

フラクタル符号に基づく圧縮領域における類似画像検索手法

横山 貴紀[†] 菅原 研[†] 渡辺 俊典[†]

本稿では、フラクタル画像圧縮によって得られる圧縮符号を対象とした、圧縮領域における類似画像検索手法を提案する。フラクタル画像圧縮は同一画像中の自己相似領域を探索し、対応領域を縮小写像として表現し、圧縮符号として記録する。この写像情報は、画像の空間領域における相似領域の関係に基づいた構造表現と見なすことができ、圧縮符号である写像集合に対して直接類似度を定義することができる。この類似度を用いた検索により、圧縮符号を復号することなく、圧縮領域において画像の検索および分類などが可能となる。デジタルカメラで撮影した画像を元に圧縮符号データベースを作成し、検索実験を行うことで提案手法の特性と有効性を示す。

PIFS Codes Based Image Retrieval Scheme in the Compressed Domain

TAKANORI YOKOYAMA,[†] KEN SUGAWARA[†]
and TOSHINORI WATANABE[†]

We propose a new compressed-domain image retrieval scheme based on the similarity of PIFS codes. In PIFS encoding, the compressed code contains a set of contractive mapping information between similar regions in an image. This mapping information can be represented as vectors carrying the structural feature of the image. We introduce a new similarity measure between two vector sets, and exploit them in image categorization and retrieval. In this paper, we explain this scheme and demonstrate the effectiveness of the proposed method by experiments.

1. はじめに

音声、画像、動画などの様々なメディアを利用した情報システムの発展につれて、これらの情報に対する検索システムの実現が強く望まれるようになってきている。

メディア情報をテキスト情報の検索と同様にキーワードで検索することは一般に難しいとされている。メディア情報に適切なキーワードをつけることが難しく、すべての情報にキーワードを付与すること自体が困難であるのがその理由である¹⁾。

そのためメディアの持つ内容そのものによる検索手法が研究されている。これは一般に、内容に基づく検索と呼ばれ、メディアが持つ特徴をその内容と見なすことで検索を実現するものである²⁾。

メディア情報の1つである画像に関する検索技術の研究が、近年非常にさかに行われるようになってきた。画像の検索では、画像から抽出される様々な特徴

量を、単独あるいは複数組み合わせるなどして検索を行う。用いられる特徴としては、色、テクスチャ、対象オブジェクトの形状や位置情報、構図などがあげられる³⁾。これらの特徴をベクトル化したものを用いて検索を行うが、特徴量ベクトルが完全に一致するものだけを探すのではなく、距離や類似度の指標を基に類似したものを候補としてあげる、いわゆる類似検索が通常行われる。

これまで研究されてきた画像検索システムの多くでは、検索対象を原画像そのものとしている。そのため、原画像からの特徴抽出、特徴量ベクトルの生成、索引情報のデータベースを原画像データベースとは別に作成するなどの処理が必要となる。また、新たなデータを追加する場合には、そのつど検索に必要な処理コストが発生する。

ところで、ネットワーク上の通信や、データの蓄積などにおいて、画像情報は何らかの圧縮処理が施された圧縮データ(以下、圧縮符号と呼ぶ)の形で使用される。そこで、圧縮符号を直接活用することができれば、この環境が最大限活用できる。また、通常の画像検索システムに必要な原画像側での特徴抽出処理を省

[†] 電気通信大学大学院情報システム学研究科
Graduate School of Information Systems, University of
Electro-Communications

くことができる。

原画像を対象とした手法を画素領域 (Pixel Domain) 法と呼ぶのに対し、圧縮符号を対象とした手法は圧縮領域 (Compressed Domain) 法と呼ばれる⁴⁾。圧縮領域における検索手法では、それぞれの圧縮手法に応じた特徴量の定義や類似性判定などが必要となる。これまでに、離散コサイン変換や Wavelet 変換、ベクトル量子化などの圧縮手法について、その圧縮符号の係数などを基にした数多くの検索手法が提案されている⁴⁾。

本稿では、任意の解像度で復元画像を生成でき、符号からの画像の復号に必要な計算量も少なく、画像検索システムを構築するうえで有利な特性を持つフラクタル画像圧縮⁵⁾を取り上げ、この圧縮符号を対象とした、圧縮領域における類似画像検索の手法を提案する。

フラクタル画像圧縮を用いた画像検索手法には以下のようなものがあげられる。Zhang らの Joint Fractal Coding⁶⁾は、圧縮時の圧縮特性を用いるもので、検索時には質問画像と対象となるすべての画像を用いて圧縮を行い、圧縮時の振舞いに基づいて検索を行う。Tan らの Fractal Neighbor Distance⁷⁾は、圧縮符号の復号特性に着目しており、検索対象となる画像に質問画像の圧縮符号を順次適用し、適用前の画像との距離を求めて検索を行う。Marie-Julie らは圧縮に用いる分割領域のサイズを固定とし、複数の領域サイズにおいて圧縮を実行する多重圧縮と多重解像度表現により検索を行う手法⁸⁾を提案している。上田らは、Ida らの領域分割手法⁹⁾に基づき、相似領域の検索範囲と縮小性を限定して得られる圧縮符号に対して、テンプレートマッチングを行う手法¹⁰⁾を提案している。

以上のように、提案されているフラクタル画像圧縮を用いた既存の検索手法では、検索に圧縮あるいは復号処理などを必要とし、圧縮領域での検索の利点を必ずしも活かしていないと見ることができる。

そこで本稿では、圧縮符号から構成されたデータベースに対して、圧縮領域において直接検索を行う方式を提案する。この手法は、一般的に用いられているフラクタル画像圧縮手法から得られる圧縮符号のみを検索対象とし、検索用途向けに調整した特殊な圧縮手法を必要とせず、いっさいの圧縮や復号処理を必要しない。本稿は、フラクタル符号が記述する相似関係に基づいた、画像空間上の構造特性に着目した新たな検索手法の枠組みを提案するものである。

以下、2章ではフラクタル画像圧縮について説明し、3章では圧縮符号が持つ縮小写像のベクトル集合としての表現を述べ、このベクトル集合を基にした符号間

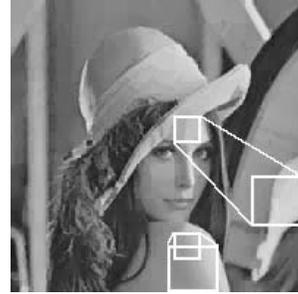


図 1 画像中の相似領域の例

Fig. 1 An example of similar regions in the image.

の類似度を定義する。4章では提案する類似度を用いた検索実験により、その特性と有効性を検証し、最後にまとめを行う。

2. フラクタル画像圧縮

フラクタル画像圧縮は、図 1 で示すような画像中の自己相似領域に基づく圧縮手法である。圧縮の原理は Barnsley の反復関数系 (IFS: Iterated Function System) によるフラクタル画の作成方法に基づいている⁵⁾。フラクタル画像圧縮の特色を以下にまとめる。任意の解像度による復元： 圧縮符号から任意の解像度で画像を復元することができる。これにより、あらかじめすべてのサイズの画像を用意する必要がなく、要求に応じてサムネイルのような小さな画像から、大きな画像を生成することができる。さらに、画像本来の解像度を超えた場合でも、良好な表現特性 (フラクタルズーム¹¹⁾) を持っている。

フラクタル画像圧縮の 2 次利用： フラクタル画像圧縮に基づいた領域分割手法^{9),12)} や、高度な輪郭線抽出法¹³⁾などが提案され、実際の製品¹⁴⁾として実用化されている。また、電子透かしへの適用¹⁵⁾が提案されるなど、フラクタル画像圧縮から派生する、圧縮用途以外のアプリケーションが充実しつつある。

画像空間の相似領域に基づいた構造表現： フラクタル画像圧縮は、画像空間における相似領域に基づいた関係を圧縮符号として記録する。この相似領域の関係は、画像の変動に対し不変性を持つと考えられ⁷⁾、ロバスト性の高い類似検索を実現できる可能性がある。

以上から、フラクタル画像圧縮が画像データベースを構築するうえで有利な特性を備えていると考え、提案手法のベースとなる圧縮方式として取り上げる。

以下、フラクタル画像圧縮としては、Jacquin の

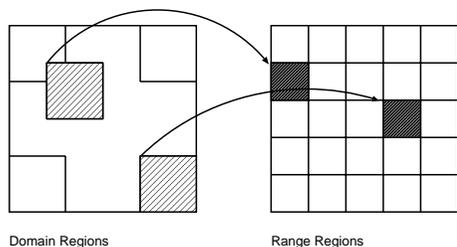


図 2 同一画像のレンジとドメインの領域割当て
Fig.2 Domain region and range region.

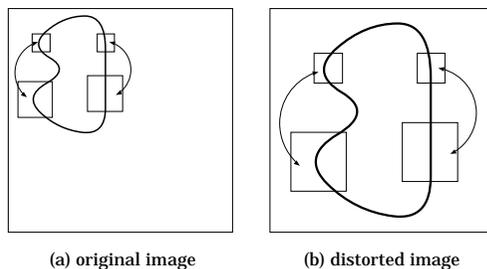


図 3 画像の変動下における相似領域関係の不変性
Fig.3 Invariance of the relation of similar regions under some distortions.

PIFS (Partitioned IFS) 符号化手法¹⁶⁾ に基づくものを想定する．この手法は、画像を分割することで Barnsley の手法を自動化したものであり、現在フラクタル画像圧縮手法の基礎となっている¹⁷⁾．なお、圧縮原理などの詳細については他の文献^{11),18)} に譲ることとし、本稿では説明を省略する．

2.1 フラクタル符号

フラクタル符号には領域間の相似関係が記録されている．図 2 のように、同一画像を「ドメイン」と呼ばれる領域と、それよりも小さい「レンジ」と呼ばれる領域に分割する．

あるレンジ領域 R_i の位置情報を x_{R_i}, y_{R_i} , その輝度値を z_{R_i} とする．この領域と相似関係にあるドメイン領域 D_i の位置情報を x_{D_i}, y_{D_i} とし、輝度値を z_{D_i} とする．このとき、縮小写像 w_i は式 (1) のアフィン変換で表現される．

$$\begin{bmatrix} x_{R_i} \\ y_{R_i} \\ z_{R_i} \end{bmatrix} = w_i \begin{bmatrix} x_{D_i} \\ y_{D_i} \\ z_{D_i} \end{bmatrix} \tag{1}$$

$$= \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & p_i \end{bmatrix} \begin{bmatrix} x_{D_i} \\ y_{D_i} \\ z_{D_i} \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix}$$

ここで a_i, b_i, c_i, d_i は空間の回転と縮小を表す係数、 p_i は輝度値の縮小係数である． e_i, f_i は空間のオフセット値、 o_i は輝度のオフセット値を表す．

グレースケール画像に適用するため、上式には輝度情報に関する縮小写像が含まれている．ドメイン領域の縮小写像で得られる輝度値と、対応するレンジ領域の輝度値との相関を最小 2 乗推定して得られる o_i, p_i が通常用いられる．

すべてのレンジ領域について相似なドメイン領域を決定し、その関係を縮小写像として記録したものがフラクタル符号となる．フラクタル符号は、写像 w_i を構成するために必要な係数の情報から成り立っている．

2.2 レンジ領域の分割

圧縮時に必要な画像のレンジ領域は、画像を固定サイズで一様に分割するのではなく、Quadtree 分割¹⁸⁾ のような、領域の変動に応じて領域のサイズを可変する分割手法が通常用いられる¹⁷⁾．

この分割手法を用いた圧縮では、最も大きいサイズのレンジ領域から相似領域の探索を始める．画像中のすべてのドメイン領域を縮小した領域の輝度値と、レンジ領域の実輝度値との誤差が閾値を超える場合には、さらにレンジ領域を 4 分割し、近似度の良いドメイン領域を探索する．これを許される分割サイズまで繰り返すことで、領域の変動に応じた領域分割が行われる．

このように可変サイズの領域分割手法を用いた場合、大きな固定サイズを用いた領域分割に比べ、復号時の歪みが少なくなる．また、レンジの領域数は縮小写像数と同じであり、領域数の減少は符号の圧縮率の向上につながる．

3. フラクタル符号の圧縮領域における類似検索手法

フラクタル符号から得られる、縮小写像の集合を基にした画像の検索手法を説明する．

フラクタル画像圧縮により記録された相似領域の関係は、画像の各種変動下で不変性を持つと考えられている⁷⁾．たとえば図 3 のように、原画像に含まれる図形が拡大し移動する変動が加えられた場合でも、四角形で示すような互いに相似な領域の関係は変化しない．提案手法は、この不変性を持つ相似領域の関係を、画像空間上の構造であると見なし、この構造に対し類似度を定義し、圧縮領域での類似画像検索を実現する．

手順としては、圧縮符号から得られる写像情報を画像空間上のベクトルとして抜き出し、このベクトル集合をレンジ領域のサイズごとに区分して部分集合化する．この部分集合に対し、画像空間上の構造を反映した類似度を定義する．検索はこの類似度により実行さ

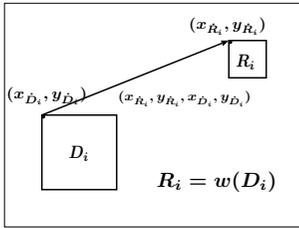


図4 $(x_{R_i}, y_{R_i}, x_{D_i}, y_{D_i})$ による画像の構造表現
Fig.4 Image structure representation by
 $(x_{R_i}, y_{R_i}, x_{D_i}, y_{D_i})$.

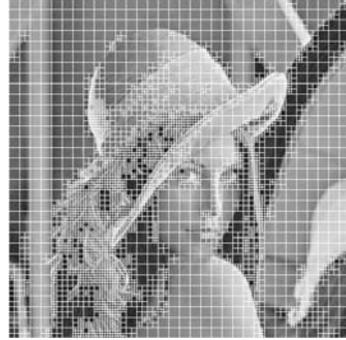


図5 Quadtree 分割によるレンジ領域の割当て
Fig.5 Range size regions by Quadtree partitioning.

れる．各処理の詳細について以下に述べる．

3.1 相似領域の写像ベクトル表現

本手法では輝度変動に関してロバストであるために、写像 w_i の式 (1) における輝度に関する係数 o_i, p_i を使用しない．また、アフィン変換による縮小と回転項に関する係数 a_i, b_i, c_i, d_i も画像内の部分領域間の構造関係とは独立であると考えて無視し、 e_i, f_i のみに着目する．

具体的には、領域 R_i, D_i のそれぞれの左上端点を $(x_{R_i}, y_{R_i}), (x_{D_i}, y_{D_i})$ と表したとき、写像 w_i が表す相似関係を、4次元ベクトル $(x_{R_i}, y_{R_i}, x_{D_i}, y_{D_i})$ として扱う．この4次元の写像情報は、図4のように画像平面上のレンジ領域 R_i と、相似なドメインの領域 D_i の関係を、ベクトルにより構造的に表現していると見なすことができる．

以下、これを「写像ベクトル」と呼び、画像の圧縮符号が持つ画像空間上の構造表現として扱う．これらの写像ベクトルは、圧縮符号から直接抽出することができ、本手法では圧縮符号と写像ベクトルの集合を等価なものとして扱う．

3.2 写像ベクトル集合のレンジサイズによる区分

圧縮手法でレンジ領域の割当てに、可変サイズの領域分割手法を用いた場合、レンジ領域のサイズ(以下、「レンジサイズ」と呼ぶ)は画像の局所領域における輝度変動により変化する．

テクスチャやエッジ部に相当する輝度変化の激しい個所には最も小さなレンジサイズの領域が割り当てられ、変動の緩やかな部分には大きなレンジサイズの領域が割り当てられる傾向にある．図5はQuadtree分割を用いたときの、圧縮時に割り当てられた領域を格子で表現したものであり、これらの傾向を確認できる．

このようにレンジサイズによって領域の特性が異なり、結果として生成される写像ベクトルの集合も異なってくる．図6は、標準画像 lenna の圧縮符号から得られるレンジサイズごとの写像ベクトルを、それぞれ図示したものである．小さなレンジサイズの集合では、

顔や髪、エッジ部などの変動の激しい領域どうしに相似関係が形成されていることが確認できる．また大きなレンジ領域の集合では、変動の少ない背景部分に相当する領域どうしに相似関係が形成されていることが確認できる．

このように、レンジサイズごとに生成される写像ベクトルがそれぞれ異なるため、本手法では写像ベクトル集合を、レンジサイズごとに区分して扱うこととする．

3.3 写像ベクトル集合間の類似度の定義

前節までで、フラクタル符号から画像の特徴を表す写像ベクトル集合を抽出し、レンジサイズによって集合を区分した．検索はこの区分されたベクトル集合に対する類似度を基に行われる．画像の圧縮符号から得られる写像ベクトル集合に基づいた類似度の算出方法の詳細を以下に述べる．

2枚の画像を I_A, I_B とする．画像 I_A の圧縮符号中の写像ベクトルを $a_i^r = (x_{R_i^r}, y_{R_i^r}, x_{D_i^r}, y_{D_i^r})$ とする．ここで r はレンジサイズを表し、 i はベクトルのシーケンス番号である． I_B の写像ベクトルについても同様に $b_j^r = (x_{R_j^r}, y_{R_j^r}, x_{D_j^r}, y_{D_j^r})$ と表す．

I_A が持つレンジサイズ r の写像ベクトル集合を $A_r = \{a_1^r, \dots, a_{m(r)}^r\}$ 、同様に I_B 側は $B_r = \{b_1^r, \dots, b_{n(r)}^r\}$ とする．このとき、それぞれの写像集合のベクトル数 $m(r) = |A_r|$ および $n(r) = |B_r|$ は異なっていてよい．ここで $|\cdot|$ は集合の要素数であり、写像集合の写像ベクトル数を表す．

以上から、 r_{\min}, r_{\max} をレンジサイズの最小と最大とすると、 I_A の写像ベクトル集合は $A = \{A_{r_{\min}}, \dots, A_{r_{\max}}\}$ となり、同じく I_B については $B = \{B_{r_{\min}}, \dots, B_{r_{\max}}\}$ と表される．

写像ベクトル集合 A, B 間の類似度、つまり画像の圧縮符号間の類似度を以下のように考える．まず、

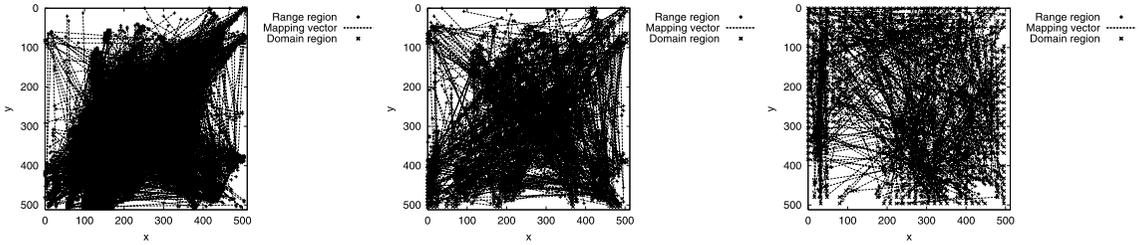


図 6 画像ベクトル集合のレンジサイズに基づく区分: (左) $r = 4$ (中) $r = 8$ (右) $r = 16$

Fig. 6 Subsets of Lenna's mapping vectors: (left) range size is 4, (center) range size is 8 and (right) range size is 16.

A_r の任意のベクトル a_i^r に対応する, B_r 中のベクトル b_j^r を決定する. 本手法では次の関係式を満たすものを対応ベクトルとした.

$$f(a_i^r) = \arg \min_{b_j^r \in B_r} \|a_i^r - b_j^r\| \quad (2)$$

ここで $\|\cdot\|$ はノルムを表す. A_r に対する B_r の対応ベクトルをすべて決定し, これを

$$f(A_r) = \bigcup_{i=1}^{m(r)} f(a_i^r) \quad (3)$$

と表す. このとき, 次式は A_r と B_r との間のベクトル対応の, 1 対 1 からのずれに着目した類似度となる.

$$s(A_r, B_r) = \frac{|f(A_r)|}{|A_r|} \quad (4)$$

ただし, この類似度は非対称であり $s(A_r, B_r) \neq s(B_r, A_r)$ となるときがある. これは集合の構成が大きく異なる場合と, お互いの画像ベクトル数が異なる場合に生じる. 特に集合の画像ベクトル数が著しく異なる場合, たとえば $|A_r| \ll |B_r|$ であるとき, つねに $s(A_r, B_r) = 1$ に, 逆は $s(B_r, A_r) = 0$ に近づく.

この非対称性への対策としては, 両者の平均をとる方法も考えられる¹⁹⁾. しかし, 単に平均をとるだけでは前述の $|A_r| \ll |B_r|$ の場合, つねに類似度は 0.5 となり, $s(A_r, B_r) = 0.5$ と $s(B_r, A_r) = 0.5$ の画像間と同様に類似していると誤った判断をされてしまう. そこで, 本手法では次式を用いることとする.

$$s^*(A_r, B_r) = \frac{|f(A_r)| + |f(B_r)|}{|A_r| + |B_r|} \quad (5)$$

s^* においては, $|A_r| \ll |B_r|$ の場合, $|f(A_r)| \leq |A_r|, |f(B_r)| \leq |A_r|$ のため, $s^*(A_r, B_r)$ は 0 に近づき, 平均化の矛盾を回避できる.

最後に, レンジサイズごとに定義した類似度 s^* を結合し, 画像の圧縮符号である画像ベクトル集合間の類似度 S を定義する.



図 7 画像データベースの画像例

Fig. 7 Some sample images in our image database.

$$S(A, B) = \sum_{r=r_{\min}}^{r_{\max}} \alpha_r s^*(A_r, B_r) \quad (6)$$

ここで α_r は, $\sum_{r=r_{\min}}^{r_{\max}} \alpha_r = 1, \alpha_r \geq 0$ を満たす重みである.

4. 実験

画像の圧縮符号データベースに対して実際に検索を行い, 提案手法の性質と有効性を検証する. 提案手法を C 言語によって実装し, コンパイラは gcc 2.95.3 を用いた. 実験は OS が Linux 2.4.20, CPU は Pentium 4 のクロック周波数 2.80 GHz, メモリは 1,000 Mbytes で構成される PC 上で行った.

4.1 画像データベース

圧縮符号の生成に用いた画像は, 図 7 に示すように人物や人形, 車などを対象としたものや, 山岳や道路, 室内などの風景, 紙面や看板などに記載された文字など様々である. 画像は, サイズが 320×240 pixel, 8 bit のグレースケールで, 合計 1,264 枚である.

この画像データベースには, 連続撮影などによって得られた図 8 (a), (b), (c) のような類似画像群が複数含まれている. このような類似画像群をあらかじめ



(a) landscape



(b) man



(c) doll

図 8 類似画像群：(a) landscape, (b) man, (c) doll

Fig. 8 Similar image groups: (a) landscape, (b) man, (c) doll.

目視により分類し、それぞれの画像群にラベルを付した。7種類、合計 91 枚を類似画像群と判断した。

これらの画像群からそれぞれ 1 枚を質問画像とし、同一画像群の画像を検索することで検索性能を確認する。検索性能については以下の指標を用いた。

$$\text{precision} = \frac{|\text{retrieved} \cap \text{relevant}|}{|\text{retrieved}|} \quad (7)$$

$$\text{recall} = \frac{|\text{retrieved} \cap \text{relevant}|}{|\text{relevant}|} \quad (8)$$

式中の retrieved は検索された画像の集合を表し、 relevant は検索結果として適切だと思われる画像の集合を表す。本実験においては質問画像が属する類似画像群を relevant とした。式 (7) は適合率であり、検索画像数に対する正しく検索された画像の割合を表す。式 (8) は再現率であり、正しく検索された画像数に対する、正しいとされる集合との割合を示す。

4.2 圧縮符号データベース

画像データベースから圧縮符号データベースを作成する。圧縮符号の生成には、Mario Polvere の Mars 1.0 を用いた。このプログラムでは、ドメインの領域サイズをレンジサイズの 2 倍とし、相似なドメイン領

表 1 圧縮符号データベースの写像数：レンジサイズ固定

Table 1 The number of the mapping vectors in the fractal code database: Fixed range size.

レンジサイズ	4	8	16
写像数	4800	1200	300

表 2 圧縮符号データベースの写像に関する統計量：レンジサイズ可変 ($r = 4, 8$)Table 2 Statistics of the mapping vectors in the fractal code database: Range size is variable ($r = 4, 8$).

レンジサイズ	all	4	8
平均値	2599	1865	733
標準偏差	828.0	1110.4	276.0
最大値	4800	4800	1200
最小値	1200	0	0

表 3 圧縮符号データベースの写像に関する統計量：レンジサイズ可変 ($r = 4, 8, 16$)Table 3 Statistics of the mapping vectors in the fractal database: Range size is variable ($r = 4, 8, 16$).

レンジサイズ	all	4	8	16
平均値	2194	1842	222	129
標準偏差	1036.6	1110.4	100.2	71.5
最大値	4800	4800	752	300
最小値	300	0	0	0

域の探索範囲は画像全体となっている。

本実験では合計 5 種類の圧縮符号データベースを用意した。レンジサイズを固定として作成した圧縮符号データベースを、それぞれレンジサイズ $r = 4, 8, 16$ の 3 種類用意した。可変サイズの領域分割を用いて生成する圧縮符号データベースは 2 種類用意した。1 つは $r = 4, 8$ のレンジサイズ、もう一方は $r = 4, 8, 16$ のレンジサイズを使用して作成したものである。

表 1 はレンジサイズ固定による圧縮符号データベース中の、1 つの圧縮符号に含まれている写像数を表している。また、表 2、表 3 はレンジサイズ可変の圧縮符号データベースについて、1 つの圧縮符号に含まれる平均の写像数と標準偏差、および最大値と最小値を、各レンジサイズとすべてのレンジサイズの合計 (all) について算出したものである。

次に、これらの圧縮符号データベースの圧縮特性を表 4 に示す。表中で「 $r = 4, r = 8, r = 16$ 」はレンジサイズ固定で作成した圧縮符号データベースを表し、用いたレンジサイズを示している。「#2」は 2 種類のレンジサイズ、「#3」は 3 種類のレンジサイズを用いた圧縮符号データベースであることを表している。

圧縮率 (Compression Ratio) は画像のファイルサイズを compressed 、圧縮符号のファイルサイズを uncompressed とした式 (9) を用いた。復号した画像の品質評価には、式 (10) の PSNR (Peak Signal-to-

表 4 圧縮符号データベースの圧縮特性

Table 4 Compression characteristics of the fractal code databases.

	$r = 4$	$r = 8$	$r = 16$	#2	#3
圧縮率 (%)	19.8	5.1	1.3	11.4	9.8
PSNR (dB)	31.1	26.9	23.84	29.9	30.4
圧縮時間 (sec.)	1.20	0.29	0.14	0.54	0.39
復号時間 (sec.)	0.07	0.04	0.03	0.05	0.05

Noise Ratio) を用いた. M は画像の画素数, \maxval は輝度値の最大値 (255), $\|\cdot\|_2$ は L_2 ノルムを表している. また圧縮と復号にかかった時間も示す. なお, これらの値はいずれも画像 1 枚あたりの平均値である.

$$\text{Compression Ratio} = \frac{\text{compressed}}{\text{uncompressed}} \quad (9)$$

$$\text{PSNR} = 20 \log_{10} \frac{\maxval}{\frac{1}{\sqrt{M}} \|I_A - I_B\|_2} \quad (10)$$

4.3 レンジサイズごとの類似度を用いた検索

以上の圧縮データベースに対し, それぞれのレンジサイズから得られる式 (5) の類似度 s^* を用いて検索を行い, 再現率と適合率を用いて検索性能を評価する. なお, 式 (2) には L_2 ノルムを用いている.

実験は類似画像群の画像 91 枚すべてを質問画像とし, 得られた検索結果について再現率と適合率の平均値を算出する. レンジサイズ固定の圧縮符号データベースに対する結果を図 9, 図 10 に示す. 図 9 は横軸を検索画像数, 縦軸を再現率とし, 図 10 は横軸を適合率, 縦軸を再現率としている. レンジサイズ可変の圧縮符号データベースについては, 図 11, 図 12 に結果を示す. 表中の r は, 検索時に使用した類似度 s^* が対象とするレンジサイズを表している.

これらの結果から, レンジサイズ固定の圧縮符号データベースに対する検索性能はあまり良くないと見ることができる. レンジサイズ可変の場合, 2 種類のレンジサイズを用いた場合圧縮符号データベースでは, $r = 4, 8$ それぞれの類似度 s^* による検索性能がほぼ同程度であることが確認できる. また, 3 種類のレンジサイズから構成された圧縮符号データベースでは, $r = 4$ の結果が良好であることが分かる.

表 5 に類似画像群の質問画像 1 枚あたりにかかった平均の検索時間を示す. 表中の「#1」は, レンジサイズ固定の圧縮符号データベースであることを示している. 検索プログラムは, 写像ベクトル間の距離行列を用いて式 (2) の対応を決定している. そのため, 類似度 s^* の算出には, 距離行列の生成にかかる $O(m(r) \cdot n(r))$ の計算量が必要となる. 表からも写像数の影響を読み取ることができる.

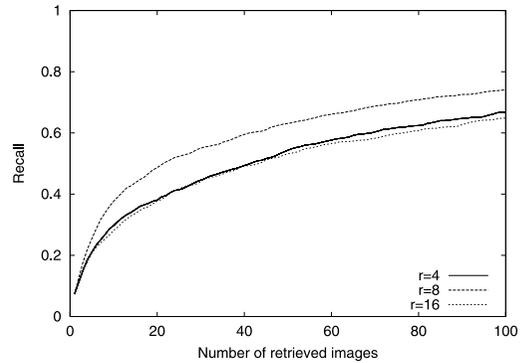


図 9 再現率: レンジサイズ固定

Fig. 9 Recall ratio vs. number of retrieved images: Range size is fixed.

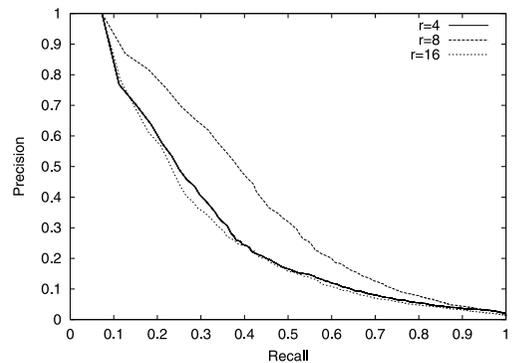


図 10 適合率と再現率: レンジサイズ固定

Fig. 10 Precision ratio vs. recall ratio: Range size is fixed.

表 5 類似度 s^* による平均検索時間 (秒)Table 5 Average retrieval times using the similarity s^* .

	$r = 4$	$r = 8$	$r = 16$
#1	2644.4	169.7	15.4
#2	117.4	105.5	-
#3	97.3	11.6	8.2

以上の検索性能と検索時間から明らかのように, レンジサイズ固定で生成した圧縮符号データベースへの提案手法の適用は好ましくないことが分かる.

4.4 類似度の結合による検索

4.3 節の結果から, 2 種類のレンジサイズを用いたレンジサイズ可変の圧縮符号データベースでは, それぞれのレンジサイズの類似度 s^* による検索性能が, ほぼ同程度の精度であると見なせた. この圧縮符号データベースを対象とし, 式 (6) の重みを $\alpha_4 = 0.5$, $\alpha_8 = 0.5$ として類似度 s^* を結合した類似度 S を用いて検索を行う. 質問画像 1 枚あたりに必要な検索時間は平均 221.1 秒であった.

表 6 類似度 S を用いた検索結果 ($\alpha_4 = 0.5, \alpha_8 = 0.5$)
 Table 6 Retrieval results by the combined similarity S ($\alpha_4 = 0.5, \alpha_8 = 0.5$).

質問画像	検索順位						
	2	3	4	5	6	7	8
landscape							
doll							
man							

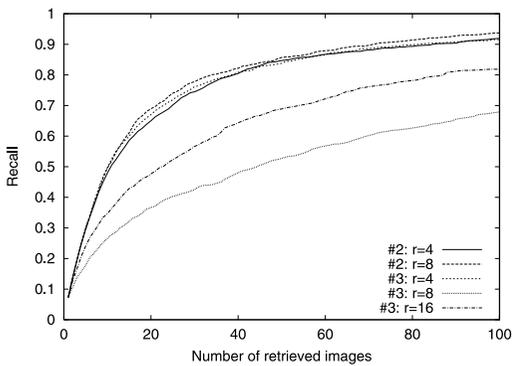


図 11 再現率：レンジサイズ可変

Fig. 11 Recall ratio vs. number of retrieved images: Range size is variable.

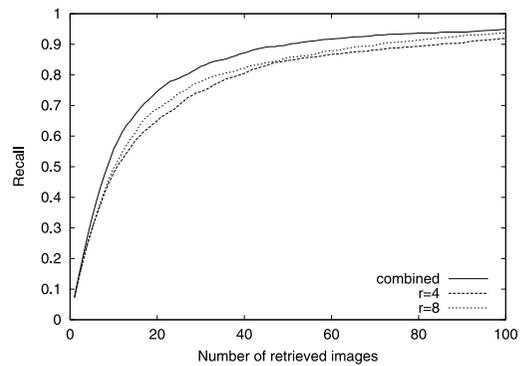


図 13 類似度 S の検索性能：再現率

Fig. 13 Performance of the combined similarity S ($\alpha_4 = 0.5, \alpha_8 = 0.5$): Recall ratio vs. number of retrieved images.

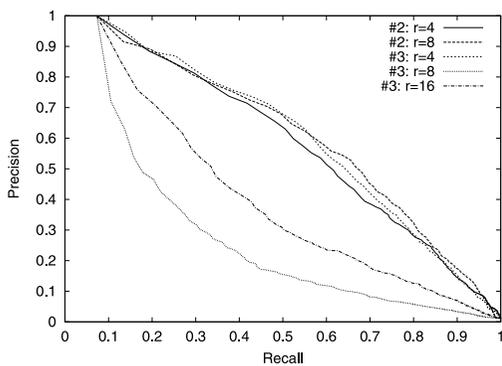


図 12 適合率と再現率：レンジサイズ可変

Fig. 12 Precision ratio vs. recall ratio: Range size is variable.

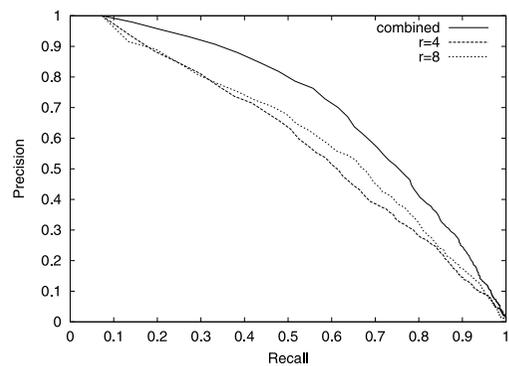


図 14 類似度 S の検索性能：適合率と再現率

Fig. 14 Performance of the combined similarity S ($\alpha_4 = 0.5, \alpha_8 = 0.5$): Precision ratio vs. recall ratio.

表 6 は、実際に質問画像を与えて検索を行った結果である。一番左の列に質問画像を、右の列には 2 位から 6 位までに得られた検索結果の画像を示す。この表

から良好な検索結果が得られていることが確認できる。図 13 は検索画像数に対する再現率を、図 14 は適合率と再現率を示す。両方のグラフには類似度 S を

表 7 各種変動画像
Table 7 Image variations.

質問画像	変動画像		
平行移動			
		...	
0 pixel	-48 pixel		+48 pixel
拡大			
		...	
150 %	100 %		200 %
回転			
		...	
0 度	-30 度		+30 度

用いた結果を combined と表し、それぞれのレンジサイズの類似度 s^* を独立して用いた検索結果と合わせてプロットする。これらの結果から、類似度 s^* の結合である類似度 S の導入により、検索性能が改善することが分かる。

4.5 画像の変動にともなう類似度の変化

ここでは、画像の変動にともなう類似度の変化を見る。画像データベースから任意に選び出した 1 枚について、各種変動を加えた画像を生成する。表 7 に示すような平行移動、拡大、回転の 3 種類の変動を作用させた。変動を受けた画像群から圧縮符号を生成し、類似度により検索を行う。前節までの結果を受け、ここでは 2 種のレンジサイズを用いた圧縮符号を生成し、 $\alpha_4 = 0.5, \alpha_8 = 0.5$ の類似度 S を用いる。

平行移動させた画像の類似度を図 15 に示す。オリジナル画像のままでは平行移動の画像が生成できないため、130%拡大した画像から 320×240 pixel のサイズで画像を切り出すこととした。この拡大により 96 pixel の水平方向の移動が可能となり、1 pixel ずつ移動させた合計 97 枚の画像を生成した。拡大した画像の中心で切り出したものを 0 pixel とし、これを質問画像として検索を行った。グラフ中の other image は、拡大した画像群以外の画像から得られた、最も高い類似度を表している。

この結果から、質問画像の近傍において高い類似度を示し、離れるに従って類似度が下がることが確認できる。類似度がとところどころ変動するのは、領域分割の影響を受けているものと考えられる。この画像ではほぼ ± 40 pixel までが other image の類似度を超え、この範囲において高い類似度を維持していると見るこ

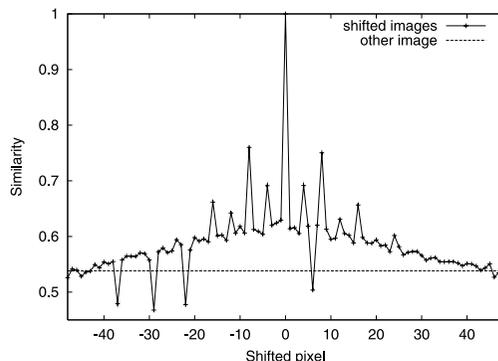


図 15 平行移動に対する類似度
Fig. 15 Similarity for shifted images.

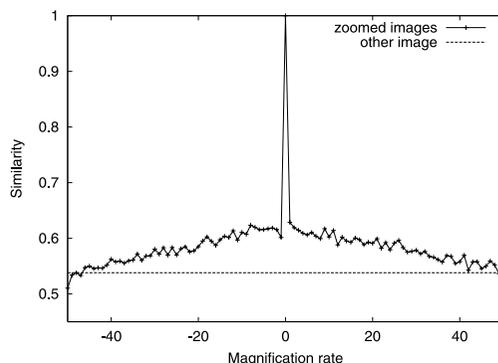


図 16 拡大に対する類似度
Fig. 16 Similarity for zoomed images.

とができる。

次に拡大画像についての結果を図 16 に示す。ここではオリジナルの画像を拡大率 100%から 200%まで、1%ずつ増加させた 101 枚の画像を生成した。質問画像には 150%拡大した画像を用いた。拡大率の変化とともに類似度が変動していることを確認できる。

最後に画像の回転について図 17 に結果を示す。画像をそのまま回転すると画像の端が切れる不具合が生じるため、150%拡大した画像を回転させ、中央部分の 320×240 pixel を切り出した。回転角は ± 30 度の範囲で 1 度刻みで行い、合計 61 枚の画像を生成した。結果から、回転角 ± 20 度程度までの範囲において、有効な類似度が得られることを確認できる。

以上の結果から、提案類似度は画像の一定範囲内の各種変動に対して、ロバストであることを確認できる。

4.6 他の検索手法との比較

提案手法と Wavelet 変換を用いた画像検索手法、輝度ヒストグラムによる検索結果を示し、検索性能の比較を行う。Wavelet 変換による検索手法として、Jacobsらの Fast multiresolution image querying²⁰⁾を

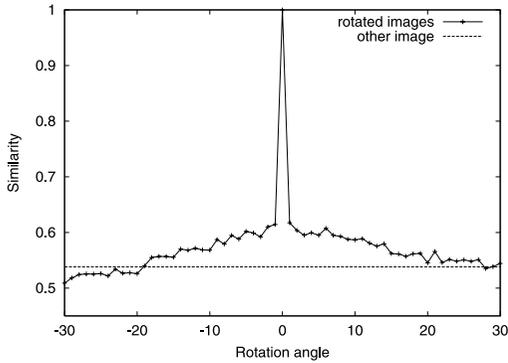


図 17 回転に対する類似度

Fig. 17 Similarity for rotated images.

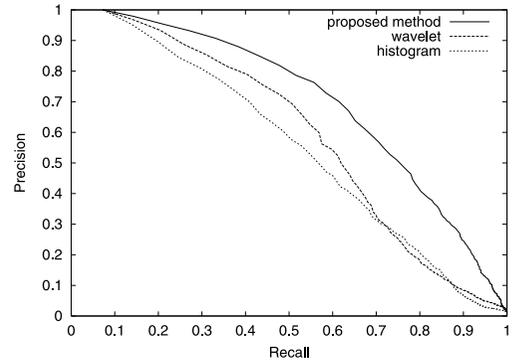


図 19 適合率と再現率：提案手法と他の検索手法

Fig. 19 Precision ratio vs. recall ratio: Proposed method and other methods.

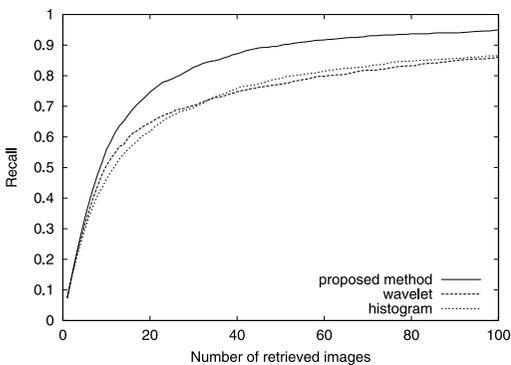


図 18 再現率：提案手法と他の検索手法

Fig. 18 Recall ratio vs. number of retrieved images: Proposed method and other methods.

用いた。この手法は Haar Wavelet を用い、得られる Wavelet 係数を量子化したものを基に検索する手法である。手法自体は圧縮符号を対象としたものではないため、使用する Wavelet 係数を上位 15% と制限することで圧縮符号と見なした。輝度ヒストグラムは、輝度値を 8 等分に量子化したものを用いた。

これらの手法について、類似画像群に対し検索した結果を示す。図 18 は検索画像数と再現率、図 19 は再現率と適合率を表す。これらのグラフから、提案手法はいずれの手法よりも検索性能が良いことが確認できる。

4.7 考 察

提案した類似度の定義は、圧縮時の領域分割のサイズが固定である場合にも適用可能ではあるが、実験結果から有効に機能しないことが分かった。類似度は写像ベクトルの 1 対 1 対応からのずれに着目しており、固定サイズの領域分割により近傍の写像との区別がつかない状態が多数発生したため、検索結果に影響したものと考えられる。

一方、可変サイズの領域分割手法により生成した圧縮符号では、類似度が有効に機能し検索が可能となった。これは、領域の特性に応じた分類をすることにより、写像ベクトルを限定することができ、写像ベクトルの 1 対 1 対応が形成されやすくなったものと考えられる。

また、可変サイズの圧縮符号データベースにおいて、最も小さいレンジサイズの類似度 s^* が高い検索性能を示した。この理由として、最も小さいレンジサイズ領域の画像の変動は一般に激しく、相似関係にある領域以外との相関が低いためであると考えられる。画像の変動によっても、その関係が変わることなく保持されるため、いずれの可変サイズの圧縮符号データベースにおいても高い検索性能を示したものと考えられる。

今回行った実験から得られた知見を以下にまとめる。

- 固定サイズの分割による圧縮符号に対して提案手法は有効に機能しない。
- 2 段階を含めた複数のレベルサイズの領域から得られる圧縮符号において、最も小さいレンジサイズの類似度 s^* が高い検索性能を示す。
- 圧縮のレンジサイズが 2 種類である場合、両レンジサイズの類似度 s^* の検索精度は等しく、これらを結合した類似度 S により検索性能がさらに改善される。
- 類似度は、画像の一定範囲内の各種変動（平行移動、拡大、回転）についてロバストである。

5. ま と め

本稿では、フラクタル画像圧縮の圧縮符号を対象とした、圧縮領域における類似検索の新たな手法を提案した。

フラクタル画像圧縮の圧縮符号は、画像内の相似領域を縮小写像によって表現する。本手法では、この相似領域の関係を画像の構造表現と考え、写像ベクトル集合という概念を導入し、画像の特徴と見なした。この写像ベクトル集合間に類似度を定義することで、画像の圧縮領域における類似検索を実現した。圧縮符号データベースを用いた実験によって、類似検索の可能性とその性質を確認した。

提案手法の類似度は、画像の領域を固定サイズで一様に分割し、生成される圧縮符号にも適用可能であるが、検索性能が高くないことを確認した。2つ以上の領域サイズを持つ圧縮符号に対して有効であり、最も小さいレンジサイズから単独で得られる類似度の検索結果が良好なことを確認した。

2種類のレンジサイズから構成される圧縮符号データベースについて、提案類似度は最も良い検索性能を示し、Waveletとヒストグラムによる検索手法との比較でも良い検索結果を示すことを確認できた。しかし、類似度算出に掛かる計算コストは高く、検索速度が遅いため、実用上改善する必要がある、今後の重要な研究課題である。

本手法により単独で検索システムを構築することが可能だが、圧縮符号に対して特別な処理や条件などを必要としないため、フラクタル画像圧縮を用いた既存の検索手法や、画素領域の手法と組み合わせることも容易である。

謝辞 本研究について電気通信大学大学院情報システム学研究所の小早川倫広氏から貴重なご助言を多数頂戴した。ここに記して謝意を表します。

参 考 文 献

- 1) 吉川正俊, 植村俊亮: マルチメディアデータのための索引技術, 情報処理学会誌, Vol.42, No.10-002, pp.953-957 (2001).
- 2) 片山紀生, 佐藤真一: 類似検索のための索引技術, 情報処理学会誌, Vol.42, No.10-003, pp.958-964 (2001).
- 3) 串間和彦, 赤間浩樹, 紺谷精一, 山室雅司: 色や形状等の表層的特徴量に基づく画像内容検索技術, 情報処理学会論文誌: データベース, Vol.40, No.SIG017 (TOD1), pp.171-184 (1999).
- 4) Mandal, M.K., Idris, F. and Panchanathan, S.: A critical evaluation of image and video indexing techniques in the compressed domain, *Image and Vision Computing*, Vol.17, No.7, pp.513-529 (1999).
- 5) Barnsley, M.F.: *Fractals everywhere*, Academic Press, San Diego (1993, 1988).
- 6) Zhang, A., Cheng, B., Acharya, R.S. and Menon, R.P.: Comparison of wavelet transforms and fractal coding in texture-based image retrieval, *Proc. Visual Data Exploration and Analysis III 2565*, pp.116-125, SPIE (1996).
- 7) Tan, T. and Yan, H.: The fractal neighbor distance measure, *Pattern Recognition*, Vol.33, No.6, pp.1371-1387 (2002).
- 8) Marie-Julie, J.M. and Essafi, H.: Digital image indexing and retrieval by content using the fractal transform for multimedia databases, *ADL '97*, pp.2-12 (1997).
- 9) Ida, T. and Sambonsugi, Y.: Image segmentation using fractal coding, *IEEE Trans. Circuits and Systems for Video Technology*, Vol.5, No.6, pp.567-570 (1995).
- 10) 上田哲史, 武内 朗, 寺田賢治: フラクタル符号を用いた画像パターン検索の一手法, 電子情報通信学会論文誌, Vol.J-82A, No.2, pp.253-257 (2002).
- 11) Welstead, S.: *Fractal and wavelet image compression techniques*, Spie Press (1999).
- 12) 兪 培, 白木教義, 徳永隆治: フラクタル変換による画像領域分割—井田の領域分割法の改善, 電子情報通信学会論文誌, Vol.J82-A, No.12, pp.1793-1800 (1999).
- 13) Ida, T. and Sambonsugi, Y.: Self-affine mapping system and its application to object contour extraction, *IEEE Trans. Image Processing*, Vol.9, No.11, pp.1926-1936 (2000).
- 14) 井田 孝: オブジェクト画像切出しソフトMaskMaster, 画像ラボ, Vol.13, No.12, pp.47-51 (2002).
- 15) 長谷山美紀, 近藤 功: フラクタル画像符号化に着想を得た著作画像の劣化を伴わない著者認証法, 電子情報通信学会論文誌, Vol.J85-D2, No.10, pp.1513-1521 (2002).
- 16) Jacquin, A.E.: Image coding based on a fractal theory of iterated contractive image transformations, *IEEE Trans. Image Processing*, Vol.1, pp.18-30 (1992).
- 17) Wohlberg, B. and de Jager, G.: A Review of the Fractal Image Coding Literature, *IEEE Trans. Image Processing*, Vol.8, No.12, pp.1716-1729 (1999).
- 18) Fisher, Y. (Ed.): *Fractal image compression: theory and application*, Springer-Verlag New York, Inc. (1995).
- 19) 横山貴紀, 渡辺俊典, 菅原 研: フラクタル符号の写像対応に基づく特徴量と類似検索について, 映像メディア学会技術報告, Vol.26, No.54, pp.29-32 (2002).
- 20) Jacobs, C.E., Finkelstein, A. and Salesin, D.H.: Fast Multiresolution Image Querying,

Proc. SIGGRAPH 95, pp.277-286 (1995).

(平成 15 年 9 月 20 日受付)

(平成 15 年 12 月 26 日採録)

(担当編集委員 市川 哲彦)



横山 貴紀 (学生会員)

2000 年電気通信大学電気通信学部電子情報学科卒業。2002 年同大学大学院情報システム学研究科博士前期課程修了。現在同大学院博士後期課程に在籍。圧縮領域における画像検索の研究に従事。

画像検索の研究に従事。



菅原 研

1992 年東北大学工学部電気工学科卒業。1997 年同大学大学院情報科学研究科博士後期課程修了。博士 (情報科学)。日本学術振興会特別研究員を経て、1999 年電気通信大学大学院情報システム学研究科助手、現在に至る。群知能ロボットシステムの研究に従事。



渡辺 俊典 (正会員)

1971 年東京大学工学部航空工学科卒業。同年日立製作所入社、中央研究所、システム開発研究所。経営生産、LSI 設計、非線形最適化、学習機械、並列分散推論 (ICOT プロジェクト) 等諸システムの開発に従事。1992 年より電気通信大学大学院情報システム学研究科教授。工学博士。電子情報通信学会、日本写真測量学会、IEEE 各会員。専門はメディアデータ自動解析および情報システム表現法。