

長岡技術科学大学研究報告 第23号( 2001 )

## カラー画像に対するマルチスケールリングフラクタル符号化

古 沢 竜 志・中 川 匡 弘

# カラー画像に対するマルチスケーリングフラクタル符号化

古 沢 竜 志\*・中 川 匡 弘\*

Multi-Scaling Fractal Coding for Color Images.

Ryuji FURUSAWA\*, Masahiro NAKAGAWA\*

For the image data compression, several fractal image coding schemes have been investigated so far utilizing the similarity and redundancy in an original image. In this report, we shall apply the multi-scaling fractal coding for color images. So as to decrease the errors involved in the self-affine contraction mappings and to make the reproduced edge more clear, we shall divide the range block into two regions with the aid of the Block Truncation Coding scheme. From the computer simulations for the standard color images, we may confirm the great advantage of our new coding scheme beyond the conventional one.

Key words : Fractal Coding, Color Image Compression, Self-similarity, Multi-Scaling

## 1. ま え が き

画像は膨大なデータ量を有するため、通信コストの削減や、蓄積媒体の節約のために、圧縮技術は必要不可欠なものとなっている。現在、多くの画像符号化手法が提案されているが、その中のひとつに、フラクタル符号化がある。フラクタル符号化は、2値のフラクタル図形を生成するシステムである、Iterated Function Systems (IFS) の逆問題としてBarnsleyにより提案され<sup>1),2)</sup>、Jacquinにより濃淡画像へと拡張された<sup>3),4)</sup>。その後も多くの検討が加えられ、符号化性能の改善が進められているが<sup>5),6),7)</sup>、記録・蓄積メディアへの利用といった、実用に耐え得るだけの性能はまだ得られていない。

フラクタル符号化は、画像の持つ部分的自己相似性を冗長性として捉えることにより画像を圧縮するもので、符号化するブロックを良く近似する自己相似性を画像内から探索し、自己相似性を表現する縮小変換の変換パラメータを符号として用いる。この符号化は、ブロック間にまたがる構造的な特徴を利用するため、ブロック歪が目立ちにくいという特徴をもっている。しかしながら、自然画像は厳密なフラクタル性を有しているわけではないので、その縮小変換は近似的なものとなるため、フラクタル符号化の符号化特性は、画像の持つ自己相似性に強く依存する。そのため、複雑な構造を持った画像などでは、一般に充分な近似度を

持った縮小変換を探索することが困難となり、高い再現性が得られなかった。そこで我々は、マルチスケーリング縮小写像を用いたフラクタル符号化手法を提案した<sup>8)</sup>。この手法は、縮小変換をブロック単位で行っていた従来手法に対して、Mitchellらにより提案されたBlock Truncation Coding (BTC)<sup>9)</sup>の手法を適用し、画像の構造的な特徴によりブロックを領域分割して、領域ごとに変換パラメータを与えるものである。それにより、より近似度の高い縮小変換の探索を得ることができ、また、画像内の物体の区画が明確となるため、鮮明にエッジを再現することが可能となる。

本論文では、マルチスケーリングフラクタル符号化のカラー画像に対する適用について検討を行う。まず、第2節においてマルチスケーリングを用いたフラクタル符号化手法について述べ、第3節でマルチスケーリングフラクタル符号化のカラー画像への応用について検討する。そして、第4節で、提案手法の有効性について検討する。

## 2. マルチスケーリングフラクタル符号化

### 2.1 基本原理

一般に、自然画像は、部分的に近似的な自己相似性を有しており、画像内のある一部分は、別な一部分を縮小することによって近似することができる。フラクタル符号化では、この性質を利用して、画像を分割し、各部分毎に最適な近似を与える縮小変換を求め、その変換パラメータを符号とする。

ここで、原画像を $\mu_{org}$ 、再生画像を $v$ とする。任意の

原稿受付：平成13年5月25日

\*長岡技術科学大学電気系

画像  $\mu_1, \mu_2$  に対して、変換  $(\cdot)$  が次式で表現される縮小性、

$$d(\mu_1, \mu_2) \leq s \cdot d(\mu_1, \mu_2), \quad 0 \leq s < 1, \quad (1)$$

を満足するならば、原画像と再生画像との誤差は、コラージュの定理<sup>2)</sup>により、

$$d(\mu_{org}, \nu) \leq \frac{1}{1-s} d(\mu_{org}, (\mu_{org})), \quad (2)$$

の不等式により規定される。ここで、 $d(\cdot; \cdot)$  は 2 画像間の距離を示し、 $d(\mu_{org}, \nu)$  は原画像と再生画像の誤差、 $d(\mu_{org}, (\mu_{org}))$  は縮小変換による近似誤差と考えることができる。このことは、充分な縮小性と充分な相似性を満たす変換が与えられたならば、原画像を充分良く近似する再生画像に収束することを保証し、また、より近似度の高い変換が与えられるほど、良好な再生画像が得られることを示している。つまり、フラクタル符号化の画像再現性は、いかに近似精度の高い縮小変換を探索できるかにかかっているといえる。

従来のフラクタル符号化は、画像を重ねりの無いレンジブロック  $R_{ij}$  に分割し、各レンジブロックに対応するサイズの大きなドメインブロック  $D_{kl}$  を探索し、最適な近似を与える変換  $\lambda_{ij}(\cdot)$  を求めることにより行われる (Fig. 1)。この変換  $\lambda_{ij}(\cdot)$  は、縮小アフィン変換により表され、以下の式によって記述される。

$$\begin{aligned} R_{ij}(x, y) &= R_{ij}(x, y) - R_{ave}, \\ D_{ij}(x, y) &= (s(D_{kl}(x, y)) - D_{ave}), \quad (3) \\ R_{ij}(x, y) &\cong \lambda_{ij} \cdot D_{kl}(x, y) \end{aligned}$$

ここで、 $R_{ij}$  と  $D_{kl}$  は、それぞれ平均値を分離されたレンジブロックとドメインブロック、 $R_{ave}$  はレンジブ

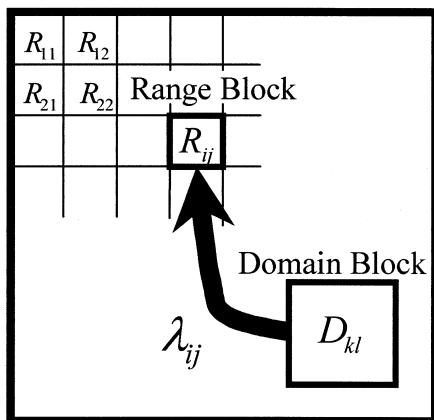


Fig. 1 Transformation for domain to range.

ックの平均値、 $D_{ave}$  はドメインブロックの平均値、 $(\cdot)$  は回転と対角線に対する反射からなる等長変換、 $s(\cdot)$  は縮小変換、 $\lambda$  はスケーリング係数を示す。 $\lambda$  は、最小自乗法により、以下の式によって求められる。

$$\lambda = \frac{\sum_{x,y \text{ Block}} R_{ij}(x, y) \cdot D_{kl}(x, y)}{\sum_{x,y \text{ Block}} D_{kl}(x, y)^2}, \quad (4)$$

しかしながら、あまり自己相似性を有していない画像や、複雑な構造を持った画像に対しては、 $\lambda$  をひとつしか用いない従来のシングルスケーリングでの変換では、相似性を探索することが困難となり、良好な画像再現性を実現することは困難であった。特に、ブロック内に輪郭部分などのような激しい輝度格差があるような場合には、輪郭がなまり、不鮮明となってしま

う。そこで、我々は、縮小変換の近似精度を向上させるため、マルチスケーリングを用いたフラクタル符号化手法を提案した<sup>8)</sup>。マルチスケーリングでは、ブロックを領域分割し、各領域に個別にスケーリングパラメータを与えることにより、より近似度の高い縮小変換を与えることが可能となる。それにより、従来手法では高い再現性が得られなかった画像に対しても、良好な再生画像が得られるものと期待される。

## 2.2 マルチスケーリング手法

マルチスケーリングの基本的な原理は、ブロックを BTC の手法を適用することにより領域分割し、各領域に対して独立したスケーリング特性を与えるというものであり、縮小変換の近似度の向上と、エッジの再現性の向上を目的とする。なお、本論文では、マルチスケーリングによるレンジブロックの分割数を 2 とする。

### 2.2.1 マルチスケーリングによる縮小変換

マルチスケーリングを用いた符号化は、以下のように行われる。まず、ブロック内をハイレベル領域とローレベル領域とに分割する。このとき、ブロック内のグレイレベルの平均値を求め、平均値以上である画素をハイレベル領域、平均値未満である画素をローレベル領域と設定する。そして、ハイレベル領域には 1 を、ローレベル領域には 0 を与え、ビットプレーン  $B_{ij}$  を作成する (Fig. 2)。

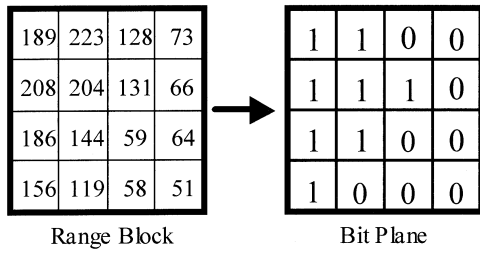


Fig. 2 (a) Making bit plane.



Fig. 2 (b) Bit plane

$$B_{ij}(x, y) = \begin{cases} 1, & R_{ij}(x, y) \geq R_{ave} \\ 0, & R_{ij}(x, y) < R_{ave} \end{cases} \quad (5)$$

更に、縮小変換、平均値分離、等長変換を施したドメインブロック  $D'_{ij}$  に対して、ビットプレーンを当てはめ、レンジブロックと同形状の、ハイレベル領域とローレベル領域とに分割する。そして、最小自乗法により求められた以下の式によって、ハイレベル領域とローレベル領域に対するスケーリング係数  $\alpha_H$ 、 $\alpha_L$  が求められる。

$$\alpha_H = \frac{\sum_{x,y}^{High} R_{ij}(x, y) \cdot D_{kl}(x, y)}{\sum_{x,y}^{High} D_{kl}(x, y)^2}, \quad (6)$$

$$\alpha_L = \frac{\sum_{x,y}^{Low} R_{ij}(x, y) \cdot D_{kl}(x, y)}{\sum_{x,y}^{Low} D_{kl}(x, y)^2}. \quad (7)$$

これらの2種類のスケーリング係数  $\alpha_H$  と  $\alpha_L$  を用いることにより、式(3)は、

$$R'_{ij}(x, y) = \begin{cases} R_{ij}(x, y) - R_{Have} & B_{ij}(x, y) = 1, \\ R_{ij}(x, y) - R_{Lave} & B_{ij}(x, y) = 0, \end{cases}$$

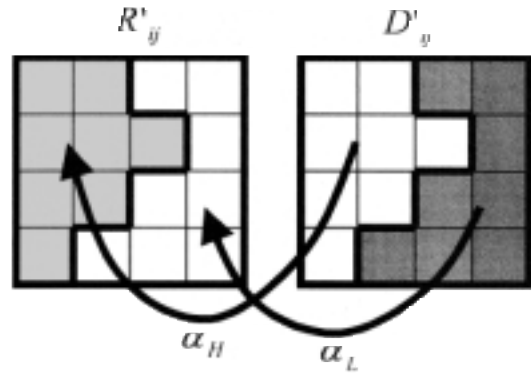


Fig. 3 Transformation with

$$D_{ij}(x, y) = \begin{cases} (s(D_{kl}(x, y)) - D_{Have}) & B_{ij}(x, y) = 1, \\ (s(D_{kl}(x, y)) - D_{Lave}) & B_{ij}(x, y) = 0, \end{cases}$$

$$R_{ij}(x, y) = \begin{cases} \alpha_H \cdot D_{kl}(x, y), & B_{ij}(x, y) = 1, \\ \alpha_L \cdot D_{kl}(x, y), & B_{ij}(x, y) = 0, \end{cases} \quad (8)$$

と書かれる (Fig. 3)。ここで、 $R_{Have}$  は、レンジブロックのハイレベル領域の平均値、 $R_{Lave}$  は、レンジブロックのローレベル領域の平均値、 $D_{Have}$  は、ドメインブロックのハイレベル領域の平均値、 $D_{Lave}$  は、ドメインブロックのローレベル領域の平均値である。

マルチスケーリングでは、ブロックごとに、ブロックの平均値  $R_{Have}$ 、 $R_{Lave}$ 、スケーリング係数  $\alpha_H$ 、 $\alpha_L$ 、等長変換、ドメインブロックの位置  $(k, l)$ 、そして、ビットプレーン  $B_{ij}$  が符号として用いられる。

### 2.2.2 ビットプレーンの符号化

マルチスケーリングでは、ハイレベルとローレベルの領域情報として、ビットプレーン  $B_{ij}$  を必要とするが、そのままではビットプレーンのみでも 1 [bpp] の符号を必要とする。これは、画像符号化において大きな負担となる。そこで、効率的にビットプレーンの表現のためのビットを削減する必要がある。

そこで今回は、ビットプレーンを2値のベクトルとして考え、ベクトル量子化を行った。まず、全ブロックからビットプレーンを求め、そのヒストグラムを作成する。そして、出現頻度の高いパターンを初期符号帳として用い、LBGアルゴリズム<sup>10)</sup>によって最適な代表ベクトルを作成する。代表ベクトルの個数を64とした場合、ビットプレーンのビットレートは、0.4~0.5 [bpp]まで、削減することが可能となる。

### 2.3 符号化手法

マルチスケーリングは、十分な近似を得ることので

きなかったブロックに対して、近似度を高め、エッジの再現性を向上させることを目的としている。そのため、マルチスケーリング処理は、強いコントラストを含むレンジブロックに対して行われるのが効果的であると考えられる。従って、ここでは、ブロック内の分散<sup>2</sup>を求め、高い分散を持つブロックに対して優先的にマルチスケーリングを適用する。

まず、ブロック判別のために、画像をレンジブロック $R_{ij}$ に分割し、全ブロックごとに分散<sup>2</sup>を求める。このブロックの中から、あらかじめ設定された閾値 $Z_b$ 以下のブロックをシェードブロックと判別する。シェードと判別されたブロックは分散が低く、ブロック内を平均値のみで十分に近似できるので、平均値のみを符号として用い、自己相似性の探索は行わない。これにより、符号化時間とビットレートとを削減することができる。そして、閾値 $Z_b$ 以上のブロックをエッジブロックと判別する。エッジと判別されたブロックを、更に分散<sup>2</sup>に従って一定の比率で2分する。<sup>2</sup>の低い集合はシングルスケーリングブロックと判別され、従来用いられてきた縮小変換によって自己相似性を探索する。一方、<sup>2</sup>の高い集合はマルチスケーリングブロックと判別され、我々が提案したマルチスケーリングを用いた縮小変換によって自己相似性を探索する。そして、最終的には、総当り的に縮小変換を探索し、最も自乗誤差の少なかった縮小変換の変換パラメータを符号として用いる。

復号処理は、任意の初期画像を用いてドメインブロックからレンジブロックへの変換を繰り返し施すことにより実現される。以下に、その手順を述べる。

まず、任意の初期画像を用意し、レンジブロック $R_{ij}$ に分割する。続いて、各 $R_{ij}$ に対応するドメインブロック $D_k$ を抽出し、それを、シングルスケーリングブロックは(9)式で、マルチスケーリングブロックは(10)で示される変換によって、 $R_{ij}$ に置き換える。

$$R_{ij} = \cdot (s(D_k(x,y)) - D_{ave}) + R_{ave} \cdot \quad (9)$$

$$R_{ij} = \begin{cases} H \cdot (s(D_k(x,y)) - D_{Have}) + R_{Have}, & B_{ij} = 1 \\ L \cdot (s(D_k(x,y)) - D_{Lave}) + R_{Lave}, & B_{ij} = 0 \end{cases} \quad (10)$$

また、シェードブロックに対しては、 $R_{ij} = R_{ave}$ と置き換える。以上の処理を、全てのレンジブロックに対して実行し、第一回目の復号処理とする。更に、その復号された画像に更なる変換を繰り返すことによって、画像のグレイレベルが収束し、再生画像が得られる。

Table 1 Setting of coding.

$R_{ave}$	$k, l$		
6 bits	10 bits	4 bits	3 bits

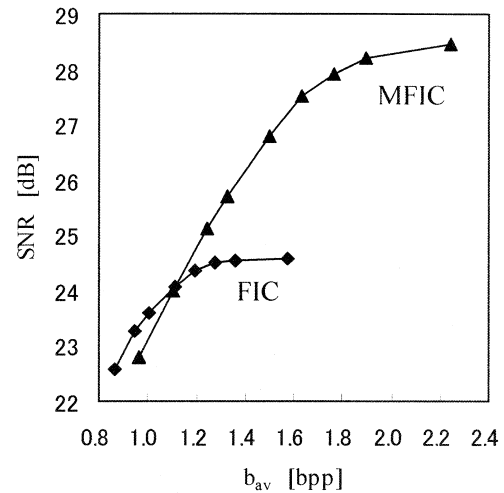


Fig. 4 Coding property

#### 2.4 シミュレーション結果

ITEの標準画像、“Church”(256×256[pels], 8[bits])を用いて、符号化シミュレーションを行った。符号化パラメータは、Table 1のように設定した。また、レンジブロックサイズは4×4、ドメインブロックサイズは8×8とし、シングルスケーリングブロックとマルチスケーリングブロックとの閾値は、シェードブロックとの閾値決定後に、シングルスケーリングブロックとマルチスケーリングブロックの比率が1:2となるように設定した。スケーリング係数は、シングルスケーリングブロック、マルチスケーリングブロック共に、{±0.2, ..., ±0.9}と、4 bitsで量子化した。マルチスケーリングフラクタル符号化(MFIC)と、従来のシングルスケーリングのみのフラクタル符号化(FIC)との、シミュレーション結果をFig. 4に示す。ビットレートは、シェードブロックとエッジブロックの割合を変化させることにより調節した。フラクタル符号化では、ビットレートの増加に伴い、どうしてもSNRは飽和してしまうが、MFICではFICと比較してその飽和値が約4[dB]向上した。これは、コントラストの高いブロックに対して、マルチスケーリングが有効に機能している為である。従って、FICよりも近似精度の高い再生画像を得ることが可能となる。しかしながら、低ビットレートにおいては、マルチスケーリングブロックの必要とする符号量が多いため、相対的にシェードブ

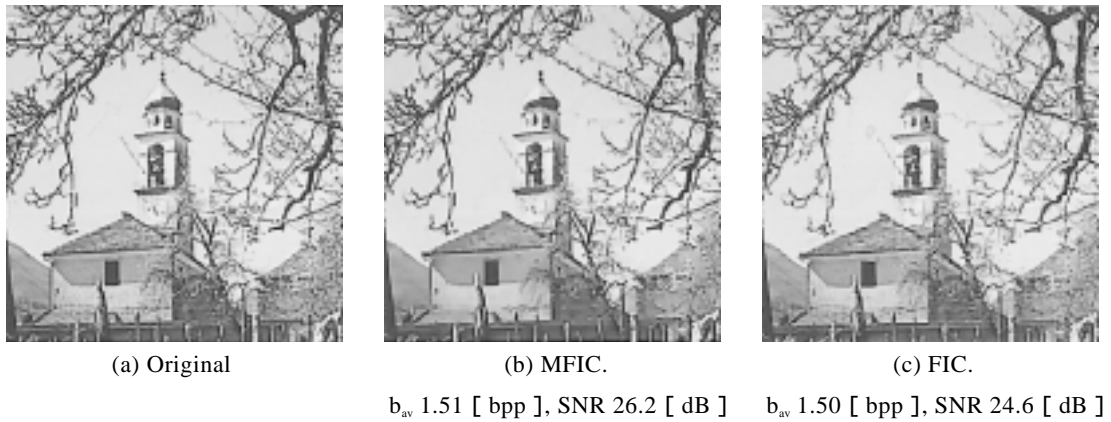


Fig. 5 Decoded images

ロックの割合が高くなり、MFICのSNRが低下している。

再生画像をFig. 5に示す。FICでは、木の枝等、細かな部分等において、十分な縮小変換が発見できない為にエッジが不鮮明となるなどの劣化が目立つが、MFICでは、マルチスケーリングにより物体と背景の差異が明確となり、細部まで鮮明に再現されている。しかしながら、MFICでは、マルチスケーリングブロックの必要とするビットが多いため、同ビットレートにおいてはFICよりもシェードブロックの割合が多くなってしまふ。そのため、マルチスケーリングブロックと判定された箇所はエッジ部が急峻となっているが、シェードブロックとエッジブロックを判別する閾値が高く設定され、細い木の枝等、比較的分散の低い個所がシェードブロックとなり、特にシェードブロックが集まっている部分において、タイル状の歪みとなって現れる。

### 3. カラー画像への応用

#### 3.1 基本構成

カラー画像に対するフラクタル符号化は、文献7,11,12等により行われているが、符号化性能の改善のために、第2節で述べたマルチスケーリングフラクタル符号化をカラー画像に対して適用し、その有効性を検討する。カラー画像に対する符号化では、空間的な冗長性と共に、3プレーン間の冗長性を考慮することが重要となってくる。一般に、輝度成分の劣化は歪として知覚されやすく、色差成分の劣化は比較的知覚されにくい。そこで、ここでは、RGBプレーンを、輝度成分Yと、色差成分IQとに変換し、符号化を行う。

RGBとYIQとの変換式は、

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.271 & -0.648 \\ 1.000 & -1.110 & -1.700 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \quad (12)$$

で与えられる。ここで用いるYIQは、NTSC方式で用いられている方法である。Y成分とIQ成分との視知覚特性を考慮して、符号化する際には、輝度成分Yに対して重点的にビットを割り当て、色差成分IQに対しては、ビットの割り当てを減少させることにより、効率的な符号化の実現を図る。

カラー画像に対するマルチスケーリングフラクタル符号化の構成図をFig. 6に示す。まず、RGBをYIQに変換した後、Y成分に対して第2節で述べたマルチスケーリングフラクタル符号化を施す。そして、IQ成分に

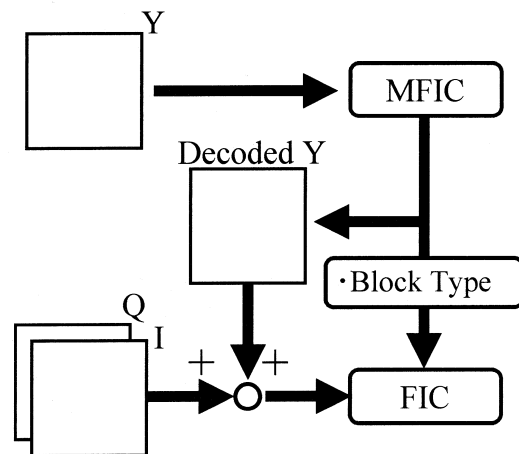


Fig. 6 Coding for color image

対しては、情報量が少ないため、文献7に従い、Y成分とIQ成分との相関を利用して、復号されたY成分をオフセットとして加え、符号化を行う。また、ブロックタイプはY成分と同一のものとする。但し、IQ成分については、ビットを削減するため、マルチスケーリングは用いず、エッジブロックは全てシングルスケーリングのみとする。また、ドメインブロックの位置と等長変換は、IQ成分で共有化する。

### 3.2 シミュレーション

ITEの標準画像、“Church”(256×256 [pels], 24 [bits]) に対して、符号化シミュレーションを行った。レンジブロックサイズは4×4、ドメインブロックサイズは8×8とした。また、カラー画像に対しては、マルチスケーリングブロックによるビットの負担がモノクロ画像よりも大きくなるため、マルチスケーリングブロックとシングルスケーリングブロックの個数比を、1:3と設定した。その他の符号化特性は、Table

Table. 2 Setting of coding

	$R_{ave}$	$k, l$	$\alpha$	$\varepsilon$
Y	6 bits	10 bits	3 bits	3 bits
I	5 bits	6 bits	3 bits	3 bits
Q	5 bits		3 bits	

2のように設定した。スケーリング係数は、シングルスケーリングブロック、マルチスケーリングブロック共に、 $\{\pm 0.2, \pm 0.4, \pm 0.6, \pm 0.8\}$ と3bitsで量子化した。

Table 3とFig. 7に符号化特性を示す。マルチスケーリングを用いることにより、2.53 [bpp]においてRGB各プレーンでSNRが、それぞれ1.0 [dB], 1.6 [dB], 0.4 [dB]向上した。また、SNRの飽和値も、RGB各プレーンでそれぞれ、1.6 [dB], 2.4 [dB], 0.6 [dB]向上した。Fig. 8に再生画像を示す。マルチスケーリングを用いることによって、木の枝等の複雑な構造を持つ

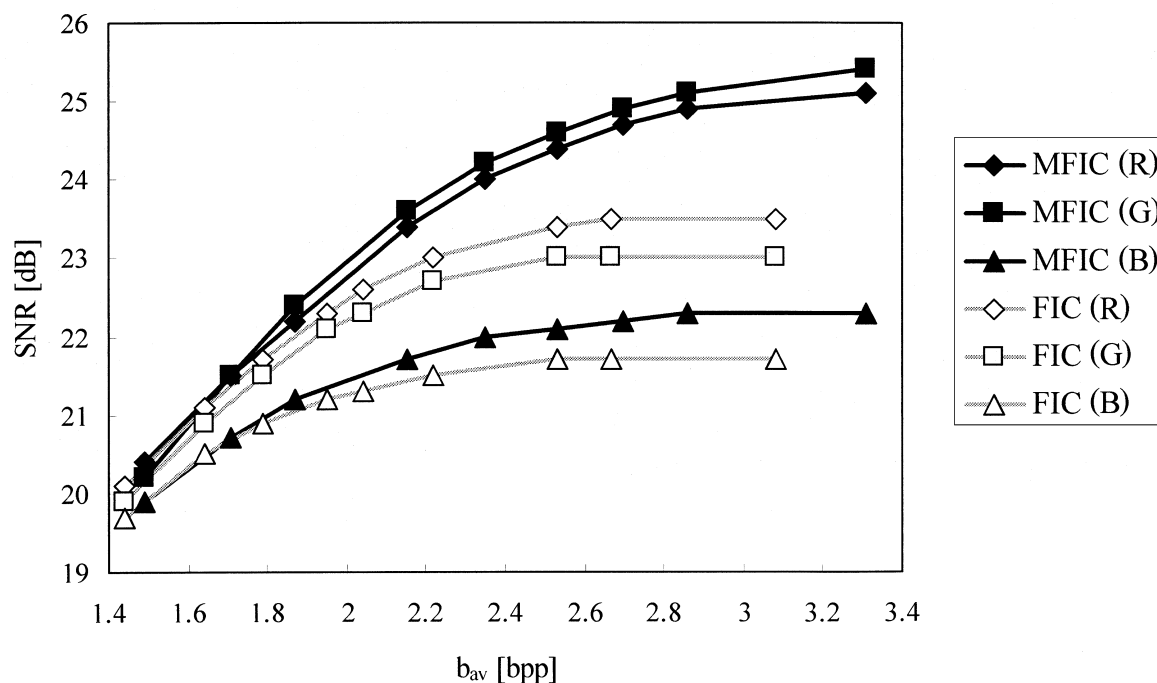


Fig. 7 Coding property for color images

Table. 3 Result of simulation

	SNR(R)[dB]	SNR (G)[dB]	SNR (B)[dB]	$b_{av}$ [bpp]
MFIC	24.4	24.6	22.1	2.53
FIC	23.5	23.2	21.7	2.53

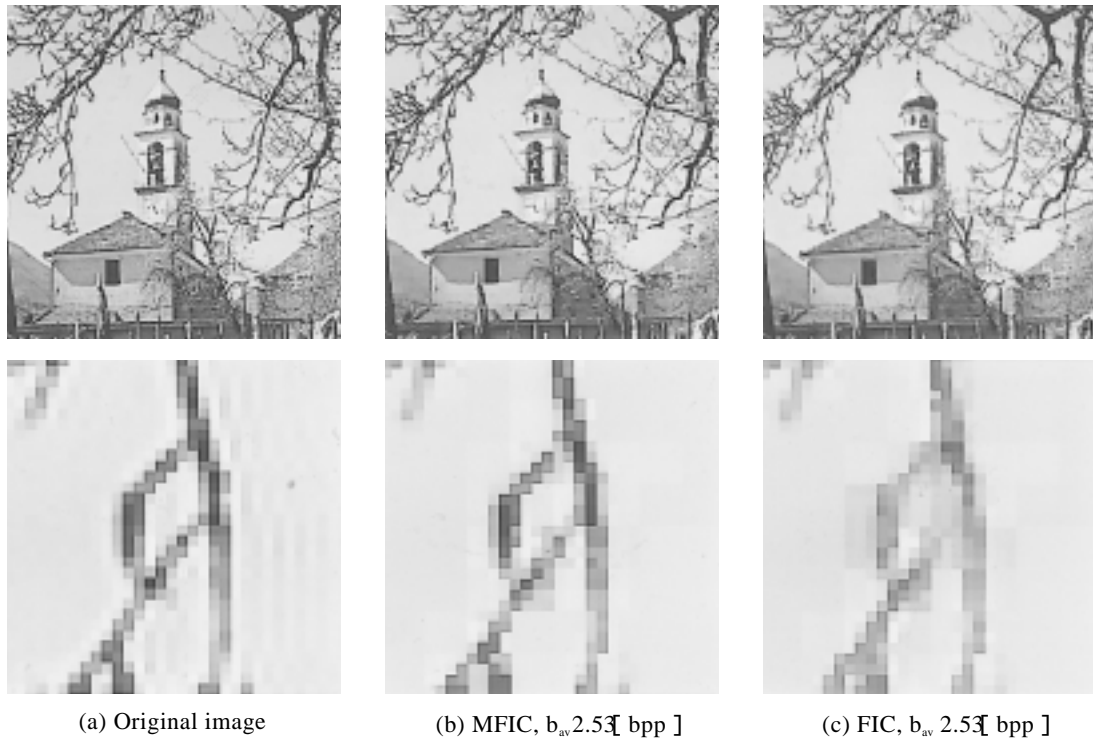


Fig. 8 Decoded images

た部位では、鮮明にエッジが再現され、良好な再生画像が得られた。しかしながら、全体的にシェードブロックの割合が増えているために、従来手法ではエッジブロックと判別されていた個所がシェードブロックと判別されてしまい、タイル状の歪となって現れている。その為、低ビットレートにおいては、従来手法よりもSNRが低下している。

#### 4. むすび

本論文では、複雑な構造を持った画像に対して、レンジブロックを領域分割して個別にスケーリングパラメータを与えることにより、より高い近似度を持った縮小変換を得ることを目的とした、マルチスケーリング手法の、カラー画像に対する適用について検討を行った。従来手法では十分な近似を与えることができなかった画像に対しても高いSNRが得られ、SNRの飽和値が、RGB各プレーンで、0.6~2.4 [ dB ] 向上した。また、高周波成分を多く含むブロックに対しても、エッジが鮮明に再現された。しかしながら、マルチスケーリングブロックが多くのビットを必要とするため、従来手法よりもシェードブロックの割合が高くなり、ブロック状の歪が目立つようになる。そのため、低ビットレートにおいては、SNRの低下がみられた。

低ビットレートでの符号化性能の改善の為、ブロックタイプの適切な判別方法を検討する必要がある。今回はマルチスケーリングブロックとシングルスケーリングブロックの比率を固定したが、マルチスケーリングブロックの割合を減らせばSNRは高まらず、割合を増やせば低ビットレートが困難となる。そこで、ビットレートに合わせてブロックタイプの割合を適応的に切り替える必要がある。また、本論文では分割数を2と設定したが、より近似度の高い縮小変換を探索するために、更に多くのレベルに分割する、多値のマルチスケーリングについての検討も考えられる。レベル数が増加すると、更に多くのビットが必要となるため、輝度レベルだけでなく、空間に対するマルチスケーリングも導入しなくてはならないと考えられる。

#### 参考文献

- 1) M.F.Barnsley, "Fractals Everywhere", Academic Press, New York, 1988.
- 2) M.F.Barnsley, "Fractal Image Compression", AK Peters, Wellesley, 1993.
- 3) A.E.Jacquin, "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations", IEEE Trans. Image Process., vol.1, No.1, pp.18-30, Jan., 1992.
- 4) A.E.Jacquin, "Fractal Image Coding: A Review", Proc. IEEE, vol.81, No.10, Oct. 1993.
- 5) 井田孝, 駄竹健志, "反復変換符号化による画像圧縮", 第5回回路とシステム軽井沢ワークショップ, 21-22, pp.137-142, 1992.



- 6) 中野勝彦, 中川匡弘, "ガウシアンピラミッドを利用した階層的IFS画像符号化", 信学論(A), vol.J78-A, no.7, pp.856-863, July 1995.
- 7) 中野勝彦, 中川匡弘, "カラー画像の階層的IFS符号化", 信学論(A), vol.J81-A, no.11, pp.1592-1599, Nov. 1998.
- 8) 古沢竜志, 中川匡弘, "輝度に対するマルチスケーリングを用いたフラクタル画像符号化", 信学論(A), vol.J84-A, No.3, pp.321-328, March 2001.
- 9) E.J.Delp and O.R.Mitchell, "Image Compression Using Block Truncation Coding", IEEE Trans. Commun., vol.27, No.9, pp.1335-1342, 1979.
- 10) Y.Linde, A.Buzo and R.M.Gray, "An Algorithm for Vector Quantizer Design", IEEE Trans. Communi., COM-28, No.1, pp84-95, Jan. 1980.
- 11) 大石義明, 山本強, 青木由直, "フラクタルブロック符号化を用いた画像圧縮法とその評価", テレビ技報, vol.16, pp.1-6, 1992.
- 12) 佐藤淳, 中川匡弘, "IFSに基づいた画像圧縮に関する一考察", 信学技報, NLP93-63, 16-05, pp.444-447, Nov. 1993.