グラフィクスとCAD 91−10
（1998. 8. 28）

3次元フラクタル圧縮とその適用

小向 順　　土井 章男
岩手大学工学部情報工学科

概要

フラクタル圧縮は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り，その原理はフラクタル画像を生成するIFS(Iterated Function System)理論に基づいており，任意の画像に対して，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．しかしながら，現状では2次元画像を対象にしたフラクタル圧縮が中心であり，3次元画像への適用はあまり行われていないと思われる．本論文では，MRI，CT，VHD等の3次元画像を圧縮，復元できる3Dフラクタルイメージ圧縮法，および並列計算環境における高速化手法について述べる．

3D Fractal Compression and Its Application

Jun Komukai and Akio Doi
Department of Computer Science, Faculty of Engineering, Iwate University

Abstract

Fractal Compression is an image coding method that is different from a method using statistical property of frequency domain, such as JPEG using DCT, and the idea is based on IFS (Iterated Function Systems) theory that generate fractal image, and its advantage is fast decompress using contractive transformation to for an arbitary image.
However, research for fractal compression focus on 2D image, currently, we suppose that it is not applied to 3D volume data fully.
In this paper, we describe 3D fractal image compression and, its parallelization that can compress and decompress 3D image, such as MRI, CT and VHD.

1 はじめに

フラクタル圧縮法はJacquinによって提案された画像圧縮の手法のひとつである[1],[2],[3]. 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であり, その原理はフラクタル画像を生成するIFS(Iterated Function System)理論に基づいている. ここでIFS理論は, 任意の画像に対して, 一定の縮小写像を反復的に施すことで唯一の画像に収束することを示しており, この理論を用いたフラクタル符号化は, 画像の復元が高速に行えるという利点を持っている. しかし, フラクタル圧縮法は符号化時間が膨大になるという欠点を持っており, この問題を解決するための高速化手法が研究されている[4],[5].

しかしながら, これらのフラクタル圧縮に関する研究は, その欠点である符号化時間の短縮という点に焦点があてられており, フラクタル圧縮のカラー画像への適用, 及び3次元画像への適用は未だ十分でないと思われる.

本論文では, フラクタル圧縮の3次元画像への適用について考察し, MRI(Magnetic Resonance Imaging), CT(Computerized Tomography), VHD(Visible Human Data set)等の3次元画像を圧縮, 復元できる3次元フラクタル圧縮法, および並列計算環境における高速化について述べる.

2 2D フラクタル圧縮のアルゴリズム

2.1 符号化

符号化は最初に, 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(レンジブロックと呼ぶ). 次に, 同様に原画像中から$2R \times 2R$画素のブロックd(ドメインブロックと呼ぶ)を取り出す. 各レンジブロックrに対して図1の様なアルゴリズムを用い, 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に対して, 縮小変換S, 対象変換I, 輝度スケーリングα, 輝度シフトoを施し, レンジブロックrとの距離を求める. 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(図2)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択される.

変換されたドメインブロックの内, 全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.

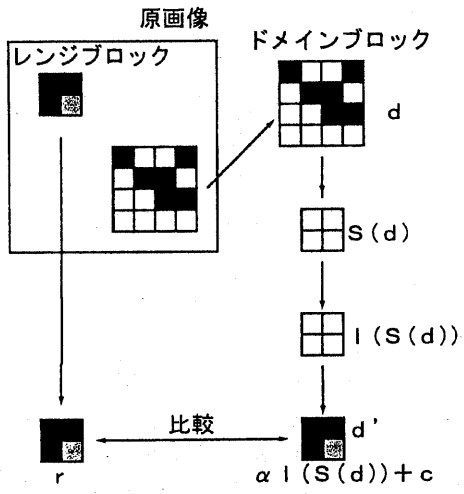


図1 符号化原理(2D)

2.2 復号化

復号化側ではレンジブロックの平均値, 輝度スケーリング値, 最適ドメインブロックの座標, 回転情報といったパラメータを受信する. これに従い, 復号化側では次の様な過程で復号処理を行う.

受信したレンジブロックの平均値を用いて解像度の低い初期画像を生成する. この初期画像を基にして繰り返し以下の処理を行うことで解像度を上げていく.

1. 最適ドメインブロックの座標からドメインブロックを切り出す.
2. 切り出したドメインブロックに縮小変換 S を施す.
3. 回転情報に従ってブロックに回転変換を施す.
4. 輝度スケーリング α, 輝度シフト o, 縮小及び回転変換されたドメインブロックの画素値 d_{ij} と平均値 $m(d)$ を用いて新しいレンジブロックの画素値 $\hat{r}_{ij}$ を計算し, 書き換える.

$$(\hat{r}_{ij}) = (d_{ij} - m(d)) * \alpha + o \qquad (1)$$

この処理を画像の変化が収束するか, 十分な解像度の画質が得られるまで行う (図 2).

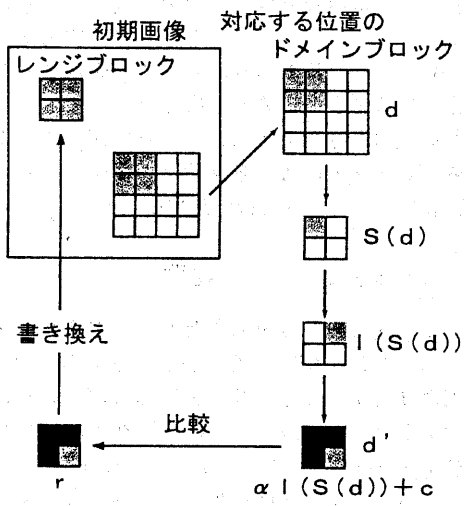


図 2　復号化原理

3　3D フラクタル圧縮のアルゴリズム

3.1　符号化

3 次元画像に対する圧縮アルゴリズムは 3 次元画像を $R \times R \times R$ のブロック (レンジブロック) に分割する. 次に, 原画像中から $2R \times 2R \times 2R$ 画素のブロック d(ドメインブロックと呼ぶ) を取り出し, 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 を施して, 最も類似したドメインブロックを選択する. 情報圧縮は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て行う [6](図 3).

3.2　複号化

3 次元画像の復号化手順は 2 次元画像の場合と同様である. 復号化側ではレンジブロックの平均値, 輝度スケーリング値, 最適ドメインブロックの座標, 回転情報といったパラメータを受信する. これに従い, 復号化側では次の様な過程で復号処理を行う.

受信したレンジブロックの平均値を用いて解像度の低い初期画像を生成する. この初期画像を基にして繰り返し以下の処理を行うことで解像度を上げていく.

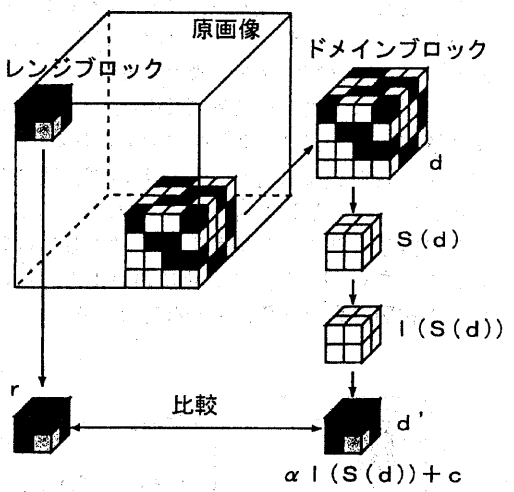


図 3　符号化原理 (3D)

3.3　カラー画像への適用

本節では, フラクタル圧縮を一般のカラー画像に適用する. ここでは, カラー画像を R, G, B に分割し, それぞれについて別々にドメインブロックの位置, 輝度スケーリング, 輝度シフト, 回転情報といった情報を格納し, 復号化側では, 各 R, G, B 毎に復号化を行い, 最終的に R, G, B 画像を統合し, 復元画像を生成している. 濃淡画像の場合と比較すると, R, G, B それぞれについて符号化手順を踏むため, 格納情報量は 3 倍になる.

4 アルゴリズムの高速化

4.1 レンジブロック分類による高速化

フラクタル圧縮においての欠点は，その膨大な符号化時間であると述べたが，その最たるは，最適ドメインブロックの探索である．本節では，符号化時間短縮のためのドメインブロック探索アルゴリズムについて述べる．

このアルゴリズム [5] は，レンジブロックと中心座標が等しいドメインブロックと，その近傍を優先的に探索する方法で，レンジブロック境界付近の画素と周辺の画素では相関が高いことを考慮している．

フラットレンジブロック (画素値の標準偏差が大) とノンフラットレンジブロック (画素値の標準偏差が小) を定義する．フラットレンジと分類された場合，レンジブロックの平均輝度値で近似される．ノンフラットの場合は，ドメインプールからドメインブロックを探索する．

レンジブロックと中心座標が等しいブロックが最適ドメインブロックとして用いられる場合，アフィン変換による対称変換を行わずに最適ドメインブロックとして用いることができる (図 4)．

この方法を 3 次元に適用した例は十分に検討されていないが，3 次元画像においても近傍の画素の相関が高い場合，十分有効である．

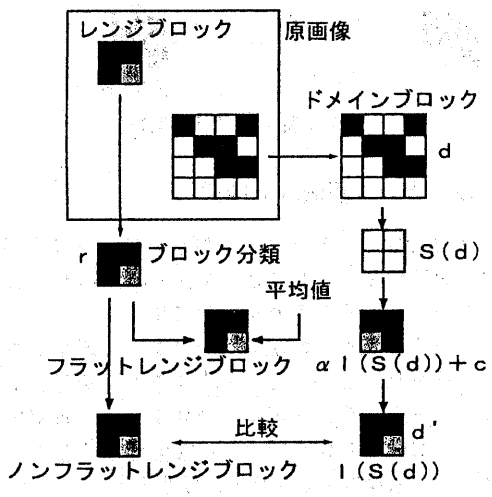


図 4　レンジブロック分類による符号化

4.2 並列計算における高速化

前節で，レンジブロックの分類，ドメインブロックの探索範囲の制限によるアルゴリズムの高速化を述べた．本節では，目的レンジブロックに対する最適ドメインブロックの探索が，それぞれで独立であり，他のレンジブロックに依存しないことに注目し，各レンジブロックに対応する最適ドメインブロック探索を，並列計算機によって並列に行う手法を提案する．

図 5 に示した様な，レンジブロックを分類して平均値で近似，またはレンジブロックに対応する最適ドメインブロックを探索するまでの過程を一つのプロセスとし，個々のプロセスを並列に計算し，プロセスが終了し次第，次のプロセスに移る．N個のレンジブロック，4 つの CPU でこのアルゴリズムを実行すると，始めは各 CPU に 1-4 番目のプロセスが割り当てられるが，各プロセスの計算時間には差があるため，計算終了したものから順に 5,6$\cdots$ とプロセスが割り当てられる．前プロセスの出力が次のプロセスの入力になる様な計算では，出力待ち時間のため，並列化の効率が最大にはならないが，ここでは各プロセスの計算は各々独立しているため，並列化アルゴリズムを最大限に生かせる．

5 シミュレーション結果

本手法の有効性を示すために，252 × 188 画素の 2D 風景画像，72 × 112 × 16 画素の肘の断面画像を用いて，圧縮，復元およびその計算時間を計測した．用いた並列計算機は，SGI 社の Origin2000(CPU $R10000 \times 16, Memory 3GB$) である．レンジブロック r として 4 × 4 × 4 画素，ドメインブロック d として 8×8×8 画素を用いた．また，輝度スケーリングは $\frac{dev(r)}{dev(d)}$，ここで $dev(r)$ と $dev(d)$ はそれぞれレンジブロックとドメインブロックの画素値の標準偏差の値を示している．輝度シフトにはレンジブロックの平均値 $m(r)$ を用いる．また，復号化処理は画像の変化の収束を計算せずに，繰り返し処理を一律 10 回ほど行っている．

レンジブロックを，フラットレンジブロックとノンフラットレンジブロックに分類し，符号化時間の高速化を計っている．また，ドメイン

ブロックの探索範囲は原画像全体からではなく，近傍画素は相関が高いことを利用してレンジブロック近傍，2次元画像では16×16ピクセルの範囲で，3次元画像では16 × 16 × 16ピクセルの範囲で行った．2次元画像および3次元画像の圧縮画像を図5に，ブロック分類による計算結果を表1に，並列化の計算結果を図6に示す．

	風景画像	肘断面画像
全探索	43173 scc	86400 scc
(1)	72 scc	4440 scc
(1),(2)	67 scc	3242 scc

(1)レンジ近傍探索
(2)ブロック分類(フラット&ノンフラット)

表1　ドメイン探索法の違いによる符号化時間の比較

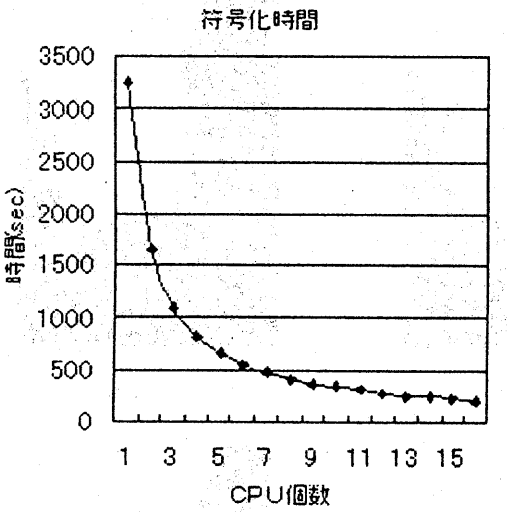


図6　符号化時間の比較

6　まとめと今後の発展

本論文では，フラクタル圧縮のカラー画像と3次元画像への拡張，ドメインブロック探索の高速化，さらに並列化による高速化について検討した．

検討の結果，フラクタル圧縮のカラー画像と3次元画像への適用が確認できた．2次元から3次元，モノクロからカラーとなることで大幅に符号化時間が増大したが，ドメインブロックの探索範囲をレンジブロック近傍とすることと，並列計算機によって並列化を行うことで，符号化の高速化が行えた．また，各レンジブロックに対するドメインブロック探索の計算が各々独立であるため，オーバーヘッドを生じることが無く，計算を行うCPUの数に比例して符号化時間が短縮できるため，フラクタル圧縮は並列計算に向いているといえる．

しかしながら，エッジ部分での復元が十分でない所があるため，復元画像の精度を向上させるアルゴリズムの実装，また，可変符号長などを用いた圧縮率の向上等が今後の課題である．

謝辞

本研究は、(財) 放送文化基金により補助を受けた．ここの感謝の意を示します．

参考文献

[1] 徳永隆治, "フラクタル", ジャストシステム社, 1993.

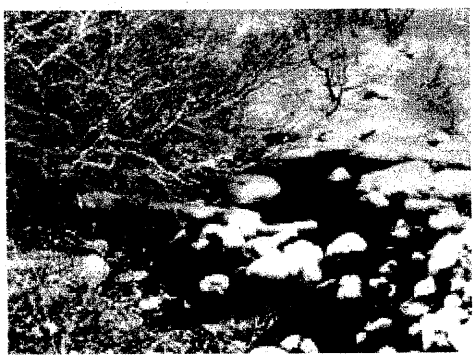
[2] 斉藤隆弘, "フラクタル符号化", テレビジョン学会誌, Vol. 50, No. 8, pp. 1054-1062, 1996.

[3] 渕上隆博, 小松隆, 斉藤隆弘, "ひずみレート・コスト関数によるＩＦＳ画像符号化法の適応化", テレビジョン学会誌, Vol. 50, No. 8, pp. 1132-1141, 1996.

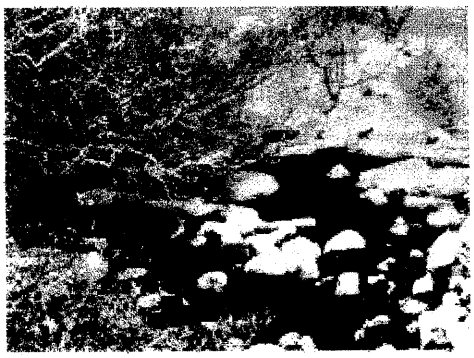
[4] 富樫慎司, 池原雅章, 野原隆, "フラクタル符号化の高速化", 情報処理学会論文誌, Vol. 38, No. 1, pp. 64-72, 1997.

[5] 黒田英夫, Dan C.popescu, 今村 幸祐, Hong Yan, "フラクタル画像符号化における高速ブロックマッチング法", 情報処理学会論文誌, Vol. 38, No. 8, pp. 1543-1553, 1997.

[6] Wayne O.Cochran, C.Hart, Patrick J.Flynn, "Fractal Volume Compression", IEEE Transactions on Visualization and Computer Graphics, Vol. 2, No. 4, pp. 313-322, 1996.



原画像(2D)

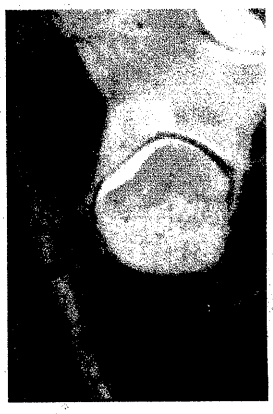


圧縮画像(2D)

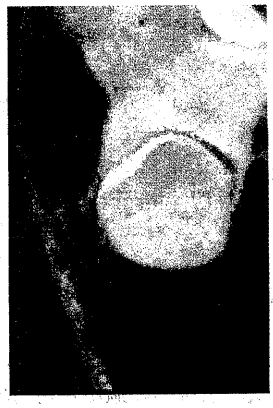
原画像(3D)

z＝0

z＝10



z＝15



圧縮画像(3D)

図5　原画像と圧縮画像の比較