LI-010

フラクタル符号のベクトル集合間類似度に基づく画像検索手法
Image Retrieval Exploiting the Similarity of Vectorized Fractal Code Sets

横山 貴紀 † **渡辺 俊典** † **古賀 久志** †
Takanori Yokoyama Toshinori Watanabe Hisashi Koga

1. はじめに

メディア情報の冗長度は高く，通常圧縮された状態でネットワーク上を流通し，データベースなどに蓄積される．このような圧縮されたメディア情報を対象とした検索手法の研究が近年盛んに行われている．この手法により，メディアの圧縮符号だけをシームレスに扱う検索システムの実現が可能となる．

私たちは画像メディアについて研究を行い，フラクタル符号の検索手法 [1] を提案した．この手法が画像の変動にロバストな検索特性を持ち，他の手法と比較して良好な検索精度を持つことを実証した．しかし，検索手法で提案した類似度算出に掛かる計算コストは高く，検索の度に類似度を算出する必要があることから，検索速度の実用面での課題が残っている．

この問題について本稿では，類似度算出に必要な符号データに最近接点探索の索引データ構造を適用し，計算時間の短縮を試みた．また，類似度の上限値を求め，類似度算出の対象画像数を段階的に絞り込み，検索速度の高速化を図った．

2. フラクタル符号の検索手法 [1]

フラクタル符号の検索手法について説明する．この手法では，相似領域を要素としたベクトル集合と見なし，ベクトル集合間の類似度に基づいて検索を行う．

2 枚の画像を I_A，I_B とし，得られるフラクタル符号を A，B とする．フラクタル符号には，画像中の相似領域の関係が記録されており，領域 (x_{R_i}, y_{R_i}) に相似な領域を (x_{D_i}, y_{D_i}) とすると，相似関係を 4 次元ベクトル $(x_{R_i}, y_{R_i}, x_{D_i}, y_{D_i})$ として表現することができる．このベクトルを a_i，b_i とすると，フラクタル符号は $A = \{a_1, \ldots, a_{|A|}\}$，$B = \{b_1, \ldots, b_{|B|}\}$ となる，4 次元ベクトルを要素とする集合として表現することができる．ここで $|\cdot|$ は集合の要素数を表す．フラクタル符号化手法では，相似領域の探索時に領域分割を用いるため，図 1 のように輝度値の変動度に応じて領域分割が行われ，その結果ベクトル集合の要素数は画像毎に異なる．

以上から，フラクタル符号間の類似問題は，ベクトル集合間の類似問題と見なすことができる．だが，集合 A，B の要素は同じ 4 次元の空間上に存在するものの，集合の要素が一致することはほとんどなく，共通集合 $A \cap B$ や和集合 $A \cup B$ を基に類似性を求めても，有効な検索結果を得ることができない．

そこで本検索手法では，2 つの集合をそれぞれ他方へ相互に写像し，写像による集合特性から類似度を求めることとした．写像は $f_B : A \to B$ を

$$f_B(a_i) = \arg\min_{b_j \in B} \|a_i - b_j\| \qquad (1)$$

とし，同様に $B \to A$ への写像を f_A とする．この写像の結果，2 つの集合間は図 2 のような関係になる．両方の集合が似ている場合，集合の要素は 1 対 1 に，似ていない場合は複数対 1 に写像される傾向がある．以上の関係に着目し，次の指標をベクトル集合間の類似度として定義した．

$$s(A, B) = \frac{|f_B(A)| + |f_A(B)|}{|A| + |B|} \qquad (2)$$

検索実験から，この類似度による検索手法が画像の平行移動，拡大，回転などの変動に対してロバストな検索特性を持ち，ウェーブレットを用いた検索手法との比較でも良好な検索精度を示すことを明らかにした．

†電気通信大学大学院情報システム学研究科

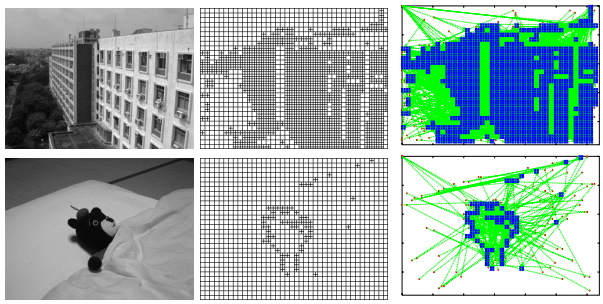

図 1: 相似領域のベクトル集合表現．左列は原画像，中央列は領域分割結果，右列はベクトル集合を表す

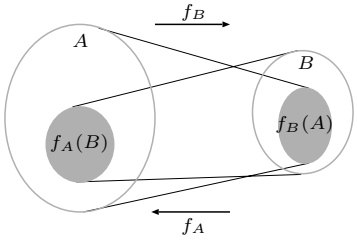

図 2: 相互の写像に基づく集合間の関係

3. 検索速度の高速化

上記提案の検索手法では，検索の度に質問画像と全ての検索対象画像との間で類似度を算出する必要がある．また，類似度算出自体に掛かる計算コストも従来手法では高く，検索対象の画像数が増えるにつれて検索時間が増大し，現実の検索システムとしては実用的ではなかった．

そこで今回，1）ベクトル集合データに索引データ構造を導入することで類似度計算を改善し，2）上限値を設定することで類似度算出の対象画像数を削減する，という 2 つの戦略を用いて検索速度の高速化を試みる．

3.1 索引データ構造を用いた類似度算出

従来手法の類似度計算では，ベクトル集合間のすべての要素間について距離行列を生成し，最小距離にある要素を全探索するものであった．そのため，集合の要素

数を $M = |A|$ と $N = |B|$ とすると，距離行列の生成に $O(MN)$，最小の距離にある要素の決定に $O(MN+NM)$ の計算コストが必要となる．

だが，式 (1) の写像は，一方の集合から取り出した要素について，他方の集合から最近接点を探索するものであり，これは索引データ構造を用いて高速に探索することが可能である．索引データ構造には，ベクトル集合の要素が 4 次元と低次元であり，最近接点の探索機能のみが必要であることから，k-d tree[2] を採用した．この手法では，木の生成に $O(M \log M + N \log N)$，対応要素の決定に $O(N(\log M + F) + M(\log N + F))$ の探索コストが掛かる（F は最近接点探索時に必要な近傍領域に含まれる要素数）．

実際に検索を行い，検索時間を計測した．検索対象画像数は 1,264 枚である（データセットの詳細や圧縮パラメータ，計算機環境は文献 [1] と同一である）．距離行列を用いる方法では，質問画像 1 枚あたり平均 221.1 秒の検索時間が必要であった．一方，k-d tree を用いた場合では，質問画像 1 枚あたり平均 77.1 秒と，索引データ構造の導入による類似度算出の改善が確認できる．

3.2 上限値による検索対象の絞り込み

索引データ構造の導入により類似度算出時間は改善されたが，画像数に比例して検索時間は増加する．検索自体の高速化には，類似度計算をできるだけ少なくする必要があり，そのためには類似度算出の対象画像を絞り込む必要がある．

類似度算出で最も計算コストを要するのは，前述した式 (1) の写像計算であり，写像を決定しない段階で検索対象を絞り込むことができるのが望ましい．式 (2) の「**類似度** s」には，次式右項のような自明な上限値が存在する．

$$s(A,B) \leq \frac{|f_B(A)| + \min(|A|,|B|)}{|A|+|B|} \leq \frac{2 \times \min(|A|,|B|)}{|A|+|B|} \tag{3}$$

この上限値を本稿では「**自明な上限値**」と呼ぶ．さらに，式 (3) 中項のように，片側の集合 A を写像し，残りの集合 B の写像によって得られる類似度の上限値を求めることができる．これを「**片側上限値**」と呼ぶ．これらの上限値を基に画像の順位をつけ，上位の画像を「類似度 s」の算出対象とする．

上限値の性能を見るために検索実験を行い，それぞれの検索精度を図 3 の「自明な上限値」および「片側上限値」に示す．実験から「自明な上限値」では，約上位 500 で再現率が 1.0 に近づくことが確認でき，この類似度を用いて類似度算出の対象を 1,264 枚から半分以下の 500 枚に絞ることができることがわかった．同じく「片側上限値」では，対象画像を 200 枚に絞ることができると見ることができる．「自明な上限値」の算出に掛かった時間は，質問画像 1 枚あたり 0.38 秒，「片側上限値」には 37.4 秒必要であった．

3.3 段階的検索

前述の「自明な上限値」と「片側上限値」の両者を用いて「類似度 s」の算出対象画像を絞り込み，検索を行う方法について説明する．この検索を「**段階的検索**」と呼び，具体的には以下の手順で行う．

step 1. すべての検索対象の画像（1,264 枚）について「自明な上限値」を求め，順位をつける．

step 2. step 1 で得られた上位の画像（1〜150 位）について「片側上限値」を求め，順位をつける．

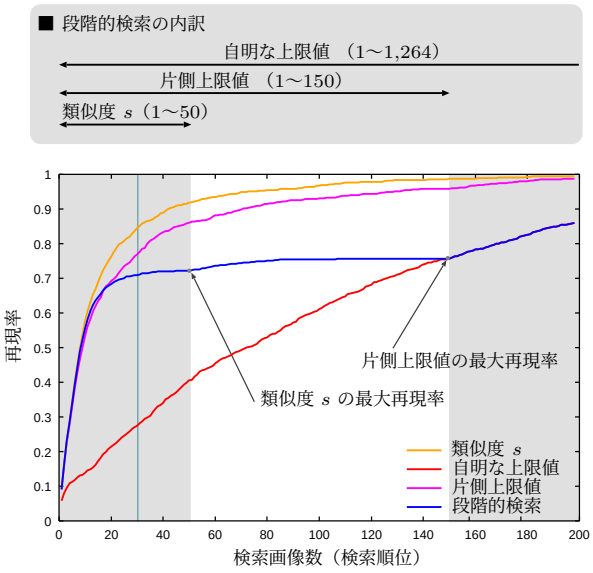

図 3: 「類似度 s」,「自明な上限値」,「片側上限値」および「段階的検索」の検索結果

step 3. step 2 で得られた上位の画像（1〜50 位）について「類似度 s」を求め，検索結果とする．

以上の手順によって得られた検索結果が図 3 の「段階的検索」である．検索画像数が 30 枚で再現率が 0.7 を超え，上位 10 以内の検索画像では「類似度 s」の結果と等しくなることが確認できる．質問画像 1 枚あたりに必要な検索時間は平均 6.4 秒と，距離行列を用いた場合の約 1/35，索引データ構造を用いた場合の約 1/12 と，高速に検索することができた．検索時間の内訳は，0.1%が「自明な上限値」，74.3%が「片側上限値」，残り 25.6%が「類似度 s」の計算であった．

3.4 今後の課題

提案した「段階的検索」では，それぞれの上限値や類似度を適用する範囲によって，最終的に得られる検索結果と検索時間が異なってくる．今回は事前に行った検索実験の結果を基にこれらを設定したが，今後は適用範囲の自動設定法などを検討する必要がある．

4. まとめ

本稿では，フラクタル符号のベクトル集合間類似度に基づく検索手法について，高速化手法を中心に報告した．ベクトル集合データに索引データ構造を導入することで類似度算出の計算コストを削減し，上限値によって類似度算出の対象画像を段階的に絞り込み，上位の検索画像の検索精度を保った上で，従来手法よりも高速に検索が可能となることを示した．

参考文献

[1] 横山貴紀, 菅原研, 渡辺俊典: 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, 情報処理学会論文誌：データベース, Vol. 45, No. SIG 4(TOD21), pp. 11–22 (2004).

[2] Bentley, J. L.: Multidimensional Binary Search Trees Used for Associative Searching, *Communications ACM*, Vol. 18, No. 9, pp. 509–517 (1975).