2008－CVIM－164 (14)
2008／9／5

自己組織化アルゴリズムを用いたデータ配置最適化による高速類似ベクトル検索

松原 大輔† 廣池 敦†

† (株) 日立製作所中央研究所 〒185–8601 東京都国分寺市東恋ヶ窪一丁目280番地
E-mail: †{daisuke.matsubara.ba,atsushi.hiroike.rx}@hitachi.com

あらまし 数百万件規模の大規模なベクトルデータを対象にした類似ベクトル検索システムでは，扱うデータ容量が膨大なため，高速アクセス可能な一次記憶だけでなく，アクセス速度の遅い二次記憶を効率的に扱う必要がある．本研究では，ベクトルデータを格納する位置を最適化することで，二次記憶を効率的に利用しつつ，高速な逐次データ登録及び高速類似検索を行うことが可能な類似ベクトル検索システムの構築を行った．その結果，二次記憶上のデータの検索速度を25.6％向上することが可能になった．

High-speed Vector Search with Data-alignment Optimization using Self-organization Algorithm

Daisuke MATSUBARA† and Atsushi HIROIKE†

† Central Research Laboratory, Hitachi, Ltd.
Higashikoigakubo 1–280, Kokubunji-shi, Tokyo, 185–8601, Japan
E-mail: †{daisuke.matsubara.ba,atsushi.hiroike.rx}@hitachi.com

Abstract In the similarity-based image retrieval system for huge amount of vector data, it is necessary to access data on the secondary storage effectively, because it is impossible to hold all vector data on the primary storage. In this paper, we describe the data-alignment method which locates similar data into closer part on the secondary storage to access at high-speed.

1. はじめに

ネットワークのブロードバンド化や記憶装置の大容量化に伴って，ユーザがアクセス可能な画像や映像の量は，日々増加の一途を辿っている．こうした流れの中で，膨大なコンテンツから，如何にして所望のものを選択するかという問題が生じている．従来は，人手もしくは半自動的にコンテンツに対してテキスト情報を付与することによってキーワード検索を可能とし，この問題を解決していた．しかし，それが可能なのは大量のコンテンツの一部のみであり，そもそもテキスト情報だけでは画像の持つ情報を的確に表すことができないという問題もある．

そこで，画像の色の分布や形状等，画像自体が持つ情報を自動的に抽出し，高次元の数値情報として表現した画像特徴量に基づき，画像間の類似性を評価して検索を行う類似画像検索技術が提案されている．しかし，類似検索を行う際には高次元の特徴量間上の距離（非類似度）を計算する必要があるため，検索対象が数百万件を超える現在，その演算量は膨大なものとなる．したがって，大規模データを対象にした類似画像検索を実現するためには，類似ベクトル検索処理の高速化が必須である．

近年，近傍探索技術の発展により，膨大な数の特徴量ベクトルを短時間で検索することが可能となってきた．短時間で検索を行うためには，予め類似したデータ同士をまとめて，索引に相当するものを作成する必要がある．索引生成技術には，主にクラスタリング[1]，木構造[2]～[4]，ハッシュ法[5],[6]，もしくはこれらを組み合わせたものなどが存在する．

これらの手法の多くは，全特徴量ベクトルを一次記憶上に展開して距離計算を行うことを前提としている．したがって，データ数が大規模になると一次記憶が不足してしまうという問題点がある．例として，1データを200次元の特徴量ベクトルで表し，1次元のスカラー値を4byteのデータとして記憶領域上に格納する場合について考える．この場合，1データに必要な容量は $200 \times 4\text{byte} = 800\text{byte}$ となる．したがって，例えば500万件のデータを扱う際には，$800\text{byte} \times 500\text{万件} = 4\text{Gbyte}$ の容量が必要となる．実運用を考えた場合，単一の特徴量のみではなく，色特徴量や輝度勾配特徴量などの複数の特徴量を用いるとすると，この容量の数倍の容量が必要となる．

以上のことから，一次記憶のみで処理を行うことは現実的ではないことが分かる．そのため，二次記憶上に特徴量ベクトルを格納し，随時二次記憶へアクセスしながら類似ベクトルを検索する必要がある．しかしながら，二次記憶へのアクセスは一次記憶へのアクセスと比較して多大な時間を要する．よって，特徴量ベクトル同士の距離計算を高速化しても，距離計算の対象となる特徴量ベクトルへのアクセス速度が遅いと，高速化の効果がほとんど得られないことになる．

この課題に対して，高次元ベクトル空間中の特徴量ベクトル同士の近傍関係をできるだけ保ちつつ，一次元の配列上にマッピングして二次記憶上に整列させる手法が提案されている [7]．この手法は，類似度が大きいデータは検索時に同時に参照される可能性が高いという考えに基づくものである．つまり，類似度が大きいデータ同士を二次記憶上でなるべく近傍に集まるように配置し，二次記憶を走査する際の負荷を低減することを目指している．しかし，マッピングを行う際に，自己組織化マップを用いているため大量の計算時間を必要とし，さらにはデータ追加に対応していないなどの問題点がある．

本研究では，数百万件規模の大規模な画像データを対象とし，二次記憶を効率的に利用しつつ，高速な逐次データ登録及び高速類似検索を行うことが可能な類似ベクトル検索システムの構築を目指す．本研究で提案する手法は [7] と同様に，類似度が大きいデータは検索時に同時に参照される可能性が高いという考えに基づき，二次記憶を一次元配列に近似し，類似度が大きいデータ同士を近傍領域に配置する．したがって本研究が対象とする問題は，高速検索を可能とする一次元配列の最適化問題であると言える．

2. 提案システム

2.1 システム概要

まず初めに本システムのデータ管理方式について述べる．本システムでは，類似したデータを集めたクラスタを作成して，クラスタの代表値のみを一次記憶上に保持し，特徴量ベクトルは二次記憶上に格納する．ここで，クラスタ代表値をクラスタ内に含まれる特徴量ベクトルの平均値とし，以下クラスタ平均と呼ぶ．クラスタには二次記憶上の一定の領域を割り当て，各クラスタに含まれる特徴量ベクトルを連続した領域内に格納する．このとき，特徴量ベクトルが格納されていない領域は空領域とする．このような特徴量ベクトル管理方式を図 1 に示す．図中の i を以下では，クラスタ番号と呼ぶ．

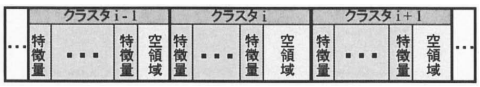

図 1　特徴量ベクトル管理方式

本システムではこのように管理されているベクトルデータを階層的に検索する．つまり，クエリデータと類似したクラスタを検索し，その後クラスタ内に含まれる全特徴量ベクトルを検索する．よって，類似したクラスタ同士を近傍領域に配置すれば高速アクセスが可能になると考えられる．

データ追加に対応するためには，上記クラスタを動的に作成・更新しつつ整列させる必要がある．そこで，登録データと類似した一部のクラスタのみを対象にクラスタリングを実行することによって，この問題を解決する手法を提案する．また，クラスタを整列させる手法として，局所的最適化と大局的最適化の二つの最適化を組み合わせた配置手法を提案する．局所的最適化では，類似したクラスタ平均を持つクラスタ同士を，一次元配列上で隣接した領域に配置することを目指す．一方，大局的最適化では，一次元配列を複数の領域に分割し，各クラスタを適切な領域に配置することを目指す．各々を最適化するエネルギー関数を定義し，特定のクラスタの格納位置を交換することによって，最急降下法を用いて最小値を探索し，最適化を行う．データ登録処理の流れは以下のようになる．

（1） 登録データと類似したクラスタを複数個取得
（2） 取得したクラスタのみを対象にクラスタリングを実行
（3） 更新クラスタの格納位置を決定
（4） 更新クラスタに含まれる特徴量ベクトルを格納

新規データが追加されるたびに上記処理を繰り返すことによって，データ登録を行う．

2.2 逐次クラスタリング手法

本手法では，k-means 法に基づいたクラスタリングを採用している．ただし，データ登録のたびに，全データのクラスタリングを行ったのでは，実用的な処理時間で登録処理を行うことは不可能である．そこで，新規データ登録時には，まず新規登録データの特徴量ベクトルとベクトル間距離が近いクラスタ平均を持つクラスタを，近接クラスタとして所定の個数検索し，近接クラスタに含まれる特徴量ベクトルのみを対象にしてクラスタリングを実行する．

図 2 は，データ登録時に実行されるクラスタリング処理を示すフローチャートである．以下，このフローチャートを用いて，逐次クラスタリングについて詳細に説明する．

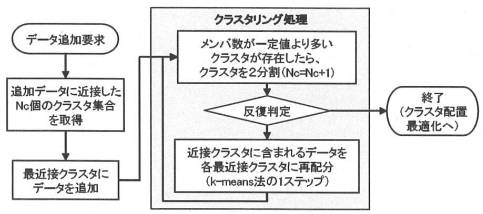

図 2　逐次クラスタリング

登録対象である新規データが与えられると，まず全クラスタを対象に検索を行い，新規データから抽出した特徴量ベクトルをベクトル間距離の近いクラスタ平均を持つ近接クラスタの集合 $C*$ を取得する．次に，近接クラスタの集合 $C*$ の中の最近接クラスタ $c*$ に，新規データを追加する．

その後，クラスタリング処理のループに入る．まず，近接クラスタの集合 $C*$ の要素である各クラスタについて，クラスタのメンバ数が制限を超えるか否かを判定する．仮に最近接クラスタ $c*$ のメンバ数が制限を超えた場合，そのクラスタを二分割し，新たに生成されたクラスタを近接クラスタの集合 $C*$ の要素に加える．クラスタを二分割する際には，クラスタ内のベクトル分布に関して主軸を求め，各メンバのベクトルの主軸への射影とクラスタ平均ベクトルの主軸への射影との大小に従って，メンバを二つのクラスタに分割する．

反復判定処理では，反復回数が反復の最大数 t_{max} 以上であると判定された場合，もしくは集合 $C*$ が変化していないと判

定された場合に，クラスタリング処理のループを抜ける．一方，反復回数が反復の最大数 t_{max} より小さく，かつ，集合 $C*$ が変化していると判定された場合， k-means 法を 1 ステップのみ実行して集合 $C*$ を更新する．すなわち，近接クラスタに含まれる全データ及び新規追加データは，その時点で最も近接したクラスタ平均を持つクラスタに配分される．この処理によって，各近接クラスタのメンバ及びクラスタ平均が更新される．

本方式におけるクラスタリング処理は，あくまでクラスタの部分集合を対象としたものであり，クラスタ全体での最適化とは厳密には異なる．しかしながら，全クラスタを対象に最適化を行ったとしても，追加データとベクトル間距離の遠いクラスタが更新される確率は小さいと考えられる．したがって，本手法によって求められる解は近似解とみなすことができる．また，データの追加は，繰り返し行われることを想定しており，その度に最適化が実行される．よって，実運用を考えた場合，ある時点での最適化を極端に重視する必要はなく，クラスタリング処理の反復の最大数 t_{max} は数回程度に抑えることができる．

以上の処理によって，登録データ数が増加するにつれて，クラスタ数も増加することになる．新クラスタが追加されるのは，クラスタ内のメンバ数がクラスタメンバ最大数 M を超えたときのみのため，M 回に 1 回程度の頻度となる．また，近接クラスタを取得する際に線形探索を行っているため，処理時間のオーダはクラスタ数を N とすると $O(N)$ となる．このとき取得する近接クラスタ数を N_c とすると，クラスタリング処理時間はデータ数 N に依存せずに一定時間 $O(M \times N_c^2)$ となる．

2.3 データ配置最適化手法

本手法では，二次記憶を一次元配列に近似し，類似したクラスタ平均を持つクラスタ同士が一次元配列上の近傍領域に配置されるように最適化処理を行う．そこで，以下の項で述べるエネルギー関数を定義し，それらのエネルギー関数が小さくなるように，二次記憶上のクラスタ位置を交換することによって，最急降下法を用いてクラスタ配置最適化を行う．以下に，前節で述べたクラスタリング処理後に実行される本手法の詳細を述べる．

まず，クラスタ位置を交換することによって，エネルギー関数から算出されるエネルギーの減少量が最大となるクラスタの組を，現在更新対象となっている近接クラスタの集合 $C*$ から探す．該当するクラスタの組が発見された場合，それらの位置を交換し，次にエネルギーの減少量が大きいクラスタの組を探す．該当するクラスタの組が発見されない場合は，現在のクラスタの配列のエネルギーが最も小さいため，処理を終了する．最後に，得られた配列上の位置に従って，クラスタメンバの特徴量ベクトルを二次記憶上へ保存する．

なお，逐次クラスタリング処理において，クラスタ数が増加した際には，現在確保されている二次記憶領域の末尾に新たに領域を確保する．これらの処理を，逐次クラスタリング処理と同様にデータ登録のたびに繰り返し行うことによって，各クラスタを最適な位置に配置する．

(1) 局所的最適化を行うエネルギー関数

局所的な最適化を行うエネルギー関数として，式 (1) に示される関数について考える．ここで，N はクラスタ数，v_i は i 番目の位置にあるクラスタのクラスタ平均を表す．

$$E_1 = \frac{1}{2}\sum_{i=1}^{N}(\|v_i - v_{i+1}\|^2 + \|v_i - v_{i-1}\|^2) \qquad (1)$$

本エネルギー関数は，一次元配列上で相前後する位置にあるクラスタのクラスタ平均の二乗距離の総和として定義される．ただし，両端の位置，すなわち 1 番目の位置と N 番目の位置のクラスタについての境界条件は，片側のみを定義している．すなわち，存在しない 0 番目の位置のクラスタと 1 番目のクラスタとのクラスタ平均同士の距離は 0 とし，存在しない $N+1$ 番目の位置のクラスタと N 番目の位置のクラスタとのクラスタ平均同士の距離も 0 とする．

この時，i 番目の位置にあるクラスタと j 番目の位置にあるクラスタを交換した場合のエネルギー関数の変化量は，式 (2) によって算出される．ただし，$i < j$ とする．

$$\Delta E_1\,(i \leftrightarrow j | j > i) = \begin{cases} (v_i - v_j)^T(v_{i-1} - v_{j+1}) & (j = i+1) \\ (v_i - v_j)^T(v_{i-1} + v_{i+1} - v_{j-1} - v_{j+1}) & (j > i+1) \end{cases} \qquad (2)$$

このエネルギー変化量に基づき，最急降下法を用いてエネルギー関数が減少するように配列内のクラスタ位置を更新すれば，一次元配列中で隣り合う位置に存在するクラスタ同士の類似度が相対的に大きい状態が実現できる．また，逐次クラスタリング処理で取得した近接クラスタ数を N_c とすると，クラスタ数に依存しない $O(N_c^2)$ の処理時間が必要となる．なお，本エネルギー関数 E_1 による配置最適化手法は，一次元の自己組織化マップをエネルギー関数を用いて再定式化したものと同等であるとみなすことができる．

(2) 大局的最適化を行うエネルギー関数

エネルギー関数 E_1 のみを用いた場合，隣接クラスタ同士の関係のみを最適化しているため，一次元配列上で一定以上離れたクラスタ同士がお互いに及ぼす影響は小さい．したがって，データの規模が増大するにつれて，類似した少数のクラスタが集まったクラスタ群が散在してしまう．よって，隣接クラスタ同士の関係だけではなく，各クラスタが一次元配列全体において，どこに配置されるべきかという大局的な情報もエネルギー関数に加えることを考える．そこで，一次元配列を複数の領域に分割し，類似したクラスタを同一の領域に配置することが可能なエネルギー関数を定義する．

こうしたエネルギー関数を定義するためのアプローチを以下に示す．まず，どのクラスタについても，類似度が高いクラスタ同士が一次元配列の近傍に配置されている状態を理想的な状態と仮定する．この理想状態において，一次元配列の持つ構造について考察する．ここでは，単純化のためにクラスタ数が 12 の時について図 3 に示す．最下層の矩形が理想状態の一次元配列のクラスタ平均を表す．最下層の配列を適当な基準で複数の領域に分割すると，各領域に含まれるクラスタ平均の平均値を計算することができる．このクラスタ平均の平均値 u を，以下

では上層クラスタ平均と呼ぶ．最下層の一次元配列と同様に，類似した上層クラスタ平均同士が近傍領域に配置されており，このような関係は階層的に成立していると考えられる．

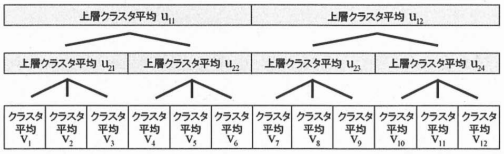

図 3　理想的な一次元配列の持つ構造

逆に，最下層のクラスタを並べるより前に上層クラスタ平均を知ることができれば，各上層クラスタ平均を基準として，それらの値に距離が近いクラスタ平均を持つクラスタを集めることができると考えられる．つまり，一次元配列全体における凡そのクラスタ位置を決定することが可能になる．

しかしながら，図 3 に示す一次元配列の構造は理想状態のものであるため，上層クラスタ平均は未知である．それゆえに，何らかの手段を用いて上層クラスタ平均を求める必要がある．

ここで，理想状態の一次元配列から得られる上層クラスタ平均が何を意味するのかについて考える．これらは，上層クラスタ平均と同じ個数に画像特徴量分布全体を分割する最適なクラスタリング結果から得られるクラスタ平均とほぼ等しいと仮定することができる．なぜならば，上層クラスタ平均は，理想状態の最下層の一次元配列に対して，類似したクラスタが多く集まっている領域を細かく分割し，あまり集まっていない領域は粗く分割している．その一方，クラスタメンバに上限を設けたクラスタリングも，このような状態を生成する手法であると考えることができるからである．

そこで，2.2 節で述べた逐次クラスタリング処理に着目する．逐次クラスタリング処理では，部分的なクラスタリングを繰り返し行い，全体のクラスタリングを実行する．また，クラスタメンバ数の上限を予め決めているため，データ数が増加するにつれてクラスタ数も同様に増加するという特徴を持つ．逐次クラスタリングによるクラスタリングの様子を，高次元特徴量分布を二次元に簡略化して図 4 に示す．●印が特徴量ベクトル，×印がクラスタ平均であり，楕円がクラスタを表す．

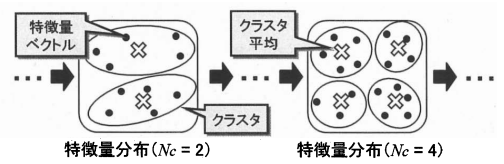

図 4　逐次クラスタリングによるクラスタリング結果の系列

逐次クラスタリングから得られるクラスタ平均の系列は，登録データ数が増加してクラスタ数が増えるにつれて，画像特徴量の分布全体を粗く近似している状態から，細かく近似している状態に変遷すると考えることができる．つまり，特徴量分布中において，画像特徴量が多く存在している領域には多くのクラスタ平均が存在し，逆に画像特徴量があまり存在していない領域には少数のクラスタ平均しか存在しないと言える．これは特徴量分布全体を，各分割領域内の画像特徴量の生起確率を一定とするベクトル量子化によって，離散的に表すことと等しいと考えることができる．よって，登録するデータのランダム性を仮定すると，逐次クラスタリングの結果として得られるクラスタ平均は，全画像特徴量をクラスタリングした結果の近似値を与えると考えることができる．

以上のことから，逐次クラスタリングの結果として得られるクラスタ平均を，上層クラスタ平均の近似値として用いる．大局的なクラスタ位置を決定するエネルギー関数を以下のように定義する．データ登録の際に，クラスタ数が $2^l (l > 0)$ の時のクラスタ平均 $U_l = (u_{l1}, \ldots, u_{lm})$ を保存する．u_{lm} はクラスタ数が 2^l の時の m 番目のクラスタ平均である．これらの過去クラスタ平均を用いて，クラスタ数が N の時のエネルギー関数 E_2 を定義する．

$$E_2 = \sum_{i=1}^{N} \sum_{l=1}^{p-1} \left\| v_i - u_{lq(l,i)} \right\|^2 \qquad (3)$$

ここで p は $2^p < N$ を満たす最大の整数であり，$p = O(\log N)$ である．また，$q(l,k) = \lfloor k \times 2^l / N \rfloor$ とする．N はクラスタ数，v_i は i 番目の位置にあるクラスタのクラスタ平均を表す．本エネルギー関数は，クラスタ平均と逐次クラスタリングの結果として得られた過去クラスタ平均の 2 乗距離の総和を表す項の和として定義されている．したがって，i 番目のクラスタが過去のクラスタ平均に対応した位置に配置されれば，エネルギー関数 E_2 は小さくなる．

このとき，i 番目の位置にあるクラスタと j 番目の位置にあるクラスタを交換した場合のエネルギー関数の変化量は，式 (4) から算出される．

$$\Delta E_2 \left(i \leftrightarrow j | j > i\right) = \left(v_i - v_j\right)^T \sum_{l=1}^{p-1} \left(u_{lq(u,i)} - u_{lq(l,j)}\right) \quad (4)$$

本エネルギー関数を用いた配置最適化処理も $O(N_c^2)$ の時間を要するため，処理時間はデータ数に因らず一定となる．

本手法では，このようにして得られた二つのエネルギー関数 E_1 と E_2 の和 E を用いて，クラスタ配置の最適化を行う．

3. 評価実験

3.1 実験内容と実験条件

二次記憶アクセスの効率化の観点から，(a) 配置最適化なし，(b) 局所的最適化のみ，(c) 局所的＋大局的最適化，の三つの方式の比較を行う．まず初めに，一次元配列上のクラスタ同士の関係を評価するために，一次元配列上の距離とクラスタ平均間の距離の関係について示す．次に，データ登録速度，検索精度及び検索速度について，二次記憶アクセスの効率化の観点から，(a)，(b)，(c)，の三つの方式の比較実験を行う．

実験データとして，映像データから OpenCV を用いて検出した顔画像 500 万件を用いた．この顔画像から輝度勾配に基

づく 200 次元の特徴量ベクトルを抽出して実験を行った．また，検索時間を計測する際は 1000 回検索を行った結果の平均値を用いた．さらに，OS によって一次記憶上にキャッシュされたデータが時間計測の結果に影響を及ぼさないように，検索のたびにキャッシュをクリアして評価を行った．使用した計算機は，CPU が Core2Duo3.0GHz，一次記憶が 4GB，OS が VineLinux4.2 である．

3.2 一次元配列上の距離とクラスタ平均間の距離の関係

500 万件の特徴量ベクトルを全て登録した結果，12245 個のクラスタが得られた．なお，クラスタメンバ数 M を 600，近接クラスタ取得数 N_c を 6 とした．クラスタ同士の関係を可視化するために，一次元配列上の距離が等しいクラスタの組を求めて，各組のクラスタ同士のクラスタ平均間のベクトル間距離を計算し，それらの平均値をプロットした．ここで，一次元配列上の距離とは，図 1 のクラスタ番号の差の絶対値である．

図 5 に全クラスタについての結果を示す．縦軸がクラスタ平均のベクトル間距離，横軸が一次元配列上の距離を表す．

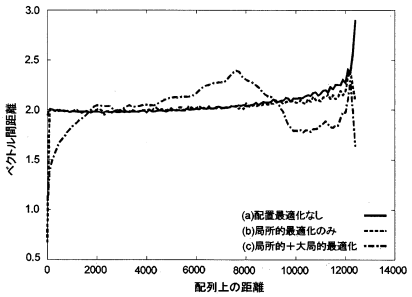

図 5　一次元配列上の距離とクラスタ平均間の距離の関係（全体）

図 6 に近傍領域（一次元配列上の距離が 200 以下のクラスタの組）のみを拡大してプロットした結果を示す．

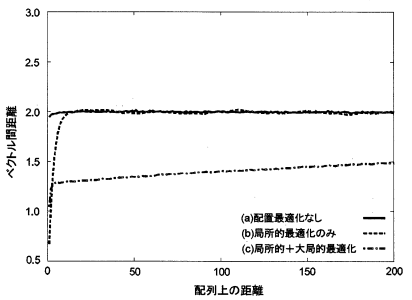

図 6　一次元配列上の距離とクラスタ平均間の距離の関係（近傍）

3.3 登録速度評価実験

登録処理中には，近接クラスタ検索処理，クラスタリング処理，クラスタ配置最適化処理，データ格納処理が存在する．各処理について，新規データを 1 つ登録するための時間を計測した結果を図 7，図 8，図 9 に示す．横軸がデータ数であり，縦軸が登録時間（ミリ秒）である．また，実線が全処理時間，点線が近接クラスタ検索時間，破線がクラスタリング処理時間，一点鎖線がクラスタ配置最適化処理時間，二点鎖線がデータ格納処理時間を表している．

3.4 検索精度評価実験

本実験では，検索精度を「本方式の検索結果について，全件探索を行った際の検索結果と比較した際の再現率」と定義する．なお，検索精度は，(a)，(b)，(c) の三方式において等しくなる．全件探索の検索結果と，上位 1 件，10 件，100 件を比較した結果を表 1 に示す．

表 1　検索精度

	上位 1 件	上位 10 件	上位 100 件
検索精度	1.00	0.898	0.758

3.5 検索速度評価実験

検索処理中には主に，近接クラスタ検索処理，近接クラスタ内データアクセス処理，近接クラスタ内データのベクトル間距離計算処理，が存在する．各処理時間を計測した結果を図 10 に示す．ここで，縦軸が時間（ミリ秒）を表している．

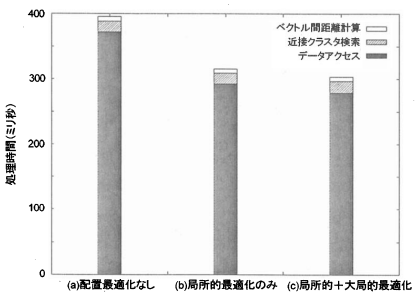

図 10　検索時間の比較

4. 考　　察

初めに，一次元配列上に配置されたクラスタ同士の関係について考察を加える．図 5 から分かるように (a) 及び (b) は，一次元配列上の距離が一定以上離れたクラスタ同士の類似度の平均値がほぼ一定であることから，ほとんど無秩序な状態になっている．一方，(c) については，一次元配列上の距離が遠くなるにつれて，類似度が小さいクラスタが配置されている様子が分かる．しかしながら，(c) においても配列全体の 3 分の 2 以上距離が離れたクラスタ同士に関しては，そのような関係を構築できていない．また，配列上の距離が近いクラスタ同士のみに着目してみると，図 6 から分かるように，隣接したクラスタ同士の類似度が大きいのは (b) である．しかし，(b) は類似度が近いクラスタ同士が集まっている範囲が (c) と比較すると非常に狭いことが分かる．このようなクラスタ配置がアクセス速度にどのような影響を与えるのかについて，以下考察を加える．

まず，登録速度について考察を加える．本システムの実際の運用形態として，監視カメラで撮影された画像から検出された

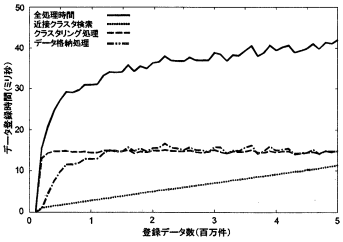

図 7　登録時間：(a) 配置最適化なし

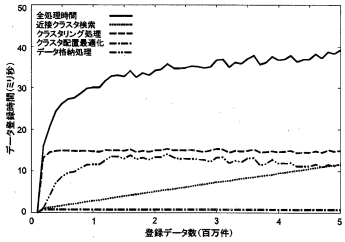

図 8　登録時間：(b) 局所的最適化のみ

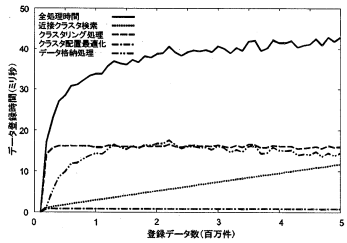

図 9　登録時間：(c) 局所的＋大局的最適化

顔画像を検索対象とする場合について考える．100 台の監視カメラが接続されたシステムの場合，1 台のカメラから 5 秒に 1 回の頻度で顔が検知されると仮定すると，1 データの登録を 50 ミリ秒以内に終える必要がある．図 7，図 8，図 9 の登録速度全体を示したグラフから，どの方式においてもこの条件は満たされていることが分かる．

また，データ格納処理時間については，(a) と比較して (b) 及び (c) の方が小さい様子が分かる．この理由として，本方式のデータ配置最適化によって，類似したクラスタ同士が近傍に集まっているため，データ格納時のシーク距離が削減されていると考えられる．また，本方式では，クラスタリング及びクラスタ配置最適化後にデータ格納をする際に，格納メンバもしくは格納位置に変化のないクラスタに関しては，データ格納処理を省略している．したがって，クラスタ配置最適化によって，クラスタ位置の交換が (b) よりも (c) の方が頻繁に行われているために，(c) のデータ格納量が増加し，(b) よりも処理に時間がかかっていると考えられる．

次に，検索精度について考察を加える．類似画像検索を行う際は，ユーザとのインタラクションを通じて所望の検索結果を得ることが多々あるため，近傍検索の実行回数が必然的に多くなってしまう．このような場合，完全な近傍探索よりも高速な近似最近傍探索の方が望ましい．表 1 から分かるように上位 10 件の検索結果を見た場合，全件探索と比較して 9 割が正解しているため，実用上十分な精度であると考えられる．

最後に，検索速度について考察を加える．近接クラスタ検索やベクトル間距離計算といった一次記憶処理のみで行われる処理に対して，二次記憶へのアクセスには多大な時間がかかっており，二次記憶アクセスの高速化が検索時間の短縮に与える影響は大きいと考えられる．したがって，1 章で述べた既存手法との比較を行う場合，一次記憶処理上での処理時間の比較はほとんど意味を成さないため，ここでは既存手法との比較は行わないこととする．

図 10 に示される結果から，(a)，(b)，(c) の順に検索時間が短くなっていることが分かる．特に (c) は (a) と比較して，処理時間を約 25.3 %削減することができた．しかしながら，(c) を (b) と比較した場合には，大局的最適化の効果が薄い．この理由として，クラスタ配置可視化の項で述べたように，一次元配列上の最近傍領域に類似したクラスタが配置されている程度が，(b) よりも (c) の方が小さいことが挙げられる．

以上のことから，登録速度に関しては (b) が最も処理時間が短いが，検索速度に関しては (c) が最も早い結果となった．したがって，本手法の適用対象が登録速度と検索速度のどちらを重視するかによって，手法を選択する必要がある．

5. ま と め

本稿では，一次記憶のみでの処理が困難な数百万件規模の大規模なベクトルデータを対象にした類似ベクトル検索システムの研究について報告した．逐次的なデータ登録及び効率的なディスクアクセスを行うことが可能な高速類似ベクトル検索手法について提案し，有効性の検証を行った．

逐次的なデータ登録のために，登録データと類似したクラスタ内のデータのみを対象にした部分的なクラスタリングを繰り返すことによって，大規模なベクトルデータ全体を実用的な精度でクラスタリングする手法を提案した．また，効率的なデータアクセスを実現するために，類似したクラスタ同士を自己組織的に二次記憶上の近傍領域に配置する手法を提案した．

大量データを対象にする検索システムの実運用を想定した場合，検索処理よりも登録処理の方が重要であると考えられる．今後は，同時に大量のデータが登録される状況を想定し，逐次登録処理の高速化を推し進めていく予定である．

文　献

[1] 宮本定明：“クラスター分析入門: ファジィクラスタリングの理論と応用”，東京，森北出版 (1999).

[2] 片山，佐藤：“マルチメディア情報の大規模処理に向けた多次元インデクシング手法の応用”，電子情報通信学会論文誌. D-II, **82**, 10, pp. 1606–1616 (1999).

[3] S. Arya, D. Mount, N. Netanyahu, R. Silverman and A. Wu: “An optimal algorithm for approximate nearest neighbor searching fixed dimensions”, Journal of the ACM (JACM), **45**, 6, pp. 891–923 (1998).

[4] T. Zhang, R. Ramakrishnan and M. Livny: “Birch: an efficient data clustering method for very large databases”, Proceedings of the 1996 ACM SIGMOD international conference on Management of data, pp. 103–114 (1996).

[5] M. Datar, N. Immorlica, P. Indyk and V. Mirrokni: “Locality-sensitive hashing scheme based on p-stable distributions”, Proceedings of the twentieth annual symposium on Computational geometry, pp. 253–262 (2004).

[6] 野口，中居，黄瀬，岩村：“特徴ベクトルの近傍探索と物体認識の効率に関する実験的検討”，情報処理学会研究報告. CVIM, 93, pp. 57–64 (2006).

[7] 北，獅々堀：“1 次元自己組織化マップを用いた高次元データの高速近傍検索”，電子情報通信学会技術研究報告. NLC, **102**, 199, pp. 29–34 (2002).