データベースシステム　125−51
（２００１．７．１８）

楕円体問合せのための類似探索手法の提案

櫻井 保志† 吉川 正俊§ 植村 俊亮§ 片岡 良治†

†NTT サイバースペース研究所
神奈川県横須賀市光の丘 1-1
{ysakurai, kataoka}@dq.isl.ntt.co.jp

§ 奈良先端科学技術大学院大学 情報科学研究科
奈良県生駒市高山町 8916 番地の 5
{yosikawa, uemura}@is.aist-nara.ac.jp

あらまし　類似検索メカニズムはユークリッド距離関数のみならず，より一般的な楕円体距離関数を扱える能力を持つことが望ましい．本論文では，空間変換法（STT; Spatial Transformation Technique）と呼ぶ楕円体問合せのための新たな探索手法を提案する．提案手法は空間変換の概念に基づいており，楕円体距離関数に基づく問合せを効率的に支援することができる．これまでにも，多次元索引構造を用いて楕円体問合せを処理する手法が提案されているが，これらの手法は問合せ点と包囲矩形との距離の計算に多くの時間を必要とし，ディスクアクセスコストよりも高い CPU コストが生じる．提案手法の基本的なアイデアは，問合せ点からの距離を楕円体距離関数で計算しなければならないような元の空間に位置する包囲矩形を，ユークリッド距離関数に基づく新たな空間に位置する空間オブジェクトに変換することである．従来手法と比較して，提案手法は空間変換による距離近似によって CPU コストの低減化を達成している．評価実験では様々な問合せ行列を用い，提案手法の有用性を示した．

キーワード　類似探索，高次元データ，楕円体問合せ，空間変換法，空間索引

A Similarity Search Technique for Ellipsoid Queries

Yasushi Sakurai† Masatoshi Yoshikawa§ Shunsuke Uemura§ Ryoji Kataoka†

†NTT Cyber Space Laboratories
1-1 Hikarinooka, Yokosuka, Kanagawa
239-0847 Japan
{ysakurai, kataoka}@dq.isl.ntt.co.jp

§Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-01 Japan
{yosikawa, uemura}@is.aist-nara.ac.jp

Abstract　Similarity retrieval mechanisms should utilize generalized quadratic form distance functions as well as the Euclidean distance function since ellipsoid queries parameters may vary with the user and situation. In this paper, we propose a spatial transformation technique that yields a new search method for adaptive ellipsoid queries. The technique is based on the notion of spatial transformation and efficiently supports adaptive ellipsoid queries with quadratic form distance functions. Although conventional search methods can support ellipsoid queries by using multi-dimensional index structures, these methods incur high CPU-cost for measuring distances between a query point and bounding rectangles with respect to quadratic form distance functions, which exceeds disk access cost on search processing. The basic idea is to transform the bounding rectangles in the original space, wherein distance from a query point is measured by quadratic form distance functions, into spatial objects in a new space wherein distance is measured by Euclidean distance functions. In contrast to the conventional methods, our proposed method significantly reduces CPU-cost due to the distance approximation by the spatial transformation; exact distance evaluations are avoided for most of the accessed bounding rectangles in the index structures. Experiments using various matrices demonstrate the superiority of the proposed method.

Key words　similarity search, high-dimensional data, ellipsoid queries, STT, spatial indices

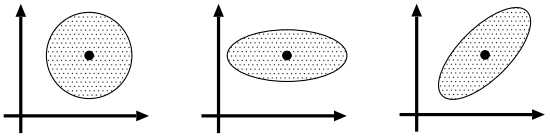
1 まえがき

マルチメディア内容検索システムは，マルチメディアデータから抽出した特徴ベクトルを用い，問合せオブジェクトと類似したデータオブジェクトを検索する．システムに実装されているパターン認識手法の多様化，データベースの大規模化に伴い，これらのシステムでは検索精度と検索性能の両方の向上が必要である．そこで探索手法は (1) より一般的な距離関数に基づいた情報検索，(2) 類似検索処理の高速化，が必要となる．

類似検索メカニズムはユークリッド距離関数のみならず，より一般的な楕円体距離関数を扱える能力を持つことが望ましい．ユークリッド距離空間では全ての次元が互いに独立しているため，利用者の好みを十分に反映することはできない．これに対して，楕円体距離関数は次元間の相関や重みを表現することができ，これらの関数を用いた検索メカニズムは関数決定の自由度が高く，利用者が望むデータオブジェクトを高い精度で検索することができる [HSE+95]．問合せ行列を M，問合せ点を q，そしてデータ集合に含まれる任意の点を p とするとき，楕円体距離は $d^2_M(p,q) = (p-q)\cdot M\cdot(p-q)^t$ のように計算され，M は正定行列であるために $d^2_M(p,q) > 0$ となる．d 次元空間では，ユークリッド距離関数は等距離面が球となる．重みつきユークリッド距離関数の等距離面は楕円体であり，その主軸は座標軸に沿ったものである．楕円体距離関数の等距離面は楕円体であり，図 1 のようにその主軸は任意の方向を向いている [ISF98]．すなわち楕円体距離関数は，ユークリッド距離関数や重みつきユークリッド距離関数を一般化したものと見なすことができ，MindReader[ISF98] は楕円体距離関数を用いた応用例の一つである．楕円体距離関数は，ユークリッド距離関数よりも利用者の意思をより正確に表現することができる．

一方，マルチメディアデータベースのサイズやデータの次元数が増えたとき，高速に検索するための索引手法が必要となる．索引手法には様々なものが提案されており [GG98]，例えば，R*-tree[BKSS90] や A-tree[SYUK00] を挙げることができる．特に A-tree は高次元データにおいて高い性能を示す [SYUK00]．しかし，これらの索引手法はユークリッド距離関数に基づく探索にのみ焦点をあてており，したがって楕円体問合せのための新たな探索手法が必要である．

本研究の目的は，利用者適応の楕円体問合せのための類似探索技術を確立することである．利用者適応の楕円体問合せとは，毎回もしくは利用者毎に，異なる重みや相関を反映した問合せのことを指す．



(a) Euclidean (b) weighted Euclidean (c) ellipsoid

図 1: 距離関数の等距離面

本論文では，利用者適応の楕円体問合せを効率的に処理する空間変換法（STT; Spatial Transformation Technique）[SYKU01] を提案する．STT の基本的なアイデアは，問合せ点からの距離を楕円体距離関数で計算しなければならないような元の空間に位置する包囲矩形を，ユークリッド距離関数に基づく新たな空間に位置する空間オブジェクトに変換することである．STT は低い CPU コストしか必要としないにもかかわらず，扁平な問合せ行列や高次元であっても高い近似精度を実現している．

2 関連研究

任意の距離関数のための索引手法として，M-tree [CPZ97] や mvp-tree [BO97] のような距離索引があるが，これらの索引は，毎回もしくは利用者毎に変動する距離関数を扱うことはできない．文献 [SK97] では，利用者適応の楕円体問合せを処理するための手法を提案している．この手法は，索引構造を用い，包囲矩形と問合せ点との厳密な距離を最急降下法に基づいて計算し，探索処理を行っている．しかし，d を次元数，ω を最急降下法の繰り返し回数とするとき，問合せ点と包囲矩形の距離計算に $O(\omega\cdot d^2)$ 時間を必要とし，これは検索処理におけるディスクアクセスに要する時間を越える．文献 [ABKS98] では，Ankerst らが CPU コストを削減するために，MBB（Minimum Bounding Box）距離関数と MBS（Minimum Bounding Sphere）距離関数を用いて厳密な距離計算回数を削減している．以下は，MBB 距離と MBS 距離の定義式である．

$$\begin{aligned} d^2_{MBB(M)}(p,q) &= \max_{i=1}^{d}\left(\frac{(p_i-q_i)^2}{(M^{-1})_{ii}}\right), \\ d^2_{MBS(M)}(p,q) &= \lambda^2_{M_{min}}\cdot(p-q)^2. \end{aligned} \qquad (1)$$

ここで，λ_{M_i} $(i=1,\cdots,d)$ は，M の固有値であり，$\lambda_{M_{min}}$ は M の最小の固有値である．MBB 距離関数は矩形を用いて楕円体問合せの領域を包囲，近似する．MBS 距離関数は球を用いて近似する．二つの関数は，距離計算に $O(d)$ 時間しか必要としない．

文献 [ABKS98] の探索アルゴリズムでは，MBB と MBS 距離関数を用いて CPU コストを低減化させており，本論文ではこの手法を MBB-MBS 近似法と呼ぶ．しかしながら MBB-MBS 近似法は，次元数が増加したり，もしくは問合せ行列によって形作られる楕円体が扁平になるにしたがい，近似精度が低下する．そして，低い近似精度は CPU 時間の増加につながる．

3 空間変換法

本節では，効率的に楕円体問合せを処理することを目的とした空間変換法（STT; Spatial Transformation Technique）[SYKU01] を提案する．STT は問合せ点と包囲矩形のと間の楕円体距離を近似する手法であり，従来手法である MBB-MBS 近似法と同様にフォルスドロップを生み出さないことを保証する．すなわち，どのような問合せに対しても正確な検索結果を提示する．

3.1 STT の概要

厳密な楕円体距離の計算において，問合せ点と索引構造に含まれる包囲矩形との距離の計算には多くの CPU コストを必要とする．ω を最急降下法の繰り返し回数であるとすると，計算時間は $O(\omega \cdot d^2)$ である．STT の基本的なアイデアは，問合せ点からの距離を楕円体距離関数で計算しなければならないような元の空間に位置する包囲矩形を，ユークリッド距離関数に基づく新たな空間に位置する空間オブジェクトに変換することである．STT は，距離計算に関して繰り返しを必要とせず，包囲矩形を空間変換することによって低い CPU コストにもかかわらず優れた近似精度を示す．本節では，まず空間変換の定義について述べ，そして索引構造に含まれる包囲矩形のための空間変換法について論ずる．

3.2 空間変換

問合せ行列 M，問合せ点 q があたえられているとき，d 次元空間 $\mathcal{S}$ における任意の点 p と q までの楕円体距離は，以下の式によって得ることができる．

$$d_M^2(p,q) = (p-q) \cdot M \cdot (p-q)^t. \tag{2}$$

問合せ行列 M は正値対称行列であるため，以下のようにスペクトル分解が可能である．

$$M = E_M \cdot \Lambda_M \cdot E_M^t. \tag{3}$$

ここで，E_M は M の固有ベクトル，Λ_M は d 個の M の固有値 λ_{M_i} $(i = 1, 2, \cdots d)$ を対角成分とする対角行列である．式 (3) を用いて，式 (2) を以下のように変形することができる．

$$d_M^2(p,q) = (p-q) \cdot E_M \cdot \Lambda_M \cdot E_M^t \cdot (p-q)^t. \tag{4}$$

ここで，ユークリッド空間 $\mathcal{S}'$ 内の点 $p' = (p-q) \cdot E_M \cdot \Lambda_M^{\frac{1}{2}}$ を考える．式 (4) より，ユークリッド空間 $\mathcal{S}'$ における原点 O と p' とのユークリッド距離は，$\mathcal{S}$ における楕円体距離 $d_M^2(p,q)$ に等しい．すなわち，$d_M^2(p,q) = p' \cdot p'^t$．ここで，$M$ の変換行列を以下のように定義する．

$$A_M = E_M \cdot \Lambda_M^{\frac{1}{2}}. \tag{5}$$

変換行列 A_M を用いることにより，$\mathcal{S}$ 内の楕円体距離関数による計算を，$\mathcal{S}'$ 内のユークリッド距離関数による計算に置き換えることができる．これを，p の p' への空間変換と呼ぶ．

3.3 矩形の空間変換

STT では，索引構造に含まれる包囲矩形を空間変換する．図 2 は矩形の空間変換の例を示している．図において $\mathcal{S}$ 内の包囲矩形 P は，$\mathcal{S}'$ 内の d 次元平行四辺形 P' に変換される．高次元空間における多角形と原点 O との距離計算は多くの計算時間を必要とする．そこで，図 2(b) のように P' を矩形 R で近似する．この近似により，少ない計算時間でユークリッド距離を得ることができる．

空間 $\mathcal{S}$ 内の矩形 P，問合せ点を q を考える．P において対角する頂点を p_a と p_b とし，P の i 次元の辺長を l_i とする．このとき，p_a の空間変換によって得られる $\mathcal{S}'$ 内の点 p'_a の位置は

$$p'_a = (p_a - q) \cdot A_M. \tag{6}$$

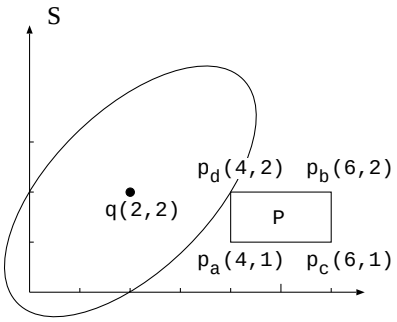
である．変換行列 A_M の要素 a_{ij} から，以下のような二種類の成分を抽出する．

$$\phi_{ij} = \begin{cases} a_{ij} & (a_{ij} < 0) \\ 0 & (\text{otherwise}), \end{cases} \qquad \psi_{ij} = \begin{cases} a_{ij} & (a_{ij} > 0) \\ 0 & (\text{otherwise}). \end{cases} \tag{7}$$

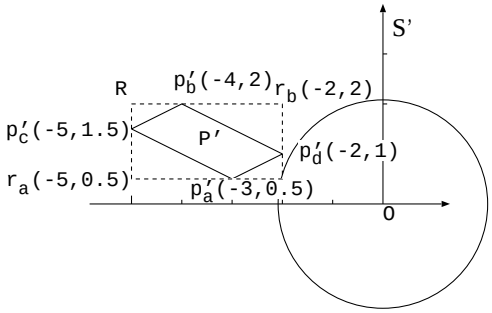
式 (6) (7) から，矩形 P の空間変換である平行四辺形を包含する矩形 R は，下式により得られる[1]．

$$R = (r_a, r_b), \tag{8}$$

[1] 式 (8) の証明は [SYKU01] 参照.



(a) A rectangle in the original space



(b) A rectangle calculated by STT

図 2: 空間変換の例

$$
\begin{aligned}
r_{a_j} &= p'_{a_j} + \sum_{i=1}^{d} l_i \cdot \phi_{ij}, \\
r_{b_j} &= p'_{a_j} + \sum_{i=1}^{d} l_i \cdot \psi_{ij} \quad (1 \le j \le d).
\end{aligned}
$$

ここで，r_a と r_b は R において対角する頂点を表している．空間 $\mathcal{S}'$ において，R は P' を完全に包含しているため，P と q の楕円体距離 $d_M^2(P,q)$ の計算を，R と O のユークリッド距離 $d^2(R,O)$ の計算に置き換えることができる．すなわち，$d^2(R,O) \le d_M^2(P,q)$ である．

例えば，図 2 に示すように，問合せ点 $q=(2,2)$ と行列

$$
M = \begin{pmatrix} 1.25 & -0.75 \\ -0.75 & 1.25 \end{pmatrix}
$$

が与えられているとする．$\mathcal{S}$ 内の包囲矩形 P の頂点 p_a，p_b，p_c，p_d は，M を用いることによって $\mathcal{S}'$ 内の平行四辺形 P' の頂点 p'_a，p'_b，p'_c，p'_d に変換される．また，$R=(r_a,r_b)$ は P' を包含する．$d_M^2(q,P)$ は $d^2(R,O)$ によって近似されるため，$d_M^2(q,P)$ の代わりに $d^2(R,O)$ を探索に用いることができる．

3.4 探索アルゴリズム

主要な多次元データの問合せとして，範囲問合せ，k 近傍問合せがあるが，空間変換に基づく探索アルゴリズムは両方の問合せを効率良く処理することができる．範囲問合せよりも k 近傍問合せの方が複雑であり，より多くの探索コストが必要であるため，我々は k 近傍問合せに焦点をあて，アルゴリズムを説明する．範囲問合せについても，本節で説明する考え方をあてはめることができる．

我々は，3.2 節および 3.3 節において空間変換のための式 (5) (6) (7) (8) を示した．しかし，アクセスした全ての矩形について，問合せ点との距離をこれらの式に基づいて忠実に計算することは無駄が多い．そこで，CPU 時間を削減するために二つのアイデアを導入する．

第 1 に，式 (5) (7) の計算結果は，アクセスした矩形の位置に依存しない．したがって，これらの計算を探索処理の最初に実行することにより，その計算結果はアクセスされる矩形全ての空間変換にあてはめることができる．

第 2 に，式 (8) に関する計算時間の削減である．式 (7) において，平均して ϕ_{ij}，ψ_{ij} の値の半分は 0 であることに注意して欲しい．したがって，実装上においては，$\phi_{ij} \neq 0, \psi_{ij} \neq 0$ となる行番号 i と列番号 j の全てのペアを，ノードにアクセスする前に調査することが望ましい．この調査を探索処理の最初に実行することにより，式 (8) における R の計算時間，すなわち r_{a_j} と r_{b_j} の計算時間を半分に抑制することができる．

例えば，行列 ϕ_{ij} の第 j 列において，$\phi_{ij} \neq 0$ となる成分の数を c_{a_j} とする．そして第 j 列における c_{a_j} 個の成分各々の行番号を u_{jk} $(k=1,\ldots,c_{a_j})$ とする．同様に行列 ψ_{ij} の第 j 列において，$\psi_{ij} \neq 0$ となる成分の数を c_{b_j} とし，第 j 列における c_{b_j} 個の成分各々の行番号を v_{jk} $(k=1,\ldots,c_{b_j})$ とする．このとき，式 (8) は以下のように変形することにより計算時間の短縮が可能である．

$$
\begin{aligned}
R &= (r_a, r_b), \qquad (9) \\
r_{a_j} &= p'_{a_j} + \sum_{k=1}^{c_{a_j}} l_k \cdot \phi_{(u_{jk})j}, \\
r_{b_j} &= p'_{a_j} + \sum_{k=1}^{c_{b_j}} l_k \cdot \psi_{(v_{jk})j} \quad (1 \le j \le d).
\end{aligned}
$$

ここで，式 (9) における c_{a_j} と c_{b_j} は，平均して $d/2$ である．

図 3 は，R-tree ファミリーの索引構造を用いた楕

```
Procedure search(point query, matrix M,
                                    integer k)
 1. ΦM := analyzeMatrix(M);
 2. enqueue(a_pointer_to_the_root, 0);
 3. while emptyQueue() = false do
 4.   N := dequeue();
 5.   if N is a data node then
 6.     for each entry ∈ N do
 7.       if dMBB-MBS(M)(query, entry.vector)
                          ≤ nnlist[k].dist then
 8.         if dM(query, entry.vector)
                          ≤ nnlist[k].dist then
 9.           nnlist[k].id := entry.id;
10.           nnlist[k].dist := dM(query,
                                  entry.vector);
11.           sort nnlist by distance;
12.           pruneQueue(nnlist[k].dist);
13.         endif
14.   else
15.     for each entry ∈ N do
16.       if dMBB-MBS(M)(query,
          entry.rectangle) ≤ nnlist[k].dist then
17.         R := spatialTransformation(query,
                        entry.rectangle, ΦM);
18.         if d(R, O ) ≤ nnlist[k].dist then
19.           if dM(query, entry.rectangle)
                          ≤ nnlist[k].dist then
20.             enqueue(entry.ptr,
                 dM (query, entry.rectangle));
21.       endif
22.   endif
23. enddo
24. output(nnlist);
```

図 3: 楕円体問合せのための k 近傍探索アルゴリズム

円体問合せのための探索アルゴリズムである．探索アルゴリズムは包囲矩形と問合せ点との距離を評価する場合に空間変換を用いる．厳密な楕円体距離計算よりも，MBB-MBS 近似法や STT の方が距離計算に必要とする時間が少ない．そこで，包囲矩形までの距離を評価する時，探索アルゴリズムは最初に問合せ点までの距離を近似関数によって計算する．もし近似距離が，問合せ点と現在の k 番目の最近傍との距離以下であれば，厳密な楕円体距離関数に基づいて矩形から問合せ点までの距離を評価する．

プロシージャ search では，最初に問合せ行列 M に関する変換行列 A_M の計算，および A_M の各成分の調査を実施する（ステップ 1）．そして優先キューに根ノードへのポインタ，および距離 0 を設定する（ステップ 2）．ステップ 4 では，問合せ点 $query$ から最も近いノード N を優先キューから取り出す．N がデータノードであるとき，データオブジェクトの MBB-MBS 近似距離を計算し，評価する．近似距離が k 近傍距離以下であれば，問合せ点とデータオブジェクトの厳密な距離を計算する（ステップ 5 から 8）．そして，エントリを最近傍リストへ格納し，優先キューのフィルタリングを行う（ステップ 9 から 12）．N がデータノード以外のとき，包囲矩形の MBB-MBS 近似距離を計算し（ステップ 16），MBB-MBS 近似距離が k 近傍距離以下であれば，包囲矩形の空間変換を Φ_M に基づいて計算する（ステップ 17）．ステップ 18 において空間変換によって得られた矩形 R と原点 O とのユークリッド距離を計算する．もし空間変換によって求められた距離が現在の k 近傍距離以下であれば，厳密な距離計算を行う（ステップ 19）．

後述の評価実験では，索引構造として A-tree を用いている．A-tree はユークリッド距離に基づく問合せのみならず，楕円体問合せでも有用である．A-tree の探索アルゴリズムは，他の R-tree ファミリーに属する手法のアルゴリズムとは一部異なる．A-tree の探索アルゴリズムの詳細は [SYUK00] に記載されている．

3.5 次元縮退

問合せ行列によって形作られる楕円体が扁平になると，小さい固有値の固有ベクトルが存在することになる．すなわち，空間変換によって生成される空間において，その固有ベクトルが示す次元は，他の次元と同じだけの計算時間を要するにもかかわらず，近似精度に寄与する度合いが低いことになる．STT における次元縮退では，近似精度に貢献しないような小さい固有値の次元を省略することによって，計算時間の短縮を図る．

空間変換によって作られる空間 $\mathcal{S}'$ において，原点 O に最も近接する矩形 R の頂点を $r = (r_1, r_2, \ldots, r_d)$ とする．次元縮退を実施するとき，R と O の距離は以下のようになる．

$$\tilde{d}^2(R,O) = \sum_{i=1}^{n}(r_i)^2, \quad (10)$$

$$n = COUNT\left(\lambda_j \geq \frac{\eta}{d}\cdot\sum_{i=1}^{d}\lambda_i\right) \quad (j = 1,\ldots,d).$$

ここで，η は次元縮退のためのしきい値である．また，固有値 λ_i は昇順になっているものとする．すな

わち，$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d > 0$．関数 $COUNT(\Gamma)$ は，Γ の条件を満たす要素の数を表している．式 (10) は，距離計算を行うときに 1 から n までの限られた次元しか使用しないことを示している（$n \leq d$）．したがって，次元縮退によって式 (10) の計算のみならず，式 (6) (7) (9) の計算時間も n/d に短縮される．

4 性能評価

STT の有効性を確認するため，アルゴリズムを実装し，従来手法である MBB-MBS 近似法と比較した．

4.1 実験条件

手法の性能を計測するための実データとして，画像から抽出したカラーヒストグラムによる特徴ベクトルを用いる．次元数は 8 と 27，サイズは 100,000 件である．探索性能の評価では，ページアクセス数と CPU 時間を問合せ数 100 の平均によって求めた．最近傍探索の探索数は 20 であり，問合せには索引に含まれているデータとは異なるデータを用いている．ページサイズは 8KB，CPU 時間は SUN UltraSPARC-II 450MHz によって計測した．索引は，高次元空間において優れた性能を示す A-tree[SYUK00] を用いる．A-tree における近似のための符号長は次元あたり 6 ビットである．

問合せ行列 M について，M の要素 m_{ij} を下式を用いて求める [HSE+95][ABKS98].

$$m_{ij} = exp(-\alpha(d_w(c_i, c_j)/d_{max})^2).$$

ここで，α は正の定数であり，$d_w(c_i, c_j)$ は色 c_i と c_j の間の重みつきユークリッド距離を表している．距離 d_w の要素 $w = (w_r, w_g, w_b)$ は，RGB 色空間における赤，緑，青の各成分の重みを示している[2]．評価では，α を 10 とし，w_g と w_b は 1 に固定した．w_r は 1 から 1,000 まで変化させた．我々は，8 と 27 次元の全ての行列の固有値を計算した．その固有値の分散 σ_M^2 は，以下のように計算される．

$$\sigma_M^2 = \sum_{i=1}^{d} (\lambda_{M_i} - \overline{\lambda}_M)^2, \quad \overline{\lambda}_M = \sum_{j=0}^{d} \frac{\lambda_{M_j}}{d}.$$

ここで，$det(M) = 1$ であり，λ_{M_i} は i 番目の次元の固有値を表す．また，$\overline{\lambda}_M$ は M の固有値の平均である．

表 1 は，w_r を変化させた場合の σ_M^2 を示している．σ_M^2 を計算する前に，全ての行列は $det(M) = 1$ に正規化されている．行列 M の正規化行列 N については以下のように計算する．

$$N = E_M \cdot \Lambda_N \cdot E_M^t, \quad \lambda_{N_i} = \lambda_{M_i} \cdot \left(\prod_{i=1}^{d} \lambda_{M_i} \right)^{-\frac{1}{d}}$$

ここで，Λ_N は，正規化された固有値 λ_{N_i} を対角成分とする対角行列である．

本論文では，分散 σ_M^2 を M の扁平度と呼ぶ．ここで，単位行列の扁平度は 0 であり，問合せ行列が単位行列であることはユークリッド空間における探索を意味する．α，w_g，w_b が固定されている場合，表 1 が示すように，w_r が高いほど問合せ行列は扁平である．

STT の次元縮退に関して，$\eta = 0.1$，$\eta = 0.01$，$\eta = 0.001$ の中から最も望ましい性能を示す $\eta = 0.01$ を選択した．STT が用いた次元数を表 2 に示す．

表 1: 固有値の分散

w_r		1	10	100	1000
σ_M^2	$d = 8$	0.0307	76.489	7998.6	800214
	$d = 27$	64.777	93372	9.29e8	9.29e12

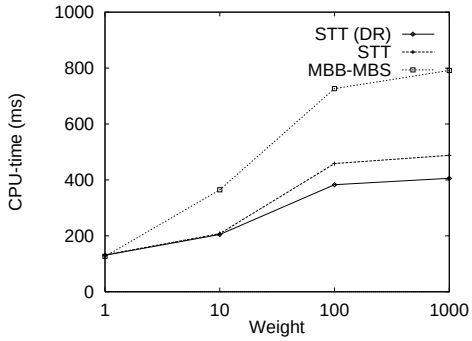
表 2: 楕円体問合せにおいて用いた次元数

w_r		1	10	100	1000
n	$d = 8$	8	8	4	4
	$d = 27$	27	18	9	9

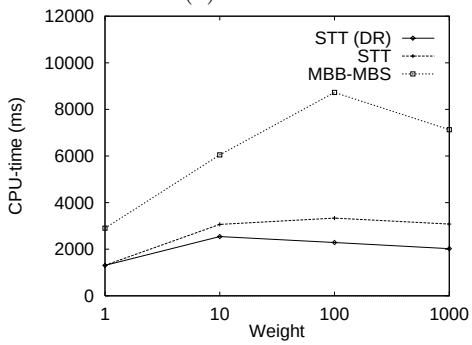
4.2 探索性能

図 4 は，CPU 時間に関する STT と MBB-MBS 近似法の比較を示している．`STT(DR)` は STT において次元縮退のテクニックを用いた場合の CPU 時間を示している．ページアクセス数については，図 5 に示す．STT と MBB-MBS 近似法ともに厳密な楕円体距離関数を用いるため，両手法は同じページアクセス数を要する．したがって，探索性能の差は CPU 時間によって決定される．楕円体問合せは問合せ点と包囲矩形との距離の計算に多くの時間を必要とする．図 4 は，STT が全てのデータ集合において CPU 時間を低減化させていることを示している．特に STT は次元もしくは問合せ行列の扁平度が高くなるほど有効であり，最大で 74%の CPU 時間を削減している．

[2] 詳細については文献 [HSE+95] 参照.

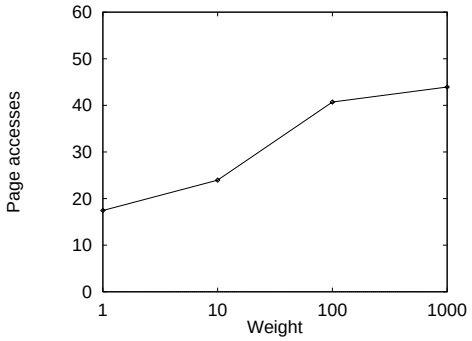


(a) d = 8

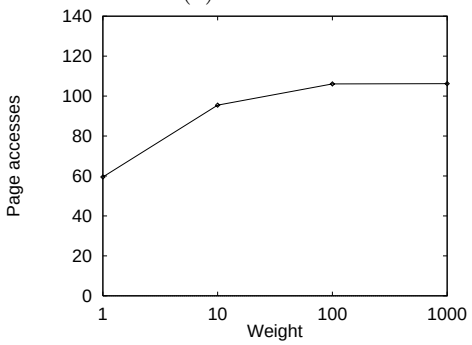


(b) d = 27

図 4: CPU コストに関する STT と MBB-MBS 近似法の比較



(a) d = 8



(b) d = 27

図 5: ページアクセス数

4.3 楕円体問合せにおける近似手法の分析

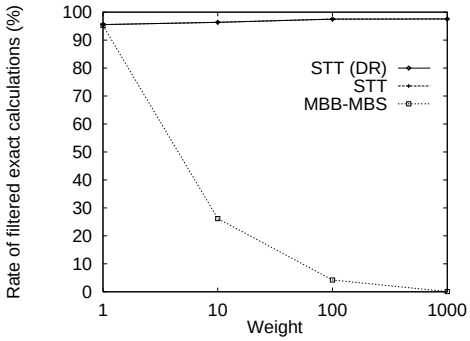
問合せ点と包囲矩形との間の近似距離が k 近傍距離を越える場合，STT は厳密な楕円体距離関数を使用しない．これは MBB-MBS 近似法も同様である．図 6 は，探索処理において，アクセスした包囲矩形に対する厳密な距離計算を回避した割合を示している．すなわち，図は近似手法の効率性を表している．

扁平度が高くなるにしたがい，MBB-MBS 近似法のフィルタリングの効果は減少する．しかし，STT はどのような次元や扁平度の問合せでも，厳密な距離計算を効率良くフィルタリングすることができる．このことから，図 4 に示すように，STT は少ない CPU 時間を達成することができる．

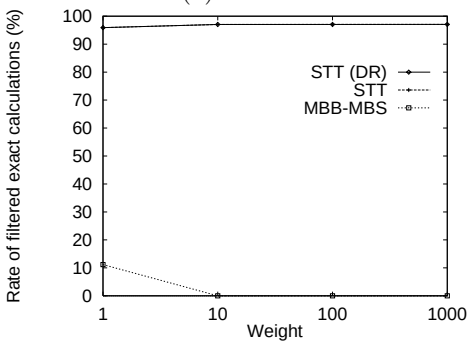
STT の次元縮退は，距離の近似にわずかしか貢献していない次元を無視する．このアイデアは問合せ行列の扁平度が高くなると有効である．表 1 および表 2 において示されているように，$w_r = 1$ の問合せ行列は扁平度が低く，全ての次元数が探索に用いられる．これに対して，$w_r = 100$，$w_r = 1000$ の問合せ行列は扁平度が高く，少ない次元しか用いない．図 6 に示されているように，次元縮退を用いた STT は，次元縮退を用いない STT と同じ近似性能を示しており，優れた特質を持っている．この結果，図 4 に示すように，次元縮退を用いた STT は CPU 時間をさらに低減化させている．

5 むすび

本論文では，楕円体問合せに優れた性能を示す空間変換法（STT; Spatial Transformation Technique）について述べた．STT は空間変換の概念を導入している．空間変換によって包囲矩形と問合せ点との距離は精度良く近似されるため，アクセスした包囲矩形の大部分について STT は厳密な距離計算を省略することができる．MBB-MBS 近似法は次元数や扁平度が高い場合に有効ではなかった．これに対して STT は，高次元もしくは問合せ行列が扁平であっても，その優れた近似精度によって効率的な探索が可能である．また，STT はフォルスドロップを生み出さないことを保証する．様々な問合せ行列やデータ集合を用いた実験では，全ての条件において STT は優れた性能を示している．



(a) d = 8



(b) d = 27

図 6: STT において厳密な距離計算を回避した割合

参考文献

[ABKS98] Mihael Ankerst, Bernhard Braunmüller, Hans-Peter Kriegel, and Thomas Seidl: "Improving Adaptable Similarity Query Processing by Using Approximations", in *Proc. of the 24th International Conference on Very Large Data Bases (VLDB)*, pp. 206–217, New York City, NY, August 1998.

[BKSS90] N. Beckmann, H. P. Kriegel, R. Schneider, and B. Seeger: "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", in *Proc. ACM SIGMOD Conf.*, pp. 322–331, Atlantic City, NJ, May 1990.

[BO97] Tolga Bozkaya and Meral Ozoyoglu: "Distance-Based Indexing for High-Dimensional Metric Spaces", in *Proc. ACM SIGMOD International Conference on Management of Data*, pp. 357–368, May 1997.

[CPZ97] Paolo Ciaccia, Marco Patella, and Pavel Zezula: "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces", in *Proc. of the 23rd International Conference on Very Large Data Bases (VLDB)*, pp. 426—435, Athens, August 1997.

[GG98] Volker Gaede and Oliver Günther: "Multidimensional Access Methods", *ACM Computing Surveys*, Vol. 30, No. 2, pp. 170–231, June 1998.

[HSE+95] James L. Hafner, Harpreet S. Sawhney, William Equitz, Myron Flickner, and Wayne Niblack: "Efficient Color Histogram Indexing for Quadratic Form Distance Functions", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 7, pp. 729–736, July 1995.

[ISF98] Yoshiharu Ishikawa, Ravishankar Subramanya, and Christos Faloutsos: "MindReader: Querying databases through multiple examples", in *Proc. of the 24th International Conference on Very Large Data Bases (VLDB)*, pp. 218–227, New York City, NY, August 1998.

[SK97] Thomas Seidl and Hans-Peter Kriegel: "Efficient User-Adaptable Similarity Search in Large Multimedia Databases", in *Proc. of the 23rd International Conference on Very Large Data Bases (VLDB)*, pp. 506—515, Athens, August 1997.

[SYKU01] Yasushi Sakurai, Masatoshi Yoshikawa, Ryoji Kataoka, and Shunsuke Uemura: "Similarity Search for Adaptive Ellipsoid Queries Using Spatial Transformation", in *Proc. of the 27th International Conference on Very Large Data Bases (VLDB)*, Roma, Italy, September 2001, (to appear).

[SYUK00] Yasushi Sakurai, Masatoshi Yoshikawa, Shunsuke Uemura, and Haruhiko Kojima: "The A-tree: An Index Structure for High-Dimensional Spaces Using Relative Approximation", in *Proc. of the 26th International Conference on Very Large Data Bases (VLDB)*, pp. 516–526, Cairo, Egypt, September 2000.