

相似領域を可変としたフラクタル符号化を用いた画像領域分割

武内 朗†(学生員) 上田 哲史†(正員)  
寺田 賢治†(正員)

Image Segmentation Using Fractal Coding with Resizable Domain Blocks

Akira TAKEUCHI†, Student Member, Tetsushi UETA†,  
and Kenji TERADA†, Regular Members

† 徳島大学工学部, 徳島市  
Faculty of Engineering, Tokushima University, Tokushima-shi, 770-8506 Japan

あらまし フラクタル符号化で用いたアフィン変換について, その逆写像を反復計算する力学系におけるアトラクタの引力圏を求めることにより, 画像の領域分割が可能である. 本論文は, 相似領域可変型フラクタル符号化法と, 領域分割手法のアルゴリズムの改良を行い, 実験によってそれらの有効性を示している.

キーワード フラクタル符号化, アフィン変換, トッピング領域, 画像領域分割

1. まえがき

近年, 新しい画像符号化としてフラクタル符号化 [1] ~ [3] が, その高圧縮可能性のため注目され, 広く研究されている. フラクタル符号化の圧縮データは, 画像の部分自己相似性と, いくつかの関連した定理に基づいて決定されたアフィン変換のパラメータ集合で構成されている. このパラメータを直接処理することで, 画像を復号する必要なく, 原画像に関するいくつかの画像処理を簡単に, かつ高速に行うことが可能である. その一例として, フラクタル符号化の圧縮データを用いた画像の領域分割手法が, 井田ら [4] によって提案されている. これは, 圧縮データから得られる変換の繰返しによって現れる点集合を, クラスタリングによっていくつかのグループに分割し, そのベースン(引力圏)を求めるという手法である. しかしこの手法では, 分割領域の境界にあいまいさが残存すること, 粒状ノイズ的な誤検出が生じること, 視覚的に判別可能な領域の誤統合, 誤分割があること, などの問題点が見られた.

そこで本論文では, 上記問題点の解消と, 精度向上を実現するために, 相似領域可変型フラクタル符号化を用いた領域分割アルゴリズムを提案する.

2. フラクタル符号化

フラクタル符号化の手順は以下のとおりである. はじめに画像を重ねりのない小領域  $R_i$  に分割し, それ

より大きく相似な領域  $D_i$  を同じ画像内から探索する. 次に相似領域  $D_i$  で小領域  $R_i$  を置き換えるような縮小アフィン変換  $f_i$  を求める(図 1).  $f_i$  は式 (1) で表される.

$$f_i : \mathbf{R}^3 \rightarrow \mathbf{R}^3$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & p_i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} s_i \\ t_i \\ q_i \end{pmatrix} \quad (1)$$

ここで,  $x, y$  は  $R_i$  内のある点の座標,  $z$  はその点の濃度値を表し,  $a_i, b_i, c_i, d_i, s_i, t_i, p_i, q_i$  はアフィン変換のパラメータである. このとき, すべての小領域  $R_i$  をそれぞれに対する相似領域  $D_i$  で置き換えた画像が, 原画像とほぼ等しいものであるならば, 任意の画像に対して, アフィン変換集合  $F = \{f_i | i = 0, 1, 2, \dots\}$  を繰返し施すことで近似画像を復号できることがコラージュ定理 [3] によって保証されている. したがって, アフィン変換の集合  $F$  のパラメータのみを圧縮データとしてもてばよい [1].

3. 従来の領域分割法とその問題点

3.1 小領域から相似領域への変換  $G$  の計算

フラクタル符号化された画像の領域分割法として井田の手法がある. この手法では, 画像再生時とは逆に小領域内の点を対応する相似領域内へ写像するような変換集合  $G = \{g_i | i = 1, 2, \dots\}$  を考える. ただし, ここでは濃度方向は無視し, 点の位置の写像のみを考えるものとする. 変換  $g_i$  は圧縮データより式 (2) を用いて容易に求めることができる.

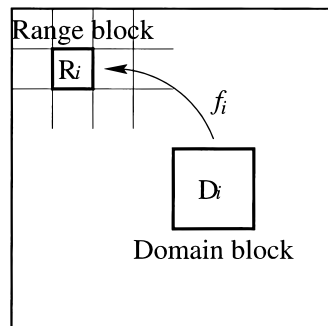


図 1 相似領域から小領域への変換  
Fig. 1 Transformation to range from domain block.

$$g_i : R^2 \rightarrow R^2$$

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} a'_i & b'_i \\ c'_i & d'_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} s'_i \\ t'_i \end{pmatrix} \quad (2)$$

ここで、 $x, y$  は点の座標、 $i$  は小領域の番号を意味し、 $a'_i, b'_i, c'_i, d'_i, s'_i, t'_i$  は定数である。ただし、 $a'_i = d_i / (a_i d_i - b_i c_i)$ ,  $b'_i = -b_i / (a_i d_i - b_i c_i)$ ,  $c'_i = -c_i / (a_i d_i - b_i c_i)$ ,  $d'_i = a_i / (a_i d_i - b_i c_i)$ ,  $s'_i = -s_i$ ,  $t'_i = -t_i$  である。この変換集合  $G$  を、画像内の各点に施す。まず、圧縮データからその点が属する小領域に対応する変換  $g_i$  を求め、その点に変換  $g_i$  を施す。求まった写像点は、その点が属する小領域の変換  $g_i$  によって写像され、以後はこれを繰り返す。このとき、図 2 のように小領域内の点に対応する相似領域内の点に写像される。例えば、ある小領域  $R_0$  内の点  $k_0, l_0, m_0, n_0$  は、対応する相似領域  $D_0$  内の点  $k_1, l_1, m_1, n_1$  に写像される。この写像点は小領域  $R_1$  内の点であるため、今度は相似領域  $D_1$  内の点  $k_2, l_2, m_2, n_2$  に写像される。このようにして次々に写像を繰り返していくと、小領域と相似領域の重なり合いにより、複雑な軌跡となる。

変換  $G$  の回数が増えるにつれて点はいくつかの集合に分かれ、ある程度の回数が行われると、それ以上集合の外形はほとんど変化しなくなる。井田らは、これは画像内にいくつかのトラッピング領域が形成されているためであると指摘している。

### 3.2 トラッピング領域と領域分割

このトラッピング領域が形成される理由は次のよう

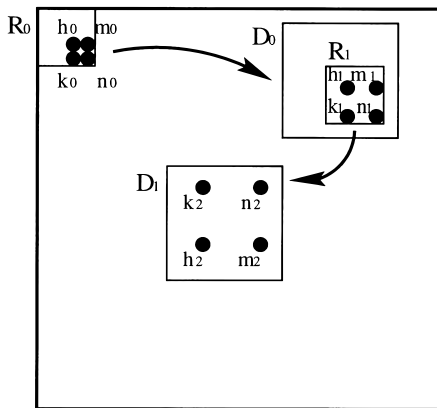


図2 変換  $G$   
Fig.2 Mapping  $G$ .

に考えることができる。図 3 のように、ある領域  $S$  の周りで相似領域を選ぶ場合、式 (1) の  $s_i, t_i$  を小さく、また  $p_i$  を 1 に近い値に、 $q_i$  を 0 に近い値に制限すると、相似領域は以下の 3 パターンになる可能性が高い [4]。

- (1) 小領域が  $S$  の内部：相似領域も  $S$  の内部。
- (2) 小領域が  $S$  の外部：相似領域も  $S$  の外部。
- (3) 小領域が  $S$  のエッジを含む：相似領域も  $S$  のエッジを含む。

エッジを含む小領域に対しては、それ自身を含むように相似領域を決定する可能性が高く、エッジにそって環状にエッジを含む相似領域が選ばれる。そのため、エッジ内部と外部の相似領域は重なりをもたないように選ばれる。そして、エッジ周辺の点はエッジの内部、若しくは外部へ押し出されるように写像されることから、トラッピング領域が形成されにくくなる。それに対して、エッジ内部、外部ではそれぞれの相似領域の重なりが著しく多くなるため、トラッピング領域が形成されやすくなると考えられる。

各分割領域にはそれぞれトラッピング領域が形成され、異なる領域からスタートした初期点は異なるトラッピング領域に至ることから、写像により各初期点がどのトラッピング領域に入ったかでラベル付けすることで、領域の分割が可能となる。しかし、実際には、トラッピング領域の位置を正確に特定することは非常に困難である。井田らは、一定回数変換  $G$  を計算した後の点を、クラスタリングによりある程度の要素を含む集合に分割し、それぞれのクラスタをトラッピング領域とみなす手法を提案している [4]。ところがこの手法では、分割された領域の境界のあいまいさや、

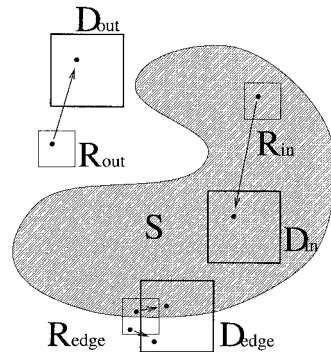


図3 領域  $S$  周辺の小領域と相似領域  
Fig.3 Range and domain blocks around segment  $S$ .

イズ状の誤検出，誤った領域の統合，分割などの問題点が見られた．これらの原因としては，フラクタル符号化の際に適切に相似領域が求まっていないこと，本来同じトラッピング領域内の点でも，点の距離が離れているため，異なるトラッピング領域内の点として分割されてしまっていることなどが考えられる．

#### 4. 相似領域可変型フラクタル符号化

前述のとおり，エッジ周辺の点は変換  $G$  によりエッジから離れるように写像されるため，分割すべき領域内部のすべての点は，そのエッジを超えて写像されることはない．しかし，エッジを含む小領域で適切でない相似領域が求まっていると，エッジをまたいで写像する点が現れ，結果として領域の誤分割，統合の可能性が高くなる．

実際にはエッジを含んだ小領域では，適切な相似領域を探索するのは非常に困難であり，特に相似領域が小領域より大きくなるほどそれが顕著になる．例えば，図 4 の小領域  $R$  に対して，その相似領域が  $D$  と求まったとする．このとき小領域内の点を変換  $G$  により写像すると，図 5 となる．ここで，小領域  $R$  において，エッジ左上の点を  $a$ ，右下の点を  $f$  で表している．小領域ではエッジ右下にあった一部の点が，変換  $G$  により図 5 の相似領域  $D$  内の点  $a \sim f$  はエッジ左上に写像されてしまっている．このままでは誤って領域分割

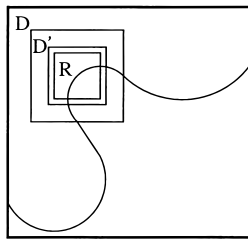


図 4 小領域  $R$  に対する相似領域  $D, D'$   
Fig. 4 Domain block  $D (D')$  for range block  $R$ .

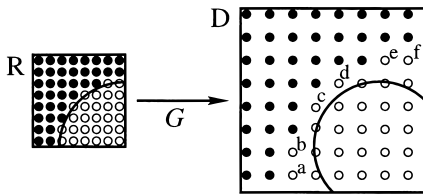


図 5 相似領域が大きい場合  
Fig. 5 Case of large domain block.

される．

そこで本論文では，相似領域の大きさを小領域に近くすることで，より適切な相似領域を探索する手法を提案する．この手法を用いることで，小領域  $R$  に対して，その相似領域は  $D'$  と求まり，図 6 のように，図 5 に見られた変換  $G$  による誤った領域への写像が減少すると考えられる．ただし，すべての小領域に対する相似領域を小さくしてしまうと，小領域，相似領域の重なり合いが減るため，幾何学的に本質的な情報が減り，領域分割はもとより，画像の再生でさえなくなる．このことから，エッジを含んだ小領域のみ相似領域を小さくするものとする．エッジを含むかどうかの判定は，小領域の濃度分散値を用いて行う．分散値があるしきい値より大きければ，エッジを含むとみなし相似領域を小さくする．本論文では，エッジを含む小領域では相似領域をその 1.25 倍にし，それ以外は 2 倍とする．本論文ではこの提案手法を，相似領域可変型フラクタル符号化と呼ぶ．

更に領域分割の改良方法を考える．まず，クラスタリングの前に写像点の周期点分類を行う．トラッピング領域内の点は外部へ写像されないことから，周期点はすべて同じトラッピング領域内の点であるはずである．したがって，クラスタリングを行わなくても写像点の分類ができる．次に，それぞれの分割すべき領域の境界にはエッジが存在する可能性が高く，この場合エッジを含まない小領域が隣り合っている場合は，その小領域が同じ領域に含まれると考えられるため，それら小領域内の点は変換  $G$  を計算せず，隣り合ったすべての小領域内の点を同じ点（固定点）に写像させる処理を行う．本手法では，隣り合った小領域の中で最上段左端の小領域の左上の点に写像するものとする．これらの処理を施した後，クラスタリングにより写像点をいくつかの集合に分割し，それぞれをトラッピング領域とみなし初期点をラベル付けする．この結果，領域の誤分割，誤統合が減少する [5] ~ [7] ．

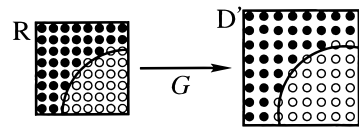


図 6 相似領域が小さい場合  
Fig. 6 Case of small domain block.

## 5. 適用例

図 8 と図 9 に、それぞれ図 7 を従来のフラクタル符号化を用いて圧縮した場合と、相似領域可変型フラクタル符号化を用いて圧縮した場合の圧縮データより再生した画像を示す。ここで、図 7 は  $256 \times 256$  pixel の白黒濃淡画像であり、表 1 のパラメータにより符号化を行った。相似領域は小領域の左上の画素を中心に表 1 に示す  $s_i, t_i$  の範囲を左上から右下へ探索し、小領域と対応する画素ごとの 2 乗誤差の総和が最も小さいものを相似領域とした。また、相似領域可変型フラクタル符号化において小領域がエッジを含むかどうかのしきい値は 500 とし、小領域内の濃度分散値がしきい値を超える場合はエッジを含むとした。従来法の結果と比べて、提案手法の結果では、人物の目や口といった部分にひずみが強く見られる。これは、エッジを含む小領域が隣接し、それぞれに求まる 1.25 倍の相似領域同士の重なり合いが少ないため、うまく再生できないと考えられる。

井田らの手法により図 7 に関して領域分割を行った結果を図 10 に示す。提案手法により領域分割を行った結果を図 11 に示す。両手法ともクラスタリングには K-mean 法 [8], [9] を用い、クラスタ数を 50 とした。図 12 (a), (b) は図 10, 図 11 それぞれの左上、帽子と背景の境界付近を拡大した図である。

提案手法を用いた手法では、井田らの手法に見られた分割された領域の境界のあいまいさや、ノイズ状の誤検出、誤った領域の統合、分割などの問題点が減少している。特に、原画像の帽子と背景の境界などにおいて、領域と領域の境界がより明確になっており、相似領域可変型フラクタル符号化による相似領域探索が、領域分割に有効であることがいえる。

更に、図 13 を原画像として井田らの手法と提案手法により領域分割した結果をそれぞれ図 14, 図 15 に示す。図 7 の場合と同様に井田らの手法に見られた問題点が提案手法により減少している。

まだいくつかの部分でノイズなどの問題点が残って

表 1 符号化パラメータ

Table 1 Parameters of fractal coding.

- ・ 小領域サイズ :  $8 \times 8$
- ・ 相似領域の回転 : 0, 90, 180, 270
- ・ 相似領域の反転 : 左右反転
- ・  $s_i, t_i$  : -16, -15, ..., 15, 16
- ・  $p_i$  : 0.9
- ・  $q_i$  : -15, -14, ..., 14, 15



図 7 原画像 1  
Fig. 7 Original image 1.



図 8 従来のフラクタル符号化の再生画像  
Fig. 8 Decoding image of original fractal coding.



図 9 相似領域可変型フラクタル符号化の再生画像  
Fig. 9 Decoding image of proposal fractal coding.

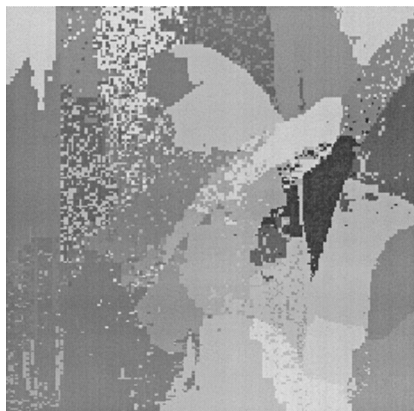


図 10 井田らの手法の適用例  
Fig. 10 Segmentation result of Ida's algorithm.



図 13 原画像 2  
Fig. 13 Original image 2.

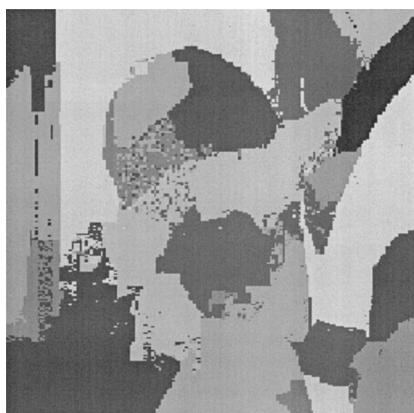


図 11 提案手法の適用例  
Fig. 11 Segmentation result of proposal algorithm.

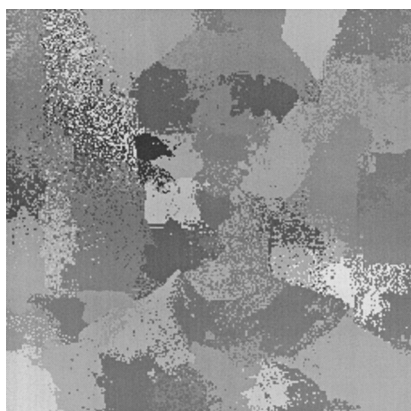


図 14 井田らの手法の適用例  
Fig. 14 Segmentation result of Ida's algorithm.

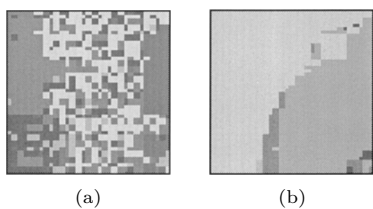


図 12 適用例の拡大図  
Fig. 12 Enlarged figures of segmentation results.

いるが、これは相似領域の大きさ決定のしきい値や、クラスタリングに用いるパラメータの細かい設定により、十分な品質の領域分割画像が得られると考えられる。

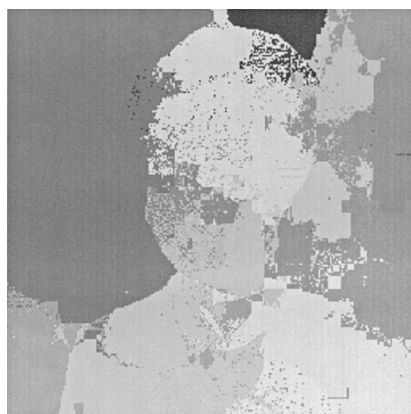


図 15 提案手法の適用例  
Fig. 15 Segmentation result of proposal algorithm.

## 6. むすび

フラクタル符号化では、各小領域に対してより適切な相似領域を求めることが、画像再生と、領域分割において重要な問題である。本論文では、井田らの提案した領域分割の精度向上のため、より適切な相似領域を求める手法を提案した。具体的にはエッジ部の小領域に対して、大きさのあまり変わらない相似領域を選ぶ相似領域可変型フラクタル符号化の提案を行い、その圧縮データを用いた領域分割手法に適用した。更に井田らの領域分割アルゴリズムを改良することで、より精度の高い領域分割画像が得られることを示した。

今後の課題としては、各種パラメータの設定や、新たな改良手法の考案により、更に精度の高い領域分割画像の取得することである。

謝辞 有益な御助言を頂いた徳島大学川上博先生に深謝致します。

### 文 献

- [1] M.F. Barnsley and L.P. Hurd, *Fractal Image Compression*, AK PETERS, Ltd., 1993.
- [2] A.E. Jacquin, "A novel fractal block-coding technique for digital images," *Proc. ICASSP-90*, pp.2225-2228, 1990.
- [3] 徳永隆治, *フラクタル—美しさを超えて, ジャストシステム*, 1993.
- [4] T. Ida and Y. Sambonsugi, "Image segmentation using fractal coding," *IEEE Trans. Circuits & Syst. for Video Technology*, vol.5, no.6, pp.567-570, Dec. 1995.
- [5] 武内 朗, 上田哲史, 寺田賢治, "フラクタル力学系を用いた画像の領域分割," *信学技報*, NLP98, May 1998.
- [6] 武内 朗, 上田哲史, 寺田賢治, "フラクタル力学系の不安定化を用いた画像の領域分割," *平 10 四国連大*, no.1-6, p.6, Oct. 1998.
- [7] 武内 朗, 上田哲史, 寺田賢治, "フラクタル力学系に基づいた画像の領域分割," *信学技報*, PRMU98, Jan. 1999.
- [8] 田村秀行, *コンピュータ画像処理入門*, pp.153-161, 総研出版, 1985.
- [9] 尾崎 弘, 谷口慶治, 小川秀夫, *画像処理—その基礎から応用まで*, 共立出版, 1983.
- (平成 11 年 9 月 24 日受付, 12 年 1 月 7 日再受付)