体に大きく顔が表示されており、途中で大きく顔を振る動作をする。CLAIRE は、画面の中央に人物がおり、口および顔が動くものの、全体としてはほとんど動きがない。CARPHONE は、人物が言葉を発しており、途中で大きく体全体を振る動作をする。

実験では、先頭の第 0 画像のみ INTRA 画像 (I picture)、第 1 画像から第 149 画像は INTER 画像 (P picture) として符号化した。INTER 画像の中では、すべてのマクロブロックについて動き補償を行うこととした。第 0 画像は、DCT 後の周波数成分を量子化係数 $Q = 13$ によって量子化する。その他の画像は、動き補償後のブロックを DCT して得られた周波数成分を $Q = 2$ (あるいは $Q = 3$) から $Q = 31$ までのいずれかの値に固定した量子化係数によって量子化する。

各量子化係数について、第 1 画像から第 149 画像までの 1 画像あたりの平均符号ビット数を横軸、原画像と再生画像の平均 PSNR を縦軸に表したグラフを図 7 に示す。便宜上グラフは折れ線で描かれているが、実際にはこの線状の離散的な値のみが有効である。グラフには、それぞれ相類似度計算手法として絶対値相類似度 (ABS)、二乗相類似度 (SQR)、疑似二乗相類似度 (PSE、図 3(c)) を用いて疑似絶対値は使用しない) を用いた場合の結果を示している。グラフはいずれも左下隅から右上隅に向かう直線状となっている。グラフ

の左下隅は平均符号ビット数が小さいがPSNRが小さく再生画像が劣悪、右上隅はPSNRが大きく再生画像が優良であるが平均符号ビット数は大きい。量子化係数が大きくなれば左下隅、量子化係数が小さくなれば右下隅へ向かう。

グラフより、PSEはSQRと同等の結果を示し、ABSはグラフが右下へずれていることが分かる。これは、同じPSNRを達成するならばPSEはABSよりも平均符号ビット数が小さくて済み、同じ平均符号ビット数を達成するならばPSEはABSよりもより大きなPSNRが得られることを表している。

以上の結果より、動画画像符号化における動き予測に疑似二乗相似度を用いることで、絶対値相似度に比べて大きく計算コストを増加することなく、二乗相似度に匹敵する符号品質が得られることが分かる。

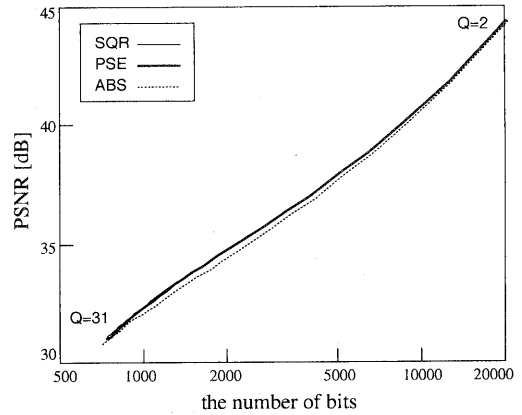
5 むすび

本研究では、画像ブロックの相似度を低コストで計算する手法を提案した。フラクタル画像符号化、ベクトル量子化による画像符号化、H.263動画画像符号化における動き予測に提案手法を適用したところ、わずかなハードウェア量の追加で画素値差の単純な絶対値による相似度よりも優れ、画素値差の二乗による相似度に匹敵する符号結果が得られ、提案手法の有効性を確認した。

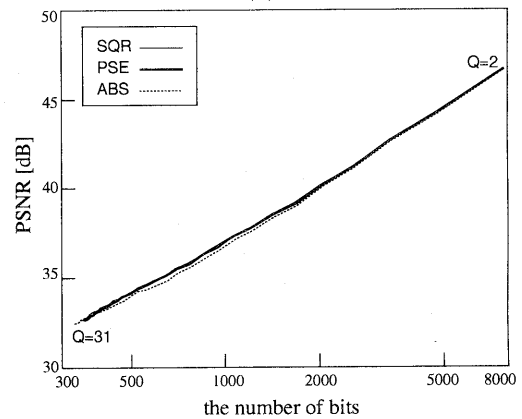
フラクタル符号化では、文献 [5] に述べられているようにドメインブロックの画素値分散スケール係数を探索ではなく解析的に算出する方法があり、この手法との計算量や処理速度の比較は今後の課題である。

参考文献

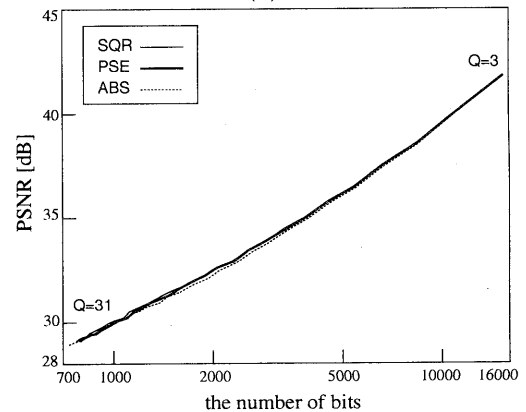
- [1] N. M. Nasrabadi and R. A. King, "Image Coding Using Vector Quantization: A Review," *IEEE Trans. Comm.*, vol. 36, pp. 957-970, Aug. 1988.
- [2] A. E. Jacquin, "Image Coding Based on a Fractal Theory of Iterated contractive Image Transformations," *IEEE Trans. Image Proc.*, vol. 1, pp. 18-30, Jan. 1992.
- [3] 藤原, ポイント図解式最新 MPEG 教科書. アスキー, 1994.
- [4] ITU-T Draft H.263, Sep. 26, 1997.
- [5] 松浦, 永野, 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.



(a)



(b)



(c)

図 7. H.263 動画画像符号化結果. (a) SUZIE. (b) CLAIRE. (c) CARPHONE.