類似検索のための索引技術

片山紀生　国立情報学研究所情報メディア研究系　katayama@nii.ac.jp
佐藤真一　国立情報学研究所ソフトウェア研究系　satoh@nii.ac.jp

画像情報，映像情報の電子化はもはや日常的な存在であり，従来にない規模の画像や映像のディジタルアーカイブが実現可能となりつつある．それに伴い，キーワードによる検索ではなく，画像や映像の内容に基づく検索が求められている．画像や映像は内容のバリエーションが大きいため，利用者の要求するものがデータベース中のものと完全に一致していることは考えにくく，何らかの観点で似ているものを探さざるを得ないことが多い．そのため，画像や映像の内容に基づく検索を実現するためには，類似性に基づくデータの検索，すなわち類似検索が不可欠な構成要素となっている．本稿では，大規模な画像・映像アーカイブを対象とした類似検索のための索引技術について紹介する．

画像や映像のディジタルアーカイブ

ディジタルカメラやDVDの普及に端的に現れているように，画像情報，映像情報の電子化はもはや日常的な存在であり，従来にない規模の画像や映像のディジタルアーカイブが実現可能となりつつある．特に，ここ数年のプロセッサ技術，ストレージ技術の発展には目を見張るものがあり，数年前には到底実現し得なかった規模のディジタルアーカイブを構築できるようになっている．

このようなアーカイブは，データベースの一種であり，従来のテキスト情報によるデータベースと同様に扱うことも考えられる．たとえば，画像や映像に対してキーワードなどのテキスト情報を付与すれば，テキスト情報のデータベースと同様に画像や映像をキーワードによって検索することができる．ところが，キーワードなどのメタ情報は，常に得られるわけではないため，画像や映像そのものの内容による検索がしばしば求められる．たとえば，大量の写真の中から，特定の写真と見た目がよく似たものを見つけたい場合（**図-1** (a)）や，映像中の顔画像の中から，特定の顔画像と似たものを見つけたい場合（**図-1** (b)）などは，画像や映像そのものの内容によって検索することが必要になる．そのような内容そのものに基づく検索のことは，一般に，内容に基づく検索（content-based retrieval）と呼ばれる．

内容に基づく検索を実現するためには，テキスト情報には見られない画像や映像に固有の課題を解決しなければならない．画像や映像の場合，内容のバリエーションが大きいため，利用者の要求するものがデータベース中のものと完全に一致していることは考えにくい．そのため，完全に一致，不一致を断定できることは少なく，何らかの観点で似ているものを探さざるを得ないことが多い．そのため，画像や映像の内容に基づく検索を実現するためには，類似性に基づくデータ

の検索，すなわち類似検索（similarity retrieval）が，不可欠な構成要素となる．

画像や映像を対象としたデータベースの研究は1980年代から活発に行われているが，1990年代半ばごろから大規模なデータベースを対象とした索引技術の研究が活発になり新たな展開をみせている．本稿では，そのような大規模な画像・映像アーカイブを対象とした索引技術について紹介する．

内容に基づく検索

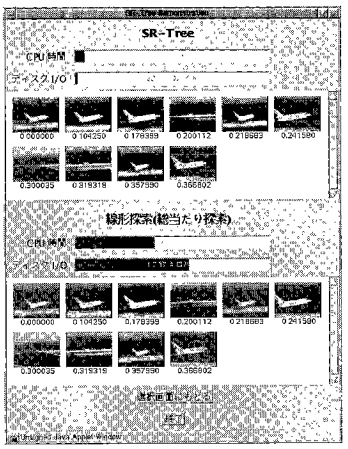
大規模アーカイブを対象とした類似検索

画像や映像の大規模アーカイブを対象とした類似検索は，一般に次の2段階の方法によって実現される．

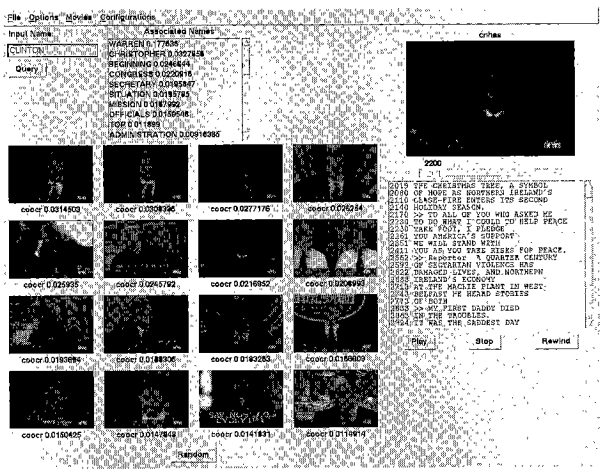
(1) 特徴量ベクトルの抽出

あらかじめ画像や映像を解析し特徴量を抽出しておくことで，特徴量どうしの比較によって類似しているものを検索する（**図-2**）．画像や映像が似ているかどうかは人間の主観に依るところが大きく，人間の認識過程が完全に解明されることがない限り，精確に数値化することは困難である．そこで，何らかの観点で似ているものを判定できるよう便宜的に設計した特徴量を計算し，特徴量が近いものどうしを似ているものと判定する．たとえば，カラー画像に対する特徴量として広く使われているものに，カラーヒストグラムがある．カラーヒストグラムは，画像中の色の出現頻度を計算したものであり，**図-3**のようにピクセルの色を何種類かに分類し，各色の割合を特徴量とするものである．これにより，色の出現割合が近いものどうしを似たものとして判定することが可能になる．たとえば，図-3では，(19，11，8，8，16，22，4，12）が特徴量となる．この例のように，特徴量は複数の数値を組にしたものが多く，そのような特徴量は特徴量ベクトルと呼ばれる．そして，それらのベクトルが存在する空間は特徴量空間と呼ばれる．

画像や映像の類似度は，特徴量空間における距離によって判定することになる．距離が近いほど類似度が高く，距離が遠いほど類似度が低いと判定する．したがって，データベースから最も似たものを探すという処理は，特徴量空間においてある点と最も近い点を見つける処理，すなわち，最近接点探索（nearest neighbor search）を行うことになる．距離の計算法は，ユークリッド距離（L_2距離）を始め，L_1距離など種々のものが考えられるが，ユークリッド距離が使われることが多い．



(a) 写真画像の検索



(b) 映像中の顔画像の検索（Name-It）

図-1 内容に基づく検索の例

(2) 特徴量ベクトルの索引付け

特徴量ベクトルによる類似性の判定は，上述のように距離の計算に過ぎないので比較的簡単な処理である．しかし，すべての特徴量ベクトルと機械的に比較したのでは，データベースの規模に比例した処理が必要となり，たとえば，データベースの規模が100倍になると，同じく100倍の処理が必要となり，システムの処理効率（レスポンスやスループット）に深刻な影響を与える．そのため，特徴量ベクトルに対する索引を構築し，探索を高速化することが必要になる．たとえば，特徴量ベクトルの要素として数値をとるものがあり，その値に対してB-treeを構築すれば，その要素について特定の範囲にあるベクトルを高速に見つけることが可能にな

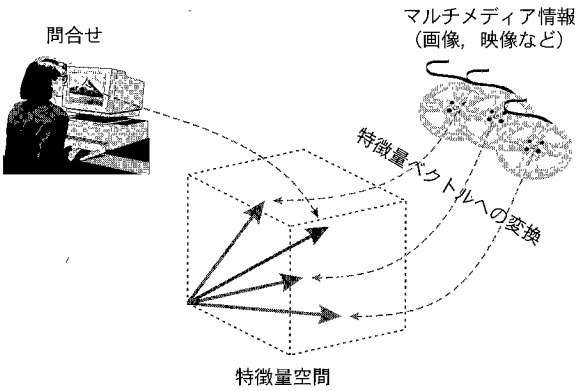


図-2　特徴量空間による内容に基づく検索

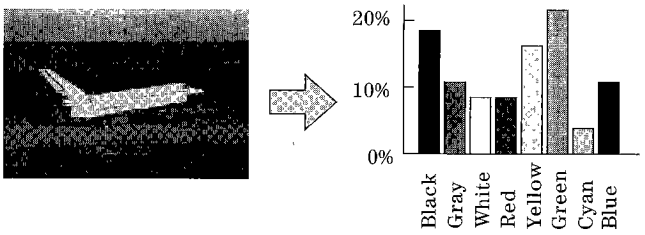


図-3　カラーヒストグラム

る．また，特徴量空間が，ユークリッド空間やメトリック空間など，幾何的性質のよい空間と見なせる場合には，R-treeなどの多次元索引を使うことによって高速化することも可能である．

内容に基づく検索の研究動向

上記のアプローチに基づくシステムとしては，1995年ごろに発表されたIBMのQBICシステムがよく知られている[1), 2)]．このシステムは，大規模な画像・映像データベースに対する内容に基づく検索の実現法を，初めて具体的に示したものであり，広く関心を集めた研究である．特徴量ベクトルとしては，カラーヒストグラム，テクスチャ特徴量，形状特徴量が使われている．この時期を境に，大規模アーカイブを対象とした類似検索のための索引技術について，さまざまな研究が行われるようになった．

(1) 画像や映像の特徴量について

画像や映像の特徴量については，パターン認識の分野で古くから研究が行われており，たとえば，指紋照合，商標照合などの目的で1980年代にはすでに研究が行われている．しかし，これらの研究は，具体的な対象に特化されており，近年見られるような多様な画像や映像を大規模に収集したアーカイブを対象としたものではなかった．そのため，そのようなアーカイブを対象とした特徴量として，Wavelet FeatureやColor Correlogramなど，さまざまな特徴量が主に画像処理，パターン認識分野で提案されている．

(2) 特徴量ベクトルの索引技術について

QBICでは，多次元索引としてR*-treeが使われていたが，1996年には，特徴量ベクトルの類似検索のための索引構造としてWhiteらがSS-tree[3)]ならびにVAMSplit R-tree[4)]という索引構造を提案している．これらの索引構造の大きな特徴は，高次元空間での類似検索に焦点を合わせた初めての索引構造であることである．その後，現在に至るまで，データベースシステム分野において，活発に研究が続けられている．

以上のように，内容に基づく検索については，大きく分けて，特徴量に関する研究と索引技術に関する研究があるが，今回の特集では，索引技術に焦点を絞り，これを中心に説明を進める．

類似検索のための高速索引技術

特徴量ベクトルのための索引技術の課題

特徴量ベクトルを対象とした類似検索は，距離の計算法として，ユークリッド距離やL_1，L_∞などの距離を使う場合，多次元空間中での最近接点探索を行うことになる．多次元空間中での最近接点探索は，計算機科学分野の古典的な問題であり，k-d treeやquad treeなどの基礎的なデータ構造を対象として，1970年代からすでにさまざまな研究が行われている．ところが，当時の研究は，比較的低次元を対象としており，また，データセットの規模も比較的小さいものであった．それに対して，内容に基づく検索で使われる特徴量ベクト

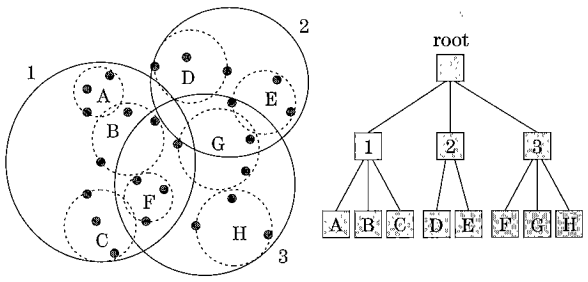


図-4　SS-treeの構造

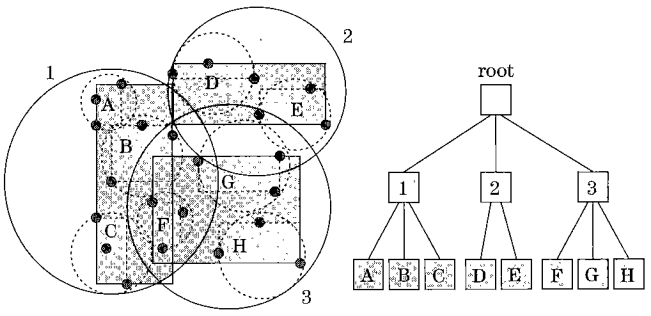


図-5　SR-treeの構造

ルは，次元数が高く，また，データセットの規模もはるかに大きいという特徴があり，従来の手法では効率のよい索引の実現が難しく，1996年ごろから，特徴量ベクトルに特に適した索引技術の提案が活発に行われるようになっている．

特徴量ベクトルを対象とした索引構造の技術的課題としては以下の点が挙げられる．

- 高次元空間での最近接点探索を効率よく行えること．
- スケーラブルであること（大規模なデータセットに対して効率がよいこと）．
- データの追加・削除を効率よくできること．
- 二次記憶を効率的に利用できること．
- 並行処理環境において効率的であること．

最初の課題は，内容に基づく検索の特徴量ベクトルに特徴的な課題である．2番目以降は，大規模アーカイブに対する索引技術としての一般的課題であり，3番目以降は必須ではないが，データベースシステムのレスポンスやスループットを高めるために実現されることが望ましい課題である．

QBICが登場した当時，以上の問題を解決した索引構造はまだ提案されていなかった．この問題に初めて具体的な解を示したのは，Whiteらによって提案されたSS-tree[3]である．また，SR-tree[5]は，SS-treeをさらに改良したものであり，そのソースコードが公開されていることもあって，最近接点探索を高速化する索引の1つとして研究や教育目的で広く使われている．

SS-treeとSR-tree: 最近接点探索のための木構造索引

SS-tree[3]の構造は，部分領域の形状として超直方体の代わりに，部分領域の重心を中心とする超球が使われていること以外，R*-treeと基本的に同様である（図-4）．また，SR-tree[5]は，SS-treeを改良したものであり，部分領域の形状として，超球と超直方体の共通部分を使う点に特徴がある（図-5）．これにより，SS-treeよりも部分領域を小さくすることが可能になり，さらなる高速化が実現されている．SS-treeとSR-treeは，R*-treeと類似した木構造を持っているものの，最大の特徴は，木構造を構築するときのアルゴリズムとして，クラスタリングに似た方法を使っていることである．これにより高次元での最近接点探索をR*-treeよりも高速に実行可能となっている．

一般に木構造索引を使って最近接点探索を行う場合，以下の手順で探索すると，必要最少限のノードを辿るだけで解を見つけられることが知られている[6),7)]．

- 根ノードから出発し，質問点から近い順に部分領域を辿っていく．
- 葉ノードに達するたびに，そこに格納されている点を調べ，それまでに遭遇した点の中で質問点に最も近いものを最近接点の候補とする．
- 候補よりも近い部分領域が残っている限り探索を続け，該当する部分領域がなくなった時点で探索が完了する．
- 最後に残った候補が最近接点探索の解である．

この探索を高速に実行するためには，木構造が次のように構築されていることが求められる．

- 互いに近い点が同じ部分領域に属すこと．
- 部分領域どうしが互いに離れていること．

SS-treeが登場するまでは，最近接点探索のための索引構造としてR*-treeが有力候補であったが，R*-treeは最近接点探索よりも矩形状の範囲探索の高速化を目的として設計されているため，上記の要請のうち，特に

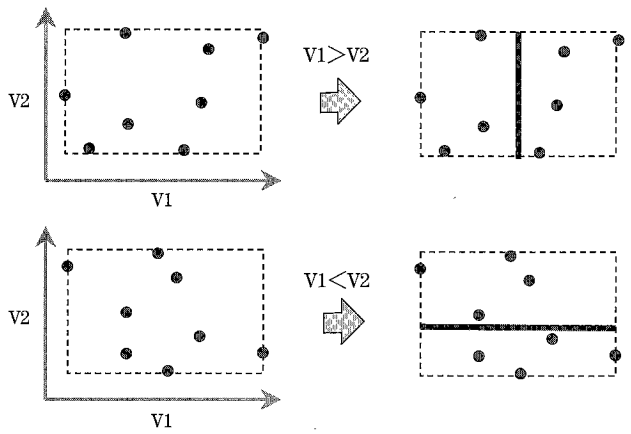


図-6　SS-treeならびにSR-treeの領域分割

最初の要請を満たしていない．たとえば，R*-treeでは，木構造の部分領域を生成するとき，細長い領域を生成することがある．細長い領域は，空間中に占める体積が少ないため，矩形状の範囲探索の高速化には効果が高い．ところが，最近接点探索にとっては，互いに大きく離れたものどうしが1つの部分領域に属してしまうため，探索の高速化につながらないことになる．

一方，SS-treeならびにSR-treeで使われている木構造の構築アルゴリズムは，粗い近似のもとで，部分領域の総分散（各軸の分散の和）を小さくするクラスタリングと見なせることが明らかになっている[8]．木構造の索引にデータを動的に追加していくと，ノードの枝が増え過ぎてしまいノードを分割することが必要になるが，SS-treeならびにSR-treeでは，分割対象となる部分領域の各空間軸についての座標値の分散を求め，最大の分散を持つ軸を分割軸として領域分割を行う（図-6）．この方法は，分割後の部分領域の総分散を小さくする働きがあり，クラスタリングの効果がある．

このような総分散を小さくするアルゴリズムとしては，与えられた点集合をボトムアップに階層的にクラスタリングするWard法がよく知られている．SS-treeならびにSR-treeの大きな特徴は，多次元索引構造に，ある種のクラスタリング手法を導入することによって，高次元空間での最近接点探索を高速化できることを示したことにある．

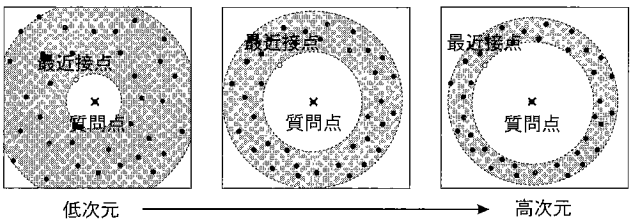


図-7　最近接点探索における次元の呪い

VA-File: 次元の呪いを緩和する索引構造

画像や映像に対する内容に基づく検索では，時には100次元を超えるような高次元の特徴量ベクトルが使われる．このような次元の高い空間では，2次元，3次元といった空間では想像できない現象が起こることが知られている．特に，空間の自由度がきわめて高いために，計算量の問題などから，計算幾何学や多変量解析の問題など種々の問題を解くことが困難になることが知られており，それらは総称して，次元の呪い（Curse of Dimensionality）と呼ばれている．高次元空間における最近接点探索では，次元の呪いとして，次元が高くなるほど，最近接点の探索が困難になるという現象が起こる．たとえば，n次元空間中に点が一様に分布している場合，ある点にk番目に近い点とk+1番目に近い点の距離の比は，次式のようになることが知られている[9]．この式から分かるとおり，k番目に近い点とk+1番目に近い点の距離の比は，nが大きくなるにつれて1に漸近する．

$$\frac{E\{d_{(k+1)NN}\}}{E\{d_{kNN}\}} \approx 1 + \frac{1}{kn}$$

また，点が一様に分布している場合，最も近い点までの距離と最も遠い点までの距離の比が，次元が高くなるにつれて1に漸近することも示されている[10]．この現象を図示すると図-7のようになる．一点を中心として他の点までの距離の分布を見ると，次元が高くなるほど距離の差がみられなくなるのである．

高次元空間において近い点と遠い点とで距離の差がなくなってしまうのは，空間の自由度が指数的に大きくなるためである．たとえば，2次元空間に100万個の点を等間隔に配置することを考えると，軸ごとに1,000分割し，1,000×1,000の位置に点を配置すれば等間隔に配置できる．ところが100次元空間の場合，2次元空間

と同様に軸ごとに1,000分割した場合，$1{,}000^{100}$という莫大な数の点が必要になる．たとえ，軸ごとに2分割しただけでも，空間が$2^{100}\approx10^{30}$個に分割されてしまうため，たかだか100万個の点では，点がきわめて粗に配置されてしまい，距離に差が生じなくなってしまうのである．

このような現象が起こると，最近接点探索の高速化は本質的に難しくなる．なぜならば，距離による差が小さいため，SR-treeのようなクラスタリングを行っても，互いに大きく重なり合ったクラスタばかりができてしまい，クラスタリングの効果が発揮されないからである．そのような状況を緩和するための索引構造として提案されたのがVA-File[11]である．VA-Fileの特徴は，クラスタリングによるグループ分けを断念し，基本的に線形探索を行うことにある．ただし，高次元ベクトルをそのまま線形探索したのではデータの読み込み量が膨らんでしまうため，ベクトルをビット数の少ない近似的な表現に変換することでデータ読み込み量の低減を図る．たとえば，各次元について6ビットで離散量子化すると，100次元でも600ビットで表現することができる．もし100次元のベクトルを倍精度実数で表現すると32ビット×100＝3,200ビット必要なので，データ読み込み量を5分の1以下に低減できることになる．もちろん，近似表現による探索だけでは正解を特定できないため，線形探索によって絞り込まれた結果について本来の精度で比較を行うことで正解が得られる．VA-Fileの構造は非常に単純であるが，この手法の大きな特徴は，点が高次元空間に広く分布している状態ではSR-treeのようなクラスタリングを用いる方法よりも，情報圧縮によるデータ入力量の削減の方が探索の高速化に有利であることを示した点にある．

なお，図-7のような現象は，あくまでも次元の高い空間に点が一様に分布している場合に起こるということに注意する必要がある．たとえば，100次元空間であっても，点が特定の2次元平面の上にだけ分布しているのであれば，点の分布としての次元数は2であるため図-7のような現象はみられない．次元の呪いが現れるかどうかは，空間の次元数ではなく，点の分布の次元数（埋め込み次元数と呼ばれる）によって決まるのである．そのため，100次元の特徴量ベクトルであっても，必ずVA-Fileの方がSR-treeよりも適しているとはいえない．

類似検索のためのさまざまな索引技術

SS-treeの登場以後，現在に至るまで，最近接点探索を高速化する手法としてさまざまな索引構造が提案されている．ユークリッド空間を対象とした索引構造としては，SS-tree，VAMSplit R-tree，SR-treeの他に，X-tree[12]，LSD^h-tree[13]，Hybrid tree[14]などが提案されており，新しい展開としては，木構造索引にVA-Fileで使われている近似表現を導入する手法として，IQ-tree[15]やA-tree[16]が提案されている．また，ユークリッド空間よりも，より一般化されたメトリック空間を対象とする索引構造として，主記憶データ構造であるmetric treeを二次記憶用に改良したM-tree[17]やMVP-tree[18]が提案されている．索引構造の研究と並行して，コストモデルに関する研究[19), 20)]，局所的な領域ごとに次元を縮退させる手法[21)]，確率的近似探索手法など[22)]，さまざまな視点から研究が行われている[23), 24)]．

今後の展望

画像や映像を対象としたデータベースの規模は急速に大きくなりつつあるため，従来の研究よりも，より具体的な課題への取組みが求められている．すなわち，実アプリケーションならではの革新的な技術や，実アプリケーションでこそ効果を発揮する技術等が期待されている．実アプリケーションならではの課題の例としては，ここ数年，特に注目を集めている特徴量空間のデータ分布の問題がある．VA-Fileの節で述べたとおり，高次元空間では，近い点と遠い点との距離に差が生じなくなることがあるが，その一方，実データの場合には，データの分布に偏りがあり，一様分布とはまったく異なる分布を持っている．分布の次元は，空間の次元よりも小さくなっていることが多く，しかも，空間全体で一様ではなく，局所的な部分領域ごとに異なる分布をしていることが多い．たとえば，**図-8**は，36次元のカラーヒストグラムを特徴量とする約60,000枚の写真画像のデータベースの例であるが，データの分布は，一方では距離の差が小さく，他方では大きくなっている．距離の差が小さい場合，検索結果としての示差性も小さいことが多いため，正確に最近接点を求める意味はあまりない．そこで，そのような場合に探索を簡素化することで，探索処理の高速化と，探索結果の示差性の判定が可能になる[25)]．このようなデータの分布特性を最近接点探索に取り入れようとする研究は，ここ数年関心が高まっており[10), 25), 26)]，実アプリケーションへの適用と相まって，重要なテーマになることが予想される．

画像や映像の大規模アーカイブを対象とした類似検索のための索引技術は，まだ比較的新しい分野であり，実アプリケーションへの適用は最近になってようやく

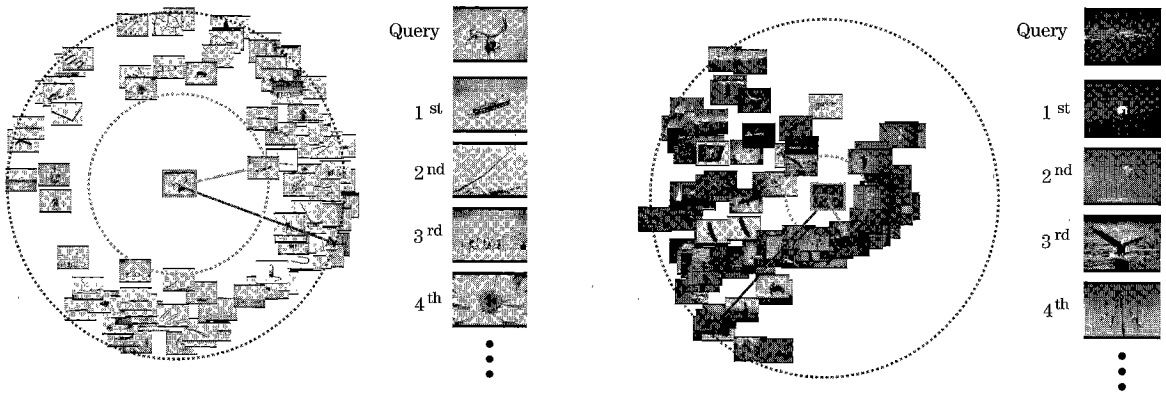


図-8　特徴量空間を局所的にみた場合のデータ分布の多様性

具体性を帯びてきた段階である．特に，テキストを対象とした索引技術に比べ，データベース管理システムへの導入がまだ広がりをみせておらず，今後，実アプリケーションへの適用やデータベース管理システムへの実装が進むにつれて，新しい研究課題が出てくるとともに，技術がさらに発展していくものと考えられる．

参考文献

1) Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D. and Equitz, W.: Efficient and Effective Querying by Image Content, J. of Intelligent Inf. Syst., Vol.3, No.3/4, pp.231-262 (1994).
2) Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D. and Yanker, P.: Query by Image and Video Content: the QBIC System, Computer, Vol.28, No.9, pp.23-32 (1995).
3) White, D. A. and Jain, R.: Similarity Indexing with the SS-tree, Proc. 12th ICDE, pp.516-523 (1996).
4) White, D. A. and Jain, R.: Similarity Indexing: Algorithms and Performance, Proc. SPIE, Vol.2670, pp.62-73 (1996).
5) Katayama, N. and Satoh, S.: The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries, Proc. SIGMOD 1997, pp.369-380 (1997).
6) Hjaltason, G. and Samet, H.: Ranking in Spatial Databases, Proc. SSD'95, LNCS 951, pp.83-95 (1995).
7) Hjaltason, G. and Samet, H.: Distance Browsing in Spatial Databases, ACM Trans. Database Syst., Vol.24, No.2, pp.265-318 (1999).
8) 片山紀生, 佐藤真一: 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, Vol.J82-D-II, No.10, pp.1606-1616 (1999).
9) Fukunaga, K.: Introduction to Statistical Pattern Recognition (2nd ed.), Academic Press (1990).
10) Beyer, K. S., Goldstein, J., Ramakrishnan, R. and Shaft, U.: When Is "Nearest Neighbor" Meaningful?, Proc. 7th ICDT, pp.217-235 (1999).
11) Weber, R., Schek, H.-J. and Blott, S.: A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces, Proc. 24th VLDB, pp.194-205 (1998).
12) Berchtold, S., Keim, D. A. and Kriegel, H.-P.: The X-tree: An Index Structure for High-Dimensional Data, Proc. 22nd VLDB, pp.28-39 (1996).
13) Henrich, A.: The LSD^h-tree: An Access Structure for Feature Vectors, Proc. 14th ICDE, pp.362-369 (1998).
14) Chakrabarti, K. and Mehrotra, S.: The Hybrid Tree: An Index Structure for High Dimensional Feature Spaces, Proc. 15th ICDE, pp.440-447 (1999).
15) Berchtold, S., Böhm, C., Jagadish, H. V., Kriegel, H.-P. and Sander, J.: Independent Quantization: An Index Compression Technique for High-Dimensional Data Spaces, Proc. 16th ICDE, pp.577-588 (2000).
16) Sakurai, Y., Yoshikawa, M., Uemura, S. and Kojima, H.: The A-tree: An Index Structure for High-Dimensional Spaces Using Relative Approximation, Proc. 26th VLDB, pp.516-526 (2000).
17) Ciaccia, P., Patella, M. and Zezula, P.: M-tree: An Efficient Access Method for Similarity Search in Metric Spaces, Proc. 23rd VLDB, pp.426-435 (1997).
18) Bozkaya, T. and Ozsoyoglu, M.: Indexing Large Metric Spaces for Similarity Search Queries, ACM Trans. Database Syst., Vol.24, No.3, pp.361-404 (1999).
19) Berchtold, S., Böhm, C., Keim, D. A. and Kriegel, H.-P.: A Cost Model For Nearest Neighbor Search in High-Dimensional Data Space, Proc. 16th PODS, pp.78-86 (1997).
20) Papadopoulos, A. and Manolopoulos, Y.: Performance of Nearest Neighbor Queries in R-Trees, Proc. 6th ICDT, pp.394-408 (1997).
21) Chakrabarti, K. and Mehrotra, S.: Local Dimensionality Reduction: A New Approach to Indexing High Dimensional Spaces, Proc. 26th VLDB, pp.89-100 (2000).
22) Ciaccia, P. and Patella, M.: PAC Nearest Neighbor Queries: Approximate and Controlled Search in High-Dimensional and Metric Spaces, Proc. 16th ICDE, pp.244-255 (2000).
23) Chávez, E., Navarro, G., Baeza-Yates, R. and Marroquín, J. L.: Proximity Searching in Metric Spaces, To appear in Comput. Surv. (http://www.dcc.uchile.cl/~gnavarro/ps/acmcs01.2.ps.gz).
24) Keim, D. A.: High-Dimensional Index Structures: Database Support for Next Decades Applications, ICDE2000 Tutorial (http://www.informatik.uni-halle.de/~keim/PS/ICDE00.pdf)
25) Katayama, N. and Satoh, S.: Distinctiveness-Sensitive Nearest-Neighbor Search for Efficient Similarity Retrieval of Multimedia Information, Proc. 17th ICDE, pp.493-502 (2001).
26) Hinneburg, A., Aggarwal, C. C. and Keim, D. A.: What is The Nearest Neighbor in High Dimensional Spaces?, Proc. 26th VLDB, pp.506-515 (2000).

(平成13年8月10日受付)