RW-Tree：非軸直交一次元範囲検索のための索引手法

能登谷淳一† 田村 壮†† 草苅 良至† 笠井 雅夫†

† 秋田県立大学 システム科学技術学部 〒 015–0055 秋田県本荘市土谷字海老ノ口 84–4
†† 秋田県立大学 システム科学技術研究科 〒 015–0055 秋田県本荘市土谷字海老ノ口 84–4
E-mail: †{notoya,m05b005,kusakari,kasai}@akita-pu.ac.jp

あらまし 多次元空間中の非軸直交範囲検索は計算幾何学の分野で近年注目を集めている問題の一つである．本研究では，非軸直交一次元範囲検索に対する効率的な索引構造である RW 木とその生成手法を提案する．我々の提案する RW 木は，軸直交範囲検索に広く利用されている R 木とデータ空間の回転変換に基づく索引である．RW 木は問い合わせ集合が従う分布に関する情報を利用し，予測される問い合わせに適した索引構造を構成する．本研究で提案する RW 木の生成手法は R 木の動的生成手法の一つである R* 木生成手法の拡張であるため，提案手法の利用によりデータベースへのデータの追加，削除等に対応する動的索引の生成が可能である．

キーワード 多次元インデックス，R 木，非軸直交範囲検索

RW-Tree：An Efficient Access Method for Nonisothetic 1D Range Searching

Junichi NOTOYA†, Takeshi TAMURA††, Yoshiyuki KUSAKARI†, and Masao KASAI†

† Faculty of Systems Science and Technology, Akita Prefectural University
Ebinokuchi 84–4, Tsuchiya, Honjyo, Akita 015–0055 Japan
†† Graduate School of Systems Science and Technology, Akita Prefectural University
Ebinokuchi 84–4, Tsuchiya, Honjyo, Akita 015–0055 Japan
E-mail: †{notoya,m05b005,kusakari,kasai}@akita-pu.ac.jp

Abstract Nonisothetic range searching in d-dimensional data space attracts attention from the computational geometry community in recent years. In this paper, we propose the RW-tree which is an efficient access method for nonisothetic 1D range searching. By combining with rotation of data space, we improved the R-tree and developed the RW-tree. The RW-trees acquire structures which are suitable for the distribution of queries. The RW-tree generation algorithm is an extension of the R*-tree which is a dynamic generation technique of the R-tree. With our generation algorithm, RW-trees deal properly with insertions and deletions of data in databases.

Key words Multidimensional Index, R-Tree, Nonisothetic Range Searching

1. はじめに

データベースシステムの応用分野の拡大に伴い，従来の応用における典型的な検索条件による検索にとどまらず，様々な種類の検索処理を効率的に解決することが求められている．特に範囲検索は広範な応用分野で利用される検索の種類であり，多種の範囲検索を効率的に解決するために多くの索引構造やその利用法がこれまでの研究によって提案されてきた [1] .

関係データベースシステムにおいては，ある関係の単一の属性の値の範囲による検索や，複数の属性の範囲の組による検索が特に重要であったため，それらの検索を高速に行うため多くの索引機構が提案された．これらの索引機構が主たる対象とする検索は，d 次元データ空間中の軸に直交する複数の超平面によって境界が与えられる問い合わせ領域による領域探索である．このような領域探索は範囲検索の一種であり，軸直交範囲検索（isothetic range searching）と呼ばれる．

これに対し，近年普及が進んでいる空間データベースシステムなどの応用分野においては，任意の形状を持つ領域内のオブジェクトの検索など，より一般的な範囲検索の重要性が増大している [2]．特に，d 次元空間中の軸に直交しない複数の超平面によって境界が与えられる問い合わせ領域による領域探索は，非軸直交範囲検索（nonisothetic range searching）と呼ばれ，計算幾何学の分野で注目されている問題の一つである [3], [4].

本稿では，非軸直交範囲検索の中でも特に応用の広い検索の種類である非軸直交一次元範囲検索に対する効率的な索引構造を与える．本研究で着目する非軸直交一次元範囲検索とは，

データベース中のデータのうち，$C_1 \leqq \sum_{i=0}^{d-1} a_i x_i \leqq C_2$ の形式の一次結合式で与えられた制約条件をみたすデータを全て求める問題であり，d 次元データ空間中の二枚の平行超平面で挟まれる部分空間を問い合わせ領域とする領域探索と等価である．

我々は従来より軸直交範囲検索に広く利用されている R 木 [5] に着目し，R 木の改良として非軸直交一次元範囲検索用の索引構造を与える．これにより，R 木の実現に関する既存の研究成果の多くの部分を非軸直交一次元範囲検索にも適用可能であると考える．

本稿の以降の部分は以下のように構成する．第 2 節では本研究に関連する過去の研究を紹介する．第 3 節では本研究で扱う問題の定義を与える．第 4 節では既存の索引構造である R 木を非軸直交範囲検索に適用する手法について述べる．第 5 節では非軸直交範囲検索に適した R 木生成のための重み付き生成戦略を提案する．第 6 節では重み付き生成戦略を利用した非軸直交範囲検索用索引構造である RW 木を提案する．第 7 節では RW 木の実験的性能評価の結果を示す．第 8 節は本稿のむすびである．

2. 関 連 研 究

2.1 データベース分野における成果

データベースの分野における過去の研究によって，軸直交範囲検索のための多くの索引構造が提案された．軸直交一次元範囲検索のための木構造索引としては，B 木，B^+ 木 [6] およびそれらの改良型の索引構造が提案され，データベースシステム実装の際に利用される主要な索引構造として採用されている．非軸直交一次元範囲検索において，検索時に与えられる可能性のある全ての問い合わせ領域の法線ベクトルが常に同一である場合には，あらかじめ全データに対して一次結合式の値を計算しておき，その値によって索引付けを行うことにより，これらの一次元索引構造を利用可能である．しかし，現実の応用においては，互いに異なる法線ベクトルを持つ複数の問い合わせ領域が与えられる可能性が存在する．そのため，非軸直交一次元範囲検索にこれらの索引を利用することは現実的ではない．

多次元の軸直交範囲検索に利用可能である索引として，グリッドファイルや，k-d-B 木，R 木など多くの索引構造が空間データベースシステムに関する研究の成果として提案されている [1]．これらの索引の多くは，軸直交範囲検索および k 近傍探索のための索引構造として知られているが，実際には任意の形状の問い合わせ領域に対する領域探索に利用可能である．従って，これらの索引を非軸直交一次元範囲検索に対して適用することが可能である．しかし，これらの索引は本来軸直交範囲検索などに利用することが想定されているため，非軸直交範囲検索に対しては十分な性能を発揮できないと考えられる．本稿では，R 木を非軸直交範囲検索に適用した場合の性能について，提案手法の性能との比較のために議論する．

2.2 計算幾何学分野における成果

非軸直交範囲検索は計算幾何学の分野において近年特に着目されている問題の一つである．特に非軸直交範囲検索の一種である単体領域探索に関する研究がこれまで多く行われてきた [3], [4]．単体領域探索のための索引構造として，分割木 [7]，切断木 [8] などが提案され，それらを用いた領域探索アルゴリズムの計算量の解析が行われてきた．非軸直交一次元範囲検索は，複数の単体領域探索の組み合わせにより表現される検索であり，これら多くの研究の成果を適用可能である問題と言える．しかし，計算幾何学分野における従来の研究では，データおよび索引構造は主記憶に配置されることが仮定され，データベースシステムの索引機構としての利用は想定されなかった [2]．特に，データベースシステムにおける利用で重要となるデータの動的な更新への対応についての考慮が不十分であった．また，これらの研究の大部分は，$d = 2$ または $d = 3$ の場合に特化した議論となっており，一般の多次元空間中の範囲検索への考察は現在のところ半空間問い合わせに関する研究 [2] など限定した範囲に留まっている．

2.3 問い合わせ分布を考慮した索引構造

従来提案された索引手法の多くは，ある特定の性質を持つ任意の問い合わせ集合に対して良好な性能を示す単一の索引構造の提供を目標として設計された．その上で，提案された索引構造の実験的性能評価にあたっては，主に実例に即して得られた特定の問い合わせ集合に対する評価が行われた．これに対し，近年，個々の問い合わせがいくつかのパラメータの組で表されるものとし，問い合わせの各パラメータ値の分布が既知である場合に，そのような分布を持つ問い合わせ集合に対して良好な性能を示す索引構造を構築する手法についての研究が行われている [9]．本研究においても，様々な応用分野での利用が想定される非軸直交一次元範囲検索に対する実用上有効性の高い索引構造の提案を目指し，問い合わせのパラメータ分布を利用した索引構造の生成手法を提案する．

3. 準　　備

本節では用語と本研究が考察の対象とする問題の定義を与える．

3.1 問い合わせ対象データ空間

本研究では問い合わせの対象となるデータ空間として各軸の値を実数とする d 次元直交空間 $\mathbb{R}^d$ を想定する．$\mathbb{R}^d$ 空間上のベクトル $\mathbf{x} = (x_0, \ldots, x_{d-1})$ に対し，その第 i 次元成分 x_i を $\mathbf{x}_{[i]}$ と書く．

$\mathbb{R}^d$ 空間中の部分空間 $\mathfrak{s} \subset \mathbb{R}^d$ を領域と呼ぶ．領域 $\mathfrak{s}$ の d 次元超体積を $\|\mathfrak{s}\|$ により表す．

位置ベクトル $\mathbf{x}$ で与えられる一点のみからなる領域 $\{\mathbf{x}\}$ を $\mathbb{R}^d$ 空間上の点領域または点と呼び，混乱の無い限り $\mathbf{x}$ と書く．

d 次元超直方体は $\mathbb{R}^d$ 空間中の領域の一種であり，各軸に対する区間の組 $\mathfrak{r} = ([l_0, u_0], \ldots, [l_{d-1}, u_{d-1}])$（ただし $l_i \leqq u_i$）で表される．$\mathbf{l}(\mathfrak{r}) = (l_0, \ldots, l_{d-1}) \in \mathbb{R}^d$ を超直方体 $\mathfrak{r}$ の下端点と呼び，$\mathbf{u}(\mathfrak{r}) = (u_0, \ldots, u_{d-1}) \in \mathbb{R}^d$ を $\mathfrak{r}$ の上端点と呼ぶ．また，各軸における区間の長さを表すベクトル $\mathbf{m}(\mathfrak{r}) = \mathbf{u}(\mathfrak{r}) - \mathbf{l}(\mathfrak{r})$ を $\mathfrak{r}$ の形状と呼び，その成分の総和 $\sum_{i=0}^{d-1} \mathbf{m}(\mathfrak{r})$ を $\mathfrak{r}$ の周長と呼ぶ．

領域 $\mathfrak{s} \subset \mathbb{R}^d$ に対し，$\mathfrak{s}$ を包含する最小の超直方体を $\mathfrak{s}$ の最小包囲矩形と呼び，$mbr(\mathfrak{s})$ と表す．点領域に対する最小包囲矩形は点領域である．

3.2 非軸直交一次元範囲検索

本研究で考察の対象とする非軸直交一次元範囲検索は以下のように定義される問題である．

［定義 1］（非軸直交一次元範囲検索） $\mathcal{D}$ を $\mathbb{R}^d$ 空間中の n 個の点からなる集合とする．このとき，式 (1) により与えられる $\mathcal{D}$ の部分集合 $S(q,\mathcal{D})$ を求める問題を非軸直交一次元範囲検索と呼ぶ．ここで，$\mathbf{a} \in \mathbb{R}^d$ および $C_1, C_2 \in \mathbb{R}$ からなる三つ組 $q = (\mathbf{a}, C_1, C_2)$ を非軸直交一次元範囲検索の問い合わせ(注1)と呼び，$\mathbf{a}, C_1, C_2$ を問い合わせ q のパラメータと呼ぶ．

$$S(q,\mathcal{D}) \equiv \{\mathbf{x} \,|\, C_1 \leqq \mathbf{a}\cdot\mathbf{x} \leqq C_2, \mathbf{x} \in \mathcal{D}\} \tag{1}$$

非軸直交一次元範囲検索は $\mathbb{R}^d$ 空間中の二枚の平行超平面 $P_1 : \mathbf{a}\cdot\mathbf{x} = C_1, P_2 : \mathbf{a}\cdot\mathbf{x} = C_2$ によって挟まれる部分空間に属する $\mathcal{D}$ の要素 $\mathbf{x}$ を全て求める問題と考えることが可能である．式 (2) で与えられる $\mathbb{R}^d$ の部分空間 $R(q)$ を平行超平面領域と呼ぶ．平行超平面領域は非軸直交一次元範囲検索を領域探索問題と捉えた際の，問い合わせ q の問い合わせ領域である．本稿においては，特に混乱の無い限り $R(q)$ を単に q と表す．

$$R(q) \equiv \left\{\mathbf{x} \,\middle|\, C_1 \leqq \mathbf{a}\cdot\mathbf{x} \leqq C_2, \mathbf{x} \in \mathbb{R}^d\right\} \tag{2}$$

このとき，q の境界である超平面 P_1 および P_2 の法線ベクトルは問い合わせパラメータ $\mathbf{a}$ により与えられる．以降においては，問い合わせパラメータ $\mathbf{a}$ を問い合わせ q の法線ベクトルと呼ぶ．なお，式 (2) より明らかに超平面 P_1, P_2 は $\mathbb{R}^d$ 空間のいずれかの軸に直交しなくともよい．

非軸直交一次元範囲探索の応用例として，多変量解析の結果得られる超平面を用いたデータの分類，各種試験等の合否判定などがある．

3.3 問い合わせ分布

一般に索引構造を利用する際には，あらかじめ作成された索引に対し，多数の問い合わせによる検索が実行される．問い合わせ q の集合を $\mathcal{Q}$ と表す．問い合わせ集合 $\mathcal{Q}$ 中の問い合わせの各パラメータ値の分布には応用分野に応じた一定の偏りが存在すると考えられる．本研究では，あらゆる問い合わせ集合に対して良好な性能を示す単一の索引を構築するのではなく，実用上頻繁に発生する問い合わせに対して特に優れた性能を示す索引の生成手法の提案を目指す．そのため，検索の際に与えられる問い合わせの各パラメータについて，その分布関数および特性値の一部が予測可能であることを想定する．本稿で提案する手法は，予測される分布関数および特性値に従う問い合わせ集合が与えられた場合に適切な索引を生成する手法である．以下では問い合わせ集合 $\mathcal{Q}$ が従うと推定される分布を $\Phi^{\mathcal{Q}}$ と表し，$\mathcal{Q}$ の問い合わせ分布と呼ぶ．

4. 非軸直交一次元範囲検索への R 木の応用

本研究ではデータベースシステムを利用して二次記憶中に格納されたデータに対する非軸直交一次元範囲検索の適用を考える．一般にデータベースシステムにおいては，データの追加，削除を含む変更が頻繁に発生する．データの追加，削除に柔軟に対応するために，非軸直交一次元範囲検索に適した木構造索引と，その動的生成手法について考える．データおよび索引が二次記憶中に格納されることを想定し，木構造索引による検索時のアクセスノード数を索引構造の性能評価の主な基準とする．

我々は非軸直交一次元範囲検索のための木構造索引の構築にあたり，既存の軸直交範囲検索手法である R 木に着目する．

4.1 非軸直交一次元範囲検索に対する R 木の有用性

本研究で非軸直交一次元範囲検索のための索引構造として R 木に基づく手法を利用する理由は以下の通りである．

- R 木が本質的に非軸直交一次元範囲検索に利用可能であるため．

過去の研究により，R 木が任意の形状の領域を問い合わせ領域とする領域探索に利用可能であることが知られている [1]．ただし，R 木を任意形状の領域探索に利用した場合の性能は十分には研究されていない．よって，非軸直交一次元範囲検索に R 木を利用した場合の性能について考察し，十分な性能が得られない場合には改良を加える必要がある．

- R 木が多くのデータベースシステムにおいて実用レベルで利用されている索引手法であるため．

R 木は Oracle [10], PostgreSQL [11], [12], Informix [13] など多くのデータベースシステム上で広く利用されている安定した実装を持つ索引機構である．安定した実装を持つ索引機構を利用して非軸直交一次元範囲検索を行うことにより，現実の応用への適用が容易に行えると考えられる．

- R 木がデータの動的更新に対応した索引であるため．

R 木には静的構築手法と動的構築手法が存在するが，動的構築手法を利用することにより，データの追加や削除などを含む更新への対応が可能となる．一般にデータベースシステムの応用においてはデータの追加，削除が頻繁に発生するため，データの動的な更新への対応は不可欠である．

4.2 R 木による多次元領域探索

R 木は多次元空間内の領域探索を効率的に処理するための平衡探索木である．一般に R 木が索引付けの対象とするデータの各要素は $\mathbb{R}^d$ 空間中の領域と対応し，R 木は各データ要素に対応する領域を最小包囲矩形で再帰的に包含することにより索引を構成する．本研究では点領域の集合 $\mathcal{D}$ に対する R 木の構築を考える．

R 木の葉ノード v は k_v 個のデータ要素への参照を，非葉ノード v は k_v 個の子ノードへの参照を保持する．k_v はあらかじめ与えられた定数 m,M に対し $m \leqq k_v \leqq M$ をみたす整数であり，ノード v の次数と呼ばれる．定数 m と M はそれぞれ R 木の最小次数，最大次数と呼ばれる．ノード v に対し，v の子（v が葉ノードの場合はデータ要素）の集合を $c(v)$ と書く．また，根を除くノード v に対し，その親を $p(v)$ と書く．

ノード v の領域 $\mathfrak{r}(v)$ は，

$$\mathfrak{r}(v) \equiv mbr\Big(\bigcup_{v' \in c(v)} \mathfrak{r}(v')\Big)$$

(注1)：本稿では問い合わせインスタンスを単に問い合わせと呼ぶ．

により再帰的に定義される．$\mathfrak{r}(v)$ を単に v と書くこともある．

R 木を利用した領域探索においては，与えられた問い合わせ領域 q に対し，q と重なる領域を持つノードを根から葉へ順に参照することにより，探索を行う．したがって，ある問い合わせ領域 q による領域探索を行うためには，q と各ノードの領域である超直方体とが重なりを持つかの判定が必要である．問い合わせ領域 q が平行超平面領域である場合には，q の法線ベクトルの各成分 $\mathbf{a}_{[i]}$ の符号に従って決定する超直方体の 2 頂点に対し，それらが q の境界である二枚の超平面のどちら側に位置するかを判定することで，$O(d)$ の計算量で重なり判定が可能である．

4.3 R* 木生成戦略

R 木の代表的な動的構築手法として，R* 木生成手法 [14] が知られている．R* 木は Kriegel らのテストコレクション [15] に対して良好な性能を示すことを目標として設計された，動的生成手法により生成される R 木である．R* 木生成手法においてはデータの動的追加・削除の際に，データが配置される木構造中の位置を以下の 4 種の評価基準に基づいて決定する．このような R 木の生成過程における処理選択の評価基準を R* 木の生成戦略と呼ぶ．

超体積の最小化　ノード領域の超体積の最小化により，検索時に無駄な領域に探索が及ぶ可能性を削減可能である．

重複超体積の最小化　木構造の同一の高さに存在するノードの領域同士が互いに重なりあう超体積を最小化する．ノード領域同士の重なりあう体積の削減により，検索時に木の複数の枝を辿る可能性を削減可能である．

記憶領域利用効率の最大化　ノードの次数を常に M に近い値に維持することにより，木の高さを削減可能である．

周長の最小化　ノード領域の周長を最小化することにより，各ノード領域の形状ができる限り超立方体に近い形状になるように調整する．

R* 木の生成戦略のうち，超体積の最小化，重複超体積の最小化，記憶領域利用効率の最大化については，非軸直交一次元範囲検索においても有用であると考えられる．

生成戦略のうち，各ノード領域の周長を最小化するという戦略は，軸直交範囲検索に関する以下の考察に基づくものである．

> 軸直交範囲検索の超直方体問い合わせ領域が超立方体に近い形状をしている場合，超立方体に近い形状の領域を持つノードを多く生成することで良好な性能を得る [14].

しかし，非軸直交一次元範囲検索の問い合わせ平行超平面領域に対しては，上記の性質はあてはまらないと考えられる．そこで本研究では，ノード領域の周長を最小化する戦略に替わる新たな戦略を考える．

5. R 木の重み付き生成戦略

我々は，非軸直交一次元範囲検索に適した R 木の動的生成手法のための戦略として，重み付き生成戦略を提案する．重み付き生成戦略は，R 木を構成する全てのノードのうち，領域が問い合わせ領域と交差するノードが少ないほど良好な性能が得られるという考察に基づく．本研究では問い合わせ集合の分布パラメータが既知である場合に，その情報を R 木の生成に利用する戦略を考える．

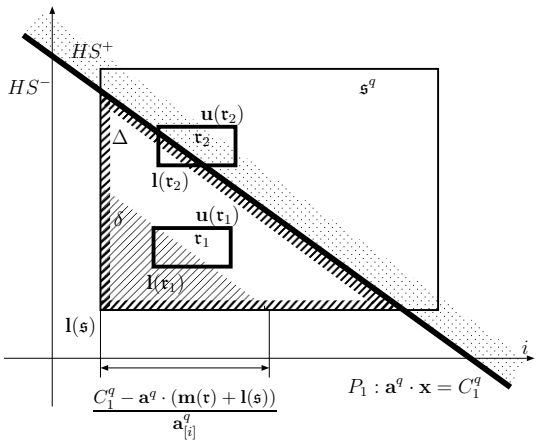


図 1　超直方体と問い合わせ領域の交差
Fig. 1　Intersection of rectangles and a query region

5.1 R 木ノードの領域形状による非軸直交一次元範囲検索性能への影響

問い合わせ集合として平行超平面領域の集合 $\mathcal{Q}$ を考える．平行超平面領域 $q = (\mathbf{a}^q, C_1^q, C_2^q)$ を $\mathcal{Q}$ の要素の一つとする．まず最初に $\mathbf{a}^q_{[i]} \geqq 0$ の場合を考えよう．q の境界となる超平面の一つ $P_1^q : \mathbf{a}^q \cdot \mathbf{x} = C_1^q$ によって分離される $\mathbb{R}^d$ の半空間のうち，q を含む側を HS^+ とし，q を含まない側を HS^- とする．

いま，ある超直方体領域 $\mathfrak{s}^q$ が存在し，R 木を構成する各ノードの領域である超直方体の下端点が領域 $\Delta = \mathfrak{s}^q \cap HS^-$（図 1 中の太斜線で囲まれる領域）中で一様に分布することを仮定する．ただし，$\mathfrak{s}^q$ は下端点 $\mathbf{l}(\mathfrak{s}^q)$ のみが HS^- に属し，他の全ての頂点が HS^+ に属する超直方体であるとする．

領域 Δ 中に下端点を持つ矩形 $\mathfrak{r}$ が問い合わせ領域 q と交差を持たないとき，$\mathfrak{r}$ の下端点が存在し得る領域は図 1 中の細斜線で示される領域 δ の内部となる．図 1 中では，$\mathfrak{r}_1$ と $\mathfrak{r}_2$ は同一の形状 $\mathbf{m}(\mathfrak{r})$ を持つ超直方体であるが，$\mathfrak{r}_1$ が領域 δ 内部に下端点を持たないため，q と交差するのに対し，$\mathfrak{r}_2$ は領域 δ 内部に下端点を持つため，q と交差しない．

これにより，$\mathfrak{r}$ が q と交差を持たない確率 $\mathcal{P}^q(\mathfrak{r})$ は

$$\mathcal{P}^q(\mathfrak{r}) = \frac{\|\delta\|}{\|\Delta\|} = \frac{\left(\Gamma_q - \sum_{i=0}^{d-1} \mathbf{a}^q_{[i]} \mathbf{m}(\mathfrak{r})_{[i]}\right)^d}{{\Gamma_q}^d}$$

で与えられる．ここで $\Gamma_q = C_1^q - \sum_{i=0}^{d-1} |\mathbf{a}^q_{[i]}| \mathbf{l}(\mathfrak{s})^q_{[i]}$ は $\mathfrak{r}$ に依存せずに決まる定数である．

$\mathbf{a}^q_{[i]} < 0$ の場合についても同様に考えると，確率 $\mathcal{P}^q(\mathfrak{r})$ は一般に

$$\mathcal{P}^q(\mathfrak{r}) = \frac{\left(\Gamma_q - \sum_{i=0}^{d-1} |\mathbf{a}^q_{[i]}| \mathbf{m}(\mathfrak{r})_{[i]}\right)^d}{{\Gamma_q}^d}$$

となる．さらに q の境界を構成するもう一方の超平面 P_2^q : $\mathbf{a}^q \cdot \mathbf{x} = C_2^q$ に関しても同様の議論を行い，同一の式を得る．

したがって，問い合わせ集合 $\mathcal{Q}$ に対して最適な R 木のノード形状は

$$\sum_{q\in\mathcal{Q}} \mathcal{P}^q(\mathfrak{r}) = \sum_{q\in\mathcal{Q}} \frac{\left(\Gamma_q - \sum_{i=0}^{d-1} |\mathbf{a}_{[i]}^q| \mathbf{m}(\mathfrak{r})_{[i]}\right)^d}{{\Gamma_q}^d}$$

を最大化する $\mathbf{m}(\mathfrak{r})$ を持つことがわかる．

ここで，各 $q \in \mathcal{Q}$ に対し，$\mathfrak{s}^q$ を適切に取ることにより，全ての $q \in \mathcal{Q}$ について定数 Γ_q を互いに等しい値にすることができる．これを Γ とすると，式 (3) の値を最大化するようにノードを生成することで，非軸直交一次元範囲検索に適した R 木が構築される．

$$\sum_{q\in\mathcal{Q}} \left(\Gamma - \sum_{i=0}^{d-1} |\mathbf{a}_{[i]}^q| \mathbf{m}(\mathfrak{r})_{[i]}\right)^d \tag{3}$$

しかし，R 木の構築時には問い合わせ集合 $\mathcal{Q}$ が明示的には与えられておらず，その分布 $\Phi^{\mathcal{Q}}$ のみが既知である場合には式 (3) の計算は困難である．そこで本研究では，式 (3) の最大化に対する近似として，問い合わせ分布 $\Phi^{\mathcal{Q}}$ より得られる法線平均ベクトル $\boldsymbol{\mu}^{\mathbf{a}}$ の各成分の絶対値 $w_i = |\boldsymbol{\mu}_{[i]}^{\mathbf{a}}|$ を成分として持つベクトル $\mathbf{w} = (w_0, \ldots, w_{d-1})$ を利用した式 (4) の最小化を用いる．$\mathbf{w}$ はノードの最小包囲矩形の各軸方向区間長を評価する際の重みを与える量であることから，重みベクトルと呼ぶ．

$$badness(\mathfrak{r}) \equiv \mathbf{w} \cdot \mathbf{m}(\mathfrak{r}) \tag{4}$$

$badness(\mathfrak{r})$ を最小化する領域を持つノードを利用することにより，現実の応用での利用に対応した生成戦略を得る．この戦略を重み付き生成戦略と呼ぶ．

5.2 アルゴリズム

提案手法は動的生成手法であるので，既に構築済みの R 木 T に対し，データ要素 $\mathbf{x} \in \mathcal{D}$ を逐次追加することにより木構造を構築する．この処理を手続き Insert と呼ぶ．

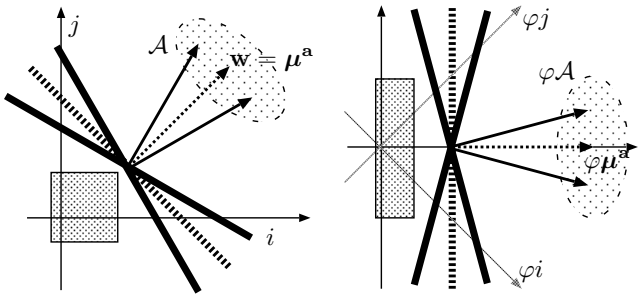
手続き Insert 中で最初に行われる処理はデータ挿入ノードの選択手続き ChooseSubtree である．手続き ChooseSubtree はデータ要素 $\mathbf{x}$ の R 木 T への追加の際に $\mathbf{x}$ を T のどのノードに追加すべきかを決定する．提案手法における手続き ChooseSubtree は R* 木生成手法におけるデータ要素挿入時ノード選択アルゴリズムと同一であるので本稿では省略する．

手続き ChooseSubtree の結果選択されたノードの次数が M 未満である場合には，選択されたノードに $\mathbf{x}$ を挿入し，データ要素追加処理は終了する．選択されたノードの次数が M である場合には手続き OverflowTreatment および手続き ReInsert を利用して選択されたノードの子を他のノードに移動し，$\mathbf{x}$ を挿入するための空きを作る．これらのアルゴリズムも R* 木生成手法におけるアルゴリズムと同一である．

既に手続き ReInsert が実行されているにもかかわらず，なおノードが M 以上の次数を持つ場合には，ノードの分割処理を行い，動的に木構造を再編する．ノードの分割処理は R 木の動的生成過程において木構造が成長する唯一の処理であり，各ノードの領域の形状はノード分割処理時の子の分配手法に支配される．そこで，本研究ではノードの分割処理の際の分配処理に，先に示した式 (4) を利用する．ノード分割アルゴリズムを以下に示す．ノード分割手続き Split は木 T 中の $M+1$ 個の子を持つノード v を分割し，T を再編する．

［アルゴリズム 1］（ノード分割）

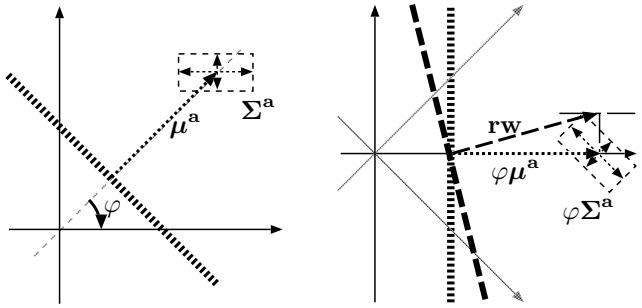
```
Split(var T; var v)
begin
  var ν : Node;
  var axis : Axis;
  axis ← ChooseSplitAxis(v);
  DistributeEntries(v, ν, axis);
  p(v) に ν を挿入;
  if M < k_{p(v)} then
    OverflowTreatment(T, p(v))
end

ChooseSplitAxis(v)
begin
  for j ← 0 to d − 1 do
  begin
    cl ← ch ← c(v);
    cl を領域の下端点の j 軸値の昇順に整列;
    ch を領域の上端点の j 軸値の昇順に整列;
    for k ← m to M − m do
    begin
      ml_j ← ml_j
             + badness(mbr(⋃_{i=0}^{k−1} cl_i))
             + badness(mbr(⋃_{i=k}^{M−1} cl_i));
      mh_j ← mh_j
             + badness(mbr(⋃_{i=0}^{k−1} ch_i))
             + badness(mbr(⋃_{i=k}^{M−1} ch_i));
      m_j ← min(ml_j, mh_j)
    end
  end;
  return m_j が最小となる j
end

DistributeEntries(var v,ν; axis)
begin
  cl ← ch ← c(v);
  cl を領域の下端点の axis 軸値の昇順に整列;
  ch を領域の上端点の axis 軸値の昇順に整列;
  for k ← m to M − m do
  begin
    ovl_k ← ‖mbr(⋃_{i=0}^{k−1} cl_i) ∩ mbr(⋃_{i=k}^{M−1} cl_i)‖;
```



（A）重みベクトルの各軸成分が同一である問い合わせ集合と生成されるノード形状　（B）回転変換された空間と生成されるノード形状

図 2　空間の回転とノード領域の形状

Fig. 2　Relationship between rotation of space and shapes of nodes



（A）変換前の法線平均ベクトルと法線標準偏差行列　（B）回転後の空間と重みベクトルの決定

図 3　回転変換を利用した重みベクトルの決定

Fig. 3　Weight vector for rotated space

$ovh_k \leftarrow \left\| mbr(\bigcup_{i=0}^{k-1} ch_i) \cap mbr(\bigcup_{i=k}^{M-1} ch_i) \right\|$;

$ov_k \leftarrow \min(ovl_k, ovh_k)$

end

ov_k が最小となる k を選択;

$c_k, \ldots, c_{M-1}$を v から削除し ν へ移動

end

提案戦略と R* 木生成戦略の違いはノード分割処理中の子の振り分け基準軸の選択手法のみであることに注意せよ．また，任意の 2 軸 i, j に対して $\mathbf{w}_{[i]} = \mathbf{w}_{[j]}$ である場合には，提案戦略は R* 木生成戦略と完全に同一である．すなわち，提案戦略は R* 木生成戦略の拡張となっている．

6. RW 木：非軸直交一次元範囲検索用索引機構

前節で示した重み付き生成戦略では，重みベクトル $\mathbf{w}$ の全ての成分が同一の値である場合に R* 木と同一の R 木を生成する．これは問い合わせ集合のパラメータ分布が未知である場合にも R* 木と等しい性能を得るような重みベクトルを設定可能であることを保障する．一方，問い合わせ集合に含まれる平行超平面領域が上記のように全ての成分が均等な法線ベクトルを持つことが既知である場合にも，重み付き生成戦略は R* と同一の木を生成してしまう．

そこで我々は先に示した重み付き生成戦略とデータ空間の座標変換を用いて，問い合わせ集合の分布が既知である場合に常に良好な性能を得る索引機構を提案する．

問い合わせ集合 $\mathcal{Q}$ に属する全ての問い合わせ $q \in \mathcal{Q}$ の法線ベクトル $\mathbf{a}^q$ からなる集合を $\mathcal{A}$ により表す．図 2A のように $\mathcal{A}$ が任意の軸 i, j に対して同一の軸方向平均長 $|\boldsymbol{\mu}^{\mathbf{a}}_{[i]}| = |\boldsymbol{\mu}^{\mathbf{a}}_{[j]}|$ を持つ分布に従う場合を考える．このとき，前節で示したように重み付き生成戦略は R* 生成戦略と同一のアルゴリズムとなり，超立方体に近い形状の領域を持つノードが多く生成される．

これに対し，図 2B のように $\mathbb{R}^d$ 空間全体に対して適切な変換を施すことにより，より問い合わせパラメータの異方性を利用した矩形の選択が可能となる．

本研究では，このような変換として $\boldsymbol{\mu}^{\mathbf{a}}$ ベクトルをデータ空間のいずれかの軸と平行となるように回転する d 次元一次変換 $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ を用いる．φ は全単射であるため，検索対象となるデータと問い合わせ領域の双方にこの変換を施すことによって，任意の問い合わせに対して変換を行わない場合と同一の検索結果を得る．$\mathbb{R}^d$ 空間における回転変換は極座標変換と平行移動の組合せによって実行可能であるため，φ による変換は N 個の要素を持つデータ集合または問い合わせ集合に対し，$O(d)$ の記憶領域を用いて $O(dN)$ の計算量で計算可能である．

前節では式 (3) の近似として $\mathbf{w}_{[i]} = |\boldsymbol{\mu}^{\mathbf{a}}|$ を用いた式 (4) を用いたが，索引付け対象空間を φ 変換することにより，より良い近似を利用可能である．

$\mathbb{R}^d$ 空間の φ 変換により，$\boldsymbol{\mu}^{\mathbf{a}}$ ベクトルは，軸に平行なベクトルに変換される．すなわち，$\varphi\boldsymbol{\mu}^{\mathbf{a}}$ の要素は一つの軸を除いて値 0 を持つ．以下においては，$\varphi\boldsymbol{\mu}^{\mathbf{a}}$ が 0 でない値を持つ軸を第 0 軸とする．φ による変換後の問い合わせ集合に含まれる超平面の法線ベクトルの各軸方向成分の平均長は，分布 $\varphi\Phi^{\mathcal{Q}}$ の標準偏差行列 $\varphi\boldsymbol{\Sigma}^{\mathbf{a}}$ の値に依存する（図 3）．そこで，生成戦略で利用する重みベクトルとして $\mathbf{w}$ のかわりに式 (5) で定義されるベクトル $\mathbf{rw} = (rw_0, \ldots, rw_{d-1})$ を用いる．ただし式 (5) 中で $(\varphi\boldsymbol{\Sigma}^{\mathbf{a}})_{[i,j]}$ は行列 $\varphi\boldsymbol{\Sigma}^{\mathbf{a}}$ の第 i 行 j 列成分を表す．

$$rw_i \equiv \begin{cases} (\varphi\boldsymbol{\mu}^{\mathbf{a}})_{[i]} & \text{for } i = 0, \\ \sum_{j=0}^{d-1} |(\varphi\boldsymbol{\Sigma}^{\mathbf{a}})_{[i,j]}| & \text{otherwise.} \end{cases} \tag{5}$$

索引付けの対象となるデータ空間の変換およびベクトル $\mathbf{rw}$ による重み付けを利用したアルゴリズムにより生成される索引機構を RW 木と呼ぶ．RW 木の生成アルゴリズムと検索アルゴリズムを以下に示す．

［アルゴリズム 2］（RW 木の生成）

CreateRWTree($\mathcal{D}$; $\Phi^{\mathcal{Q}}$)

begin

$\Phi^{\mathcal{Q}}$の値に従って φ，$\mathbf{rw}$ を求める;

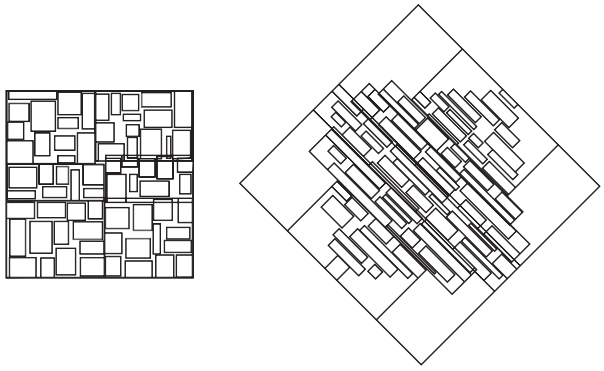
$badness(\mathfrak{r}) \equiv \mathbf{rw} \cdot \mathbf{m}(\mathfrak{r})$;

$T \leftarrow \phi$;

for $\mathbf{x} \in \varphi\mathcal{D}$ **do**

Insert(T, $\mathbf{x}$)

end



（A）R* 木　　（B）RW 木

図 4　R 木のノード領域形状
Fig. 4　Shapes of R-tree nodes

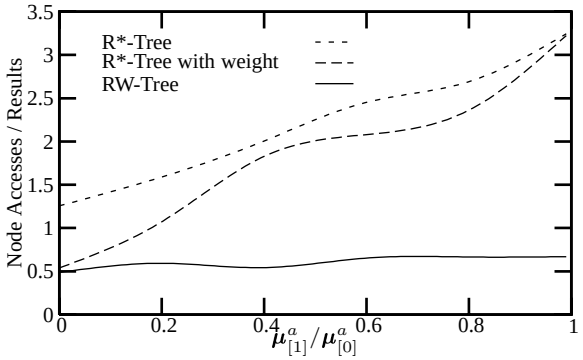


図 5　$\boldsymbol{\mu}^{\mathbf{a}}$ の値が性能に及ぼす影響
Fig. 5　Varying value of $\boldsymbol{\mu}^{\mathbf{a}}$

［アルゴリズム 3］（RW 木による非軸直交一次元範囲検索）
SearchRWTree(T; q; Φ^{Ω})
begin
　Φ^{Ω}の値に従って φ を求める;
　SearchRTree(T, φq)
end

ここで，手続き Insert は前節で示したアルゴリズムを利用する R 木生成処理を，手続き SearchRTree は通常の R 木検索処理を表す.

7. 評　　価

提案手法である RW 木の性能について考察するため，実際に RW 木生成アルゴリズムを用いて R 木を作成し，実験的評価を行った．実験においては，大森らの評価手法[9] を参考に，異なる分布関数とパラメータに従う複数の問い合わせ集合をランダムに生成し，それらの問い合わせ集合の分布パラメータ変動に対する性能の変化を観測した.

本研究では Agarwal らの考察[2] に基づき，問い合わせにより得られるデータ 1 個あたりのノードアクセス数を評価の基準とする．この基準に従い，既存の動的生成手法により生成された R 木と提案手法により生成された R 木のそれぞれを用いて検索を行った場合の性能評価を行った.

実験には Hadjieleftheriou らによる空間索引構築基盤ライブラリ[16] を利用した．ライブラリ上の R* 木生成アルゴリズムの実装および R 木検索アルゴリズムの実装をもとに，重み付き生成戦略を利用した R* 木生成アルゴリズム，RW 木生成アルゴリズムおよびそれらにより生成された木を利用する検索アルゴリズムを実装し，評価に使用した．R 木の最大次数と最小次数は各々$M = 20, m = 8$ とし，その他の R* 木生成パラメータの値は文献[14] に従った.

7.1　2 次元空間上での木の生成

評価実験に先立ち，本手法により生成されるノード領域の形状を確認するために，2 次元空間に分布するデータを用いた R 木の生成を行った．R 木の生成には $[0, 1]^2$ 空間内に一様分布する 1000 個のデータを利用した．また，問い合わせ集合は，$\boldsymbol{\mu}^{\mathbf{a}} = (1, 1), \boldsymbol{\Sigma}^{\mathbf{a}} = 0.2\mathbf{I}$ を持つ正規分布に従うものとした.

図 4 に R* および重み付き戦略で生成される R 木のノードの様子を示す．図より，R* 木（図 4A）においては正方形に近い形状の長方形が多く得られているのに対し，提案手法（図 4B）においては平均法線ベクトルに垂直な辺が長い長方形が多く得られていることが確認できる.

7.2　法線ベクトルの平均の性能への影響

性能評価のための実験として，問い合わせ集合の法線平均ベクトルが R 木による検索性能に及ぼす影響を調べた.

R 木の生成には $\mathbb{R}^3$ 空間内に分布する 10000 個のデータを利用した．データは各軸独立に標準正規分布 $N(0, 1)$ に従う.

問い合わせ分布として $\mu_0 = 1, 0 \leqq \mu_1 = \mu_2 \leqq 1$ に対し，$\mathbf{a}_{[i]}$ が各軸独立に正規分布 $N(\mu_i, 0.2)$ に，$C_2 - C_1$ が対数正規分布 $LN(-2, 0.5)$ にそれぞれ従い，C_1 の値が常に 0 であるような分布 Φ_1^{Ω} を用意した．分布 Φ_1^{Ω} は法線ベクトルに関して特性値 $\boldsymbol{\mu}^{\mathbf{a}} = (1, \mu_1, \mu_2), \boldsymbol{\Sigma}^{\mathbf{a}} = 0.2\mathbf{I}$ を持つ.

このようにして作成した R 木に対し，Φ_1^{Ω} に従う分布を持つ 1000 個の問い合わせからなる問い合わせ集合を適用した際の，検索の結果得られるデータ要素 1 個あたりのアクセスノード数を図 5 に示す.

図より，提案手法である RW 木による検索では R* 木による検索と比較して少ないアクセスノード数で検索が実行されていることがわかる．特に，R* 木においては $\boldsymbol{\mu}^{\mathbf{a}}_{[1]}, \boldsymbol{\mu}^{\mathbf{a}}_{[2]}$ の値が $\boldsymbol{\mu}^{\mathbf{a}}_{[0]}$ の値に近付くに従ってアクセスノード数が増大しているのに対し，RW 木では空間の回転変換を行った事により $\boldsymbol{\mu}^{\mathbf{a}}$ の値にかかわらずほぼ一定の性能が得られている.

また，回転変換を利用せずに，重み付き生成戦略のみを R* 木生成手法に適用した場合でも $\boldsymbol{\mu}^{\mathbf{a}}_{[1]}/\boldsymbol{\mu}^{\mathbf{a}}_{[0]}$ が 0.4 より小さい範囲では R* 木と比較して良好な性能が得られている．$\boldsymbol{\mu}^{\mathbf{a}}_{[1]}/\boldsymbol{\mu}^{\mathbf{a}}_{[0]}$ が 0.4 より大きい範囲では，重み付き生成戦略の効果は低いが，その場合でも R* 木と同程度の性能を確保している.

7.3　法線ベクトルの標準偏差の性能への影響

次に，問い合わせ集合の法線標準偏差行列が R 木による検索性能に及ぼす影響を調べた.

R 木の生成に用いるデータは前小節 7.2 の実験と同一とした.

問い合わせ分布として，$\mu_0 = 1, \mu_1 = \mu_2 = 0.2, 0 \leqq \sigma \leqq 1$

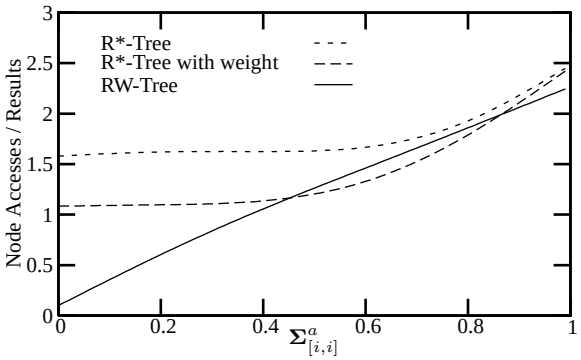


図 6　$\mathbf{\Sigma^a}$ の値が性能に及ぼす影響

Fig. 6　Varying value of $\mathbf{\Sigma^a}$

に対し，$\mathbf{a}_{[i]}$ が各軸独立に正規分布 $N(\mu_i, \sigma)$ に，$C_2 - C_1$ が対数正規分布 $LN(-2, 0.5)$ にそれぞれ従い，C_1 の値が常に 0 であるような分布 $\Phi_2^{\mathcal{O}}$ を用意した．このような分布は法線ベクトルに関して特性値 $\boldsymbol{\mu}^{\mathbf{a}} = (1, 0.2, 0.2), \mathbf{\Sigma^a} = \sigma\mathbf{I}$ を持つ．

このようにして作成した R 木に対し，$\Phi_2^{\mathcal{O}}$ に従う分布を持つ 1000 個の問い合わせからなる問い合わせ集合を適用した際の，検索の結果得られるデータ要素 1 個あたりのアクセスノード数を図 6 に示す．

図より，法線ベクトルの標準偏差が小さい場合には RW 木の性能が最も優れており，次いで重み付き生成戦略を導入した R* 木の性能が良いことがわかる．一方，標準偏差の値が 0.6 より大きな範囲においては，比較した三種の手法の性能には大きな差は見られない．これは，大きな標準偏差を持つ分布に対しては，問い合わせ集合中の各問い合わせ領域に対する法線ベクトルの推定が困難であり，R 木の生成時に適切な形状を持つノードを選択できないためであるためと考えられる．このことから，与えられた問い合わせ分布の標準偏差行列の値によって，RW 木を使うか否かを選択する手法が有用であると考えられる．

本稿では，問い合わせ集合のパラメータ分布が既知の場合の非軸直交一次元範囲検索に適した索引構成手法である RW 木を提案し，RW 木により生成された索引構造を利用した検索性能の実験的評価を行った．

RW 木生成手法は既存の索引構成手法である R* 木の拡張として得られ，問い合わせ集合のパラメータ分布が既知でない場合においても，R* 木と同一の木構造を生成する．また，実装においても R* 木の構築および検索に関する実装のほぼ全ての部分を流用可能であり，実用に供されるデータベースシステム上からの利用も容易であると考えられる．

実装評価においては，分布関数の母数を変化させて生成した多数の問い合わせ集合に対し，提案手法により生成された木構造索引を利用した検索を行った際のアクセスノード数を計測した．実験結果から，問い合わせ集合のパラメータ分散が小さい場合には提案手法が従来手法と比較して優位であることを確認した．一方，問い合わせ集合のパラメータ分散が大きな場合は提案手法は R* 木と同程度の性能を示した．この結果は提案手法のパラメータとして与えられる問い合わせ分布の推定に失敗した場合の検索性能の低下に関する示唆を与えるが，問い合わせ分布の推定の失敗が提案手法の性能に及ぼす影響については，今後より詳細な検討を行う必要がある．

提案手法の本質的問題点として，データ空間の回転変換に伴う数値計算誤差の発生がある．評価実験においては計算誤差の影響は観測されなかったが，回転変換を単精度浮動小数点数演算で実行した実験では，検索の結果本来得られるべきデータと異なるデータが検索される例がごく少数ながら観測された(注2)．したがって，提案手法はそのままでは検索結果データの正確性を要求される多くのデータベース応用分野では利用できない．回転変換に伴う計算誤差を解析し，正確な検索結果集合を返すアルゴリズムを開発することが今後の課題である．

文　献

[1] V. Gaede and O. Günther: “Multidimensional access methods”, ACM Comput. Surv., **30**, 2, pp. 170–231 (1998).

[2] P. K. Agarwal, L. Arge, J. Erickson, P. G. Franciosa and J. S. Vitter: “Efficient searching with linear constraints”, J. Comput. Syst. Sci., **61**, 2, pp. 194–216 (2000).

[3] M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf: “Computational Geometry Algorithms and Applications”, Springer-Verlag (1997).

[4] J. Matoušek: “Geometric range searching”, ACM Comput. Surv., **26**, 4, pp. 422–461 (1994).

[5] A. Guttman: “R-trees: A dynamic index structure for spatial searching”, Proc. of ACM SIGMOD Intl. Conf. on Management of Data, pp. 47–57 (1984).

[6] D. Comer: “The ubiquitous B-tree”, ACM Comput. Surv., **11**, pp. 121–137 (1979).

[7] J. Matoušek: “Efficient partition trees”, Discrete Comput. Geom., **8**, pp. 315–334 (1992).

[8] B. Chazelle: “Cutting hyperplanes for divide-and-conquer”, Discrete Comput. Geom., **9**, pp. 145–158 (1993).

[9] 大森, 佐藤, 星：“問い合わせ分布を考慮した R 木における領域分割方式”, 信学論（D-I）, **J86-D-I**, 10, pp. 746–761 (2003).

[10] C. Murray: “Oracle Spatial User’s Guide and Reference”, Oracle Corporation, release 9.2 edition (2002).

[11] The PostgreSQL Global Development Group: “PostgreSQL 7.4.2 Documentation” (2003).

[12] J. M. Hellerstein, J. F. Naughton and A. Pfeffer: “Generalized search trees for database systems”, Proc. 21st Int. Conf. Very Large Data Bases, VLDB (Eds. by U. Dayal, P. M. D. Gray and S. Nishio), Morgan Kaufmann, pp. 562–573 (1995).

[13] Informix Corporation: “Informix R-Tree Index User’s Guide” (1999).

[14] N. Beckmann, H. Kriegel, R. Schneider and B. Seeger: “The R*-tree: An efficient and robust access method for points and rectangles”, Proc. of ACM SIGMOD Intl. Conf. on Management of Data (1990).

[15] H. P. Kriegel, M. Schiwietz, R. Schneider and B. Seeger: “Performance comparison of point and spatial access methods”, SSD ’90: Proc. of the 1st Symposium on Design and Implementation of Large Spatial Databases, Springer-Verlag New York, Inc., pp. 89–114 (1990).

[16] M. Hadjieleftheriou, E. Hoel and V. Tsotras: “SaIL: A library for efficient application integration of spatial indices”, Proc. of the 16th Intl. Conf. on Scientific and Statistical Database Management (SSDBM), Santorini, Greece (2004).

(注2)：検索適合データ 2000 個に対し 1 個程度